Neural Diffusion Graph Convolutional Network for Predicting Heat Transfer in Selective Laser Melting

Benjamin Uhrich[1,2(✉)], Tim Häntschel[1,2], Martin Schäfer[3], and Erhard Rahm[1,2]

[1] Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig, Leipzig, Germany
uhrich@informatik.uni-leipzig.de
[2] Leipzig University, Leipzig, Germany
[3] SIEMENS AG, Berlin, Germany
http://www.scads.ai/

Abstract. The quality of components produced through additive manufacturing processes, such as selective laser melting (SLM), is significantly influenced by heat transfer phenomena. Numerical simulations have emerged as valuable tools for gaining a deeper understanding of these processes. Deep investigation is made possible by a large amount of sensor data in this area. Both offers the potential to reduce the cost and time associated with empirical experimentation. Physics-informed neural networks (PINNs) combine the data-driven capabilities of deep neural networks with the mathematical formulations of physical laws, such as heat diffusion. In particular, the gap between numerical simulations and data observations can be bridged. In this paper, we present a novel neural diffusion graph convolutional network (NDGCN) designed to reveal physically interpretable parameters and accurately predict heat transfer dynamics during the SLM process. Our methodology involves representing the fabricated part as a graph model, constructed from high-dimensional data. This facilitates the integration of complex geometries and thermal properties into our predictive framework.

Keywords: Graph convolutional network · Neural diffusion · 3D printing

1 Introduction

SLM has emerged as a transformative technology in additive manufacturing, enabling the fabrication of intricate components with unprecedented precision. However, the quality of SLM-produced parts is profoundly influenced by the dynamic interplay of circulating heating and cooling processes. Undesirable thermomechanical distortions, porosity, and cracks are often caused by local and global overheating or rapid cooling. To this end, a thorough understanding of heat transfer phenomena is required to optimize manufacturing results. In the pursuit of such understanding, mathematical foundations have played a central role. A partial differential equation (PDE) is used for modeling heat diffusion. These equation provide a rigorous framework for comprehending the underlying physics of heat transfer. Nonetheless, there exists a challenge: the disparities between numerical analysis based on these equation and real-world observations

R. P. Barneva et al. (Eds.): IWCIA 2024/ISAIM 2024, LNCS 14494, pp. 150–164, 2024.
https://doi.org/10.1007/978-3-031-63735-3_9

obtained from sensors. Furthermore, the collection of comprehensive data of an entire component in the complicated SLM process proves to be impossible. Traditional thermal imagers can only capture surface temperatures, leaving the dynamics of the building's interior largely undetected. To address these challenges, PINNs have emerged as a powerful tool by harmonizing data-driven modeling with the diffusion equation. This fusion of data-driven and physics-based approaches not only expedites the training process but also extends our ability to predict heat transfer behaviors within SLM, even in areas devoid of sensor data. Building on the success of neural networks, Graph Convolutional Neural Networks (GCNs) have demonstrated remarkable predictive capabilities in various applications involving large-scale data relationships. Intriguingly, the mathematical description of GCNs functionality closely aligns with the structure of the explicit Euler method, offering a promising avenue for knowledge transfer [4]. In this paper, we propose a tailored NDGCN. It is designed to model discretized temperature fields within a built part as node features, facilitating predictions of heat transfer during additive manufacturing. Additionally, we explore the incorporation of different regularization techniques based on the knowledge of heat transfer properties to enhance training efficiency and accuracy. To the best of our knowledge, this work represents the first application of a NDGCN in the context of SLM for heat transfer prediction. We make an important contribution to this as follows:

- We introduce a NDGCN for the discovery of physically interpretable parameters and the prediction of temperature fields using real measurement data.
- Our approach is benchmarked against numerical simulations employing the finite volume method (FVM) and a PINN solution.
- We show that the stability of the learning algorithm can be improved using heat transport properties in the form of control loss functions.

This paper is organized in the following manner. Related work that serves as a foundation for this paper is presented in Sect. 2. Section 3 is a benchmark of the approach in terms of learning physical parameters and in comparison to other methods. This includes a FVM solution and a PINN solution. Section 4 describes the data integration. A summary and discussion is given in Sect. 5.

2 Related Work

2.1 Numerical Analysis and PINNs

Numerical analysis are indispensable in scientific and engineering research. These methods are particularly valuable when analytical solutions are either infeasible or non-existent due to the complexity of the underlying physics, such as the heat transfer in SLM. The state of the art for approximately solve PDEs are the finite difference method, FVM, and the finite element method. Mukherjee *et al.* presented a pioneering work for simulating heat transfer and fluid flow in AM using the FVM [15,16]. PINNs represent a recent innovation at the intersection of machine learning and numerical simulations. For SLM processes involving highly physical interactions, obtaining sufficient data to train accurate machine learning models can be prohibitively expensive, time-consuming, or even impossible. Traditional data-driven methods such as deep learning

often struggle with data scarcity because they rely on large data sets for effective generalization. PINNs offer an elegant solution to this problem by seamlessly incorporating domain-specific knowledge of heat transfer into neural network architectures. Raissi *et al.* has laid the foundation for this area of research and for much of the successful work of the past few years [17]. A comprehensive overview of PINNs in heat transfer problems is given by Cai *et al.* [3]. Uhrich *et al.* are given a heat forecasting on a built part in AM using simplified 2-dimensional PINN models [21]. A multi-model, physics-informed machine learning approach based on thermal and grayscale image analysis is presented by Bauer *et al.* [2].

2.2 Differential Equation Inspired Neural Networks

Deep Neural Networks are often seen as complex "black boxes," making it challenging to understand their inner workings, despite their primary goal of learning patterns in data. This lack of transparency makes it difficult to assess a neural network's suitability for a specific task, often requiring trial and error. Additionally, the unpredictability of trained models in real-world situations adds complexity, causing concerns about their stability, robustness, and reliability. For this purpose, Weinan was the first author to introduce the bridge between deep residual networks (Resnets) [10] and ODEs [7]. The use of ODE-inspired network design for single image super-resolution is given by He *et al.*. The authors proposed several network architectures based on Runge-Kutta methods [11]. An ODE transformer network is recently presented by Khoshsirat and Kambhamettu [13]. Ruthotto and Haber are given more specific DNN architectures that are motivated by PDEs. The foundation for the network design are the different classes of PDEs [18]. Alt *et al.* has also presented the transfer of the rich set of numerical foundations from PDEs to DNNs [1]. Shen *et al.* extend the explicit forward Euler method with the implicit backward counterpart and present a neural network for single image dehazing inspired by implicit Euler ODEs in their work [19]. Uhrich *et al.* showed how to predict valve failure using a neural network inspired by mathematical formulations of the operation of an electrodynamic valve [20].

2.3 Neural Diffusion Graph Networks

A novel approach that views deep learning on graphs as a continuous diffusion process and interprets Graph Neural Networks as discrete approximations of an underlying PDE is introduced by Chamberlain *et al.*. They addresses challenges commonly faced by graph learning models, such as depth, oversmoothing, and bottlenecks, while ensuring stability with respect to data perturbations in both implicit and explicit discretization schemes. The authors develop linear and nonlinear versions of GRAND, which demonstrate competitive performance on various standard graph benchmarks [4]. Two interesting applications of such approaches are climate modeling and stream water temperature prediction, presented by Choi *et al.* and Jia *et al.*, respectively [6,12]. We follow the work of Chamberlain *et al.*, which serves as a foundation for our approach.

3 Methodology and Benchmarking

3.1 Methodology NDGCN

A Graph data structure are a collection of vertices and edges $G = (V, E)$, $V = \{1, \ldots, N\}$, $E \subseteq V \times V$. Vertex distances d are described by the euclidean distance of vertex positions $p(v) \in \mathbb{R}^3$. For simplicity, a cubic mesh graph with equidistant vertices is used for benchmarking, as shown in Fig. 1. Heat diffusion processes in SLM can be described by the heat equation:

$$\dot{T}(x,t) = \frac{\kappa(T(x,t))}{c(T(x,t))\rho} \Delta T(x,t) + Q(x,t), \\ (x_1, x_2, x_3) \in \Omega, \quad t \in [0,T] \tag{1}$$

with the thermal conductivity κ, the specific heat c, the heat generated per volume Q and the density ρ. To approximate a given initial-boundary value problem on a graph model, a NDGCN is described in the following explicit scheme using forward time difference:

$$T_{n+1} = T_n + \delta t \frac{\kappa(T_n)}{c(T_n)\rho} L T_n + \delta t \cdot Q_n \tag{2}$$

The vertices of the graph represents temperature features $T : V \times [0, \tau) \to \mathbb{R}$ and the edges are weighted $\omega(v1, v2, t) : V \times V \times [0, \tau)$, where $w = \frac{\kappa}{\rho c}$. L represents the laplacian matrix, which is a discretized form of the laplace operator:

$$L = A - D \tag{3}$$

where A is the adjacency matrix, containing the connectivity between adjacent vertices, and D is the diagonal degree matrix, i.e. the matrix that has as the i-th diagonal element, the sum of the entries in the i-th row of A.

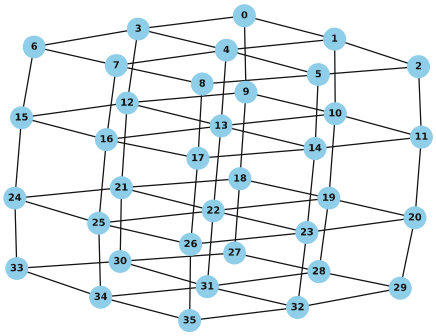

Fig. 1. Potential Cubic Mesh - Vertices and edges are equidistant

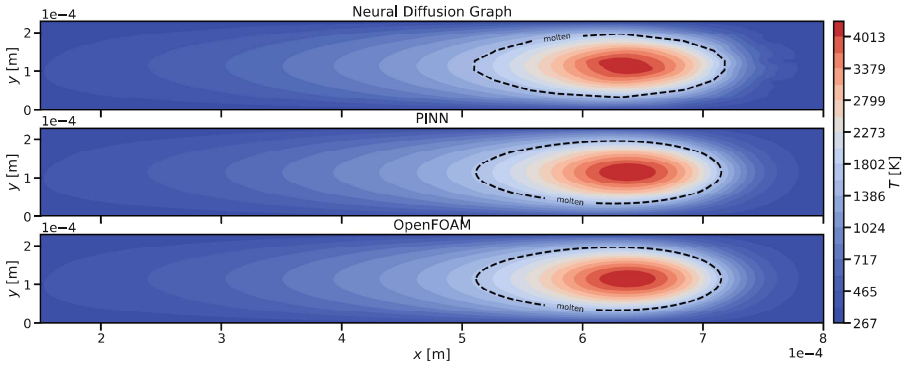

Fig. 2. Heat Transfer Prediction - The predicted solution of the NDGCN is compared with the FVM and a PINN solution as a baseline [22]. The NDGCN approach is able to reproduce the PINN and FVM solution

Due to the equal distribution of all vertices and edges, the trainable weights can be equated with the physical parameters $\kappa(T)$ and $c(T)$. The timesteps are equivalent to the number of layers in the NDGCN. The initial condition and Dirichlet boundary is formulized in the following manner:

$$T(t_0, x) = T_0, \quad (x_1, x_2, x_3) \in \Omega \tag{4}$$

$$T(t, x) = T_{\mathcal{B}}, \quad (x_1, x_2, x_3) \in \Omega, \quad t \in [0, T] \tag{5}$$

The Robin boundary condition is defined by the gaussian heat flux q of the energy source across the top surface:

$$\kappa \frac{\partial T}{\partial n} = q(x, t), \quad (x_1, x_2) \in \Omega_S \tag{6}$$

$$q = \frac{2AP}{\pi r_b^2} \exp\left(\frac{2(\mathbf{r}_0 - \mathbf{v}t)^2}{r_b^2}\right) \tag{7}$$

with $\mathbf{r_0} = (\frac{x_{1b}}{4}, \frac{x_{2b}}{2})$
where P is the laser power, x_{1b} and x_{2b} are the distance from the laser beam axis, r_b is the radius of the laser, and A is the absorption of the laser energy. v is the scanning velocity.

3.2 Benchmark

The performance of the approach to identify the physical parameters of 316L stainless steel is evaluated to benchmark the NDGCN model. The approach relies on utilizing the numerical solution of the boundaries as training data, in particular for the upper boundary. The training of the NDGCN model is guided by a loss function designed to incorporate Dirichlet boundary conditions, a crucial aspect in accurately modeling

Table 1. Thermal and mechanical properties

Properties	
Laser Power P (W)	250
Ambient Temperature T_0 (K)	293.15
Density ρ (kg/m^3)	$0.6 \cdot 7800$
Thermal conductivity κ (W/(m K))	$7.092 + 0.636 \times 10^{-2}$ T
Identified thermal conductivity κ (W/(m K))	$8.436 + 0.219 \times 10^{-2}$ T
Specific heat c (J/(kg K))	$330.9 + 0.563T - 4.015 \times 10^{-4}\mathrm{T}^2$ $+9.465 \times 10^{-8}\mathrm{T}^3$
Identified specific heat c (J(kg K))	$319.15 + 0.676\mathrm{T} - 2.285 \times 10^{-3}\mathrm{T}^2$
Absorption of the laser energy A	9×10^{-2}
Laser Beam Radius r (m)	1.4×10^{-4}

physical systems. This loss function serves as the driving force for training the model, ensuring it captures the essential physics of the problem:

$$\mathcal{L} = \frac{1}{N_v} \sum_{i=0}^{N_v - 1} (T_{\mathcal{B}}(t^i, x^i) - T_{pred}(t^i, x^i))^2 \tag{8}$$

where N_v are the number of vertices on the boundaries. Figure 2 visually demonstrates the efficacy of the NDGCN approach in reproducing the solutions obtained through FVM approximation and PINN solution. The successful reproduction of these solutions highlights the robustness and accuracy of the model. Figure 3 provides further insight into the capabilities of the NDGCN approach. It showcases that the model can predict heat transfer patterns within the constructed part without the need for additional data.

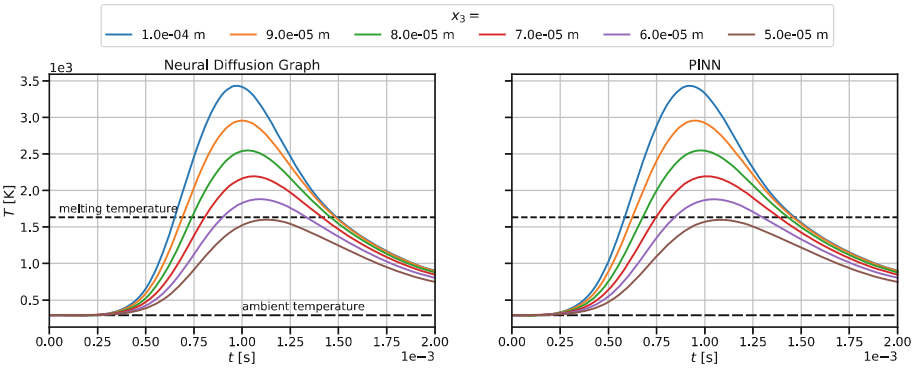

Fig. 3. Prediction of inside the built - The NDGCN model is able to predict the heat transfer in the inside of the built. The results are compared against the PINN solution

The trainable parameters of the NDGCN model are physically interpretable, as shown in Table 1. The trainable weights are modelled as parameters of the specific heat and thermal conductivity.

4 Data Integration

An arbitrary spatial structure can be approximately described using a simplicial 3-complex. For each layer in a printing job, the objective is to represent the partial object printed up to that point, by a simplicial complex, such that a) the shape of the complex closely resembles the shape of the part, and b) the dynamics of the heat distribution in the part can be modeled by a diffusion process on the underlying graph, i.e. the graph whose vertices and edges are given by the 0- and 1-simplices in the complex. Thermal images of the surface are taken periodically during the printing process. The shape of each layer of the part can be detected from these images. Stacking this information for all layers, we obtain a 3-dimensional model of the object.

4.1 Graph Construction from Thermal Images

We associate the pixels of a thermal image with points on a plane in $\mathbb{R}^3$, that is parallel to the (x, y)-plane and contains the current layer of the printing job. A point is assumed to be part of the printed object, if the temperature value of the associated pixel is above an empirical threshold. If this is the case, the point is added to a point cloud in $\mathbb{R}^3$, giving the pixel coordinates a third component for the height of the layer. A representative subset needs to be selected from the composite point cloud to build the simplicial complex. To this end, we make use of the pruning method described in [14], that iteratively removes the point with the highest scale-invariant density (SID).

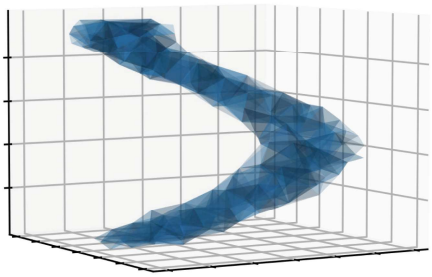

Fig. 4. Simplicial Complex - created from a helix-shaped point cloud, using SID-based pruning, Delaunay triangulation and restriction to the interior of the alpha shape

Since the data comes from $\mathbb{R}^3$ instead of a plane, we adapt the SID, by replacing the r-density with the 3-dimensional analogue:

$$d_r(v_i) = \frac{\#\{\|v_j - v_i\| < r \ : \ 1 \leq j \leq N, \ i \neq j\}}{\frac{4}{3}\pi r^3} \tag{9}$$

Which is the number of data points in the r-ball around v_i, divided by the volume of the ball. The scale-invariant density is defined as the integral over all r-densities:

$$d(v_i) = \int_0^\infty d_r(v_i)dr \tag{10}$$

and by a similar calculation as in [14], we see that:

$$d(v_i) = \left(\frac{8}{3}\pi\right)^{-1} \sum_{j \neq i} \|v_i - v_j\|^{-2} \tag{11}$$

In each step, the point with the highest spatial redundancy (measured by the SID) is removed. The resulting subset is relatively homogeneously distributed over the interior of the point cloud, but contains many boundary points, as their surrounding is partially void, resulting in a lower SID. This is desirable, as the boundary points define the shape of the printed object. For building the graphs representing the partially printed object, we iterate over the number of layers n, ranging from 1 to the total number N of layers in the printing job. For each n, the pruned set of points from the previous step is considered together with the points from the n-th layer. We restrict pruning to points from the top k layers ($k \ll N$), thus the representation for the first $n - k$ layers is inherited. A simplicial complex is constructed from the set of pruned points, using *Delaunay triangulation* [5]. The printed part must not necessarily be convex, but the shape produced by the Delaunay triangulation always is. To extract only the simplices that are within the boundary of the printed part, we use an *alpha shape* [8] to determine the hull of the point cloud. Removing the simplices that are not encased by the alpha shape, we end up with a simplicial 3-complex that resembles the shape of our printed object (see Fig. 4). The vertices and edges of this complex are used to define a graph.

The vertices are categorized based on spatial position:

1. vertices in the lowest layer are assigned to the **bottom boundary** class
2. vertices in the surface layer are assigned to the **top boundary** class
3. vertices that are part of a surface of the alpha shape, but neither in the top- nor bottom boundary are assigned to the **side boundary** class
4. vertices that are not part of either of these classes are assigned to the **interior** class

$C_i = C(v_i)$ denotes the class of the vertex $v_i,\ i = 1, \ldots N$.

4.2 Modelling Internal Heat Using Graph Diffusion

Due to non-equidistant vertices in the graph and a random laser trajectory in the 3D printing process, we propose a more complex model to predict the heat transfer based on real measurement data. Revisiting (1), we simplify notation by expressing the conductivity term $\frac{\kappa}{c\rho}$ as a single parameter α:

$$\dot{T}(x,t) = \alpha\big(T(x,t)\big)\Delta T(x,t) - Q\big(x, T(x,t)\big) \tag{12}$$

For a discrete approximation of the heat process on a graph, we replace the Laplacian Δ with the Graph Laplacian L (3).

$$\begin{aligned}\big(LT\big)(v_i,t) &= \sum_{v_j\sim v_i} a_{ij}T(v_j,t) - \Big(\sum_{v_j\sim v_i} a_{ij}\Big)T(v_i,t)\\ &= \sum_{v_j\sim v_i} a_{ij}\big(T(v_j,t)-T(v_i,t)\big)\end{aligned} \tag{13}$$

As a result, we obtain the diffusion equation on the graph:

$$\begin{aligned}\dot{T}(t) =&\alpha\big(T(v_i,t)\big)\big(LT\big)(t) - Q\big(v_i,T(v_i,t)\big)\\ =&\sum_{v_j\sim v_i}\Big(\alpha\big(T(v_i,t)\big)a_{ij}\Big(T(v_j,t)-T(v_i,t)\Big)\Big) - Q\big(v_i,T(v_i,t)\big)\end{aligned} \tag{14}$$

This equation can be rewritten using a state-dependent adjacency $A(T) = \Big(\alpha\big(T(v_i,t)\big)\cdot a_{ij}\Big)_{i,j}$, and $L(T)$ accordingly, to get:

$$\dot{T} = L(T)T - Q\big(T\big) \tag{15}$$

Here T is a vector of temperature values and Q is a vector-valued function.

Let $c_{ij}(T)$ denote the entries of $A(T)$. We assume that c_{ij} only depends on the temperature of the vertices it connects, i.e. $c_{ij}(T) = c_{ij}(T_i,T_j)$. Furthermore, c_{ij} should not depend on the specific vertices v_i, v_j, but only on their local properties, i.e.

1. their distance $\varrho_{ij} = \|v_i - v_j\|_2$
2. the vertex classes $C(v_i)$, $C(v_j)$
3. the scale-invariant densities $d(v_i)$, $d(v_j)$

Under these assumptions, there is some function φ, modelling the connectivity of adjacent nodes:

$$c_{ij}(T) = \varphi(\varrho_{ij},T_i,T_j,C_i,C_j,d_i,d_j) \tag{16}$$

In a similar fashion, the i-th entry of the dissipation vector $Q(T) = \big(Q_i(T)\big)_i$ should only depend on the local properties of v_i, i.e. T_i, C_i, d_i. This implies that there is some function ψ, s.t.:

$$Q(T) = \psi(T_i,C_i,d_i) \tag{17}$$

For the neural diffusion model, the functions $\varphi,\ \psi$ are approximated by single-hidden-layer networks, and $L,\ Q$ are composed by the evaluations of these models.

We propose a NDGCN model for predicting the temperature change over discrete time-steps for arbitrary graphs, defined by the equation:

$$T_{n+1} = T_n + \left(\sum_{j=1}^{k}\Theta_j\big(\delta t L(T_n)\big)^j\Big(T_n - \frac{\delta t}{2}Q(T_n)\Big)\right) - \delta t\cdot Q(T_n) \tag{18}$$

This model contains the trainable submodels φ, ψ determining L and Q, as well as the trainable parameters Θ_j, which define a graph convolution. The model Eq. (18) is motivated by the Taylor series for a solution of the homogeneous heat equation:

$$\begin{aligned} T(t+\delta t) &= \sum_{j=0}^{\infty} \frac{1}{j!} \left(\left(\frac{\partial}{\partial t} \right)^j T \right) (t) (\delta t)^j \\ &= T(t) + \sum_{j=1}^{\infty} \frac{1}{j!} \left(\left(\frac{\partial}{\partial t} \right)^j T \right) (t) (\delta t)^j \end{aligned} \tag{19}$$

Note that for a solution of the homogeneous heat equation, its derivative w.r.t. t is again a solution of the heat equation, since $0 = \frac{\partial}{\partial t}\left(\dot{T} - \alpha \Delta T\right) = \ddot{T} - \alpha \Delta \dot{T}$. Hence, we can replace $\left(\frac{\partial}{\partial t}\right)^j$ by $(\alpha \Delta)^j$ in the Taylor series. Approximating $\alpha \Delta$ by $L(T)$, we obtain:

$$T(t+\delta t) \approx T(t) + \sum_{j=1}^{\infty} \frac{1}{j!} \left(L(T)^j T \right) \left(t \right) (\delta t)^j \tag{20}$$

The model Eq. (18) is derived from this by evaluating the first k terms of the series, accounting for the inhomogeneity due to dissipation, and relaxing the fixed coefficients $1/j!$ to trainable parameters.

Training this kind of model entails several practical challenges: First, using the model to predict changes in surface temperature, requires knowing the initial state T_0 for all vertices, including those that cannot be observed. Additionally, the limited observability of vertices likely leads to an underdetermined optimization problem, which raises doubts about the model's ability to accurately represent the internal heat state. Finally, the model's locality means that vertices beyond a certain distance from the surface (namely, k steps in the graph) do not directly influence the observations, and thus are not tracked by the learning algorithm.

To address the first problem, the initial state T_0 is constructed by iterative model predictions for each layer, starting from the first layer, where all vertices can be observed. However, this requires a relatively well-trained model in the first place. To resolve this circular dependency, we start by training the model on just the first two layers, and then gradually increase the number of layers when the model has learned a good prediction for the current depth. For controlling the other problems, we rely on regularizing loss functions, that are motivated by known properties of heat conduction and enforce those properties on the model predictions.

First, we ensure that the relative magnitude of the convolution parameters is consistent with the derivation from the Taylor series, using the loss term:

$$\mathcal{L}_{\Theta} = \sum_{j=1}^{k} \left(\frac{\Theta_j j!}{\Theta_1} - 1 \right)^2 \tag{21}$$

For the connectivity model φ, it is known from the discretization of the continuous Laplace, that the connectivity of two adjacent vertices should be proportional to the inverse square of their distance, ϱ_{ij}^{-2}. Assuming $\varphi(\varrho_{ij}) = \tau \varrho_{ij}^{-2}$ (keeping all other

parameters constant), it follows $\varphi'(\varrho_{ij}) = -2\tau\varrho_{ij}^{-3}$, and thus $\frac{\varphi'(\varrho_{ij})}{\varphi(\varrho_{ij})} = -2\varrho_{ij}^{-1}$, which is independent of the scale τ. Therefore, the second regularizing loss term is given by:

$$\mathcal{L}_{\varphi} = \sum_{i,j:i\sim j} \left(\frac{\varphi'(\varrho_{ij})}{\varphi(\varrho_{ij})} - \left(-2\varrho_{ij}^{-1} \right) \right)^2 \tag{22}$$

which ensures consistency of the edge weights with the distances of the connected vertices. Furthermore, dissipation can only occur at the boundary, so $\psi = 0$ is required for interior points, motivating the loss:

$$\mathcal{L}_{\psi} = \sum_{i:\ C_i=\text{int.}} \psi(T_i, C_i, d_i)^2 \tag{23}$$

Using knowledge from the theoretical study of PDEs, it is also possible to make statements about the temporal evolution of the heat state. First, the total thermal energy in the body can only change because of dissipation, thus $\sum_i T_{n+1}(i) = \sum_i \Big(T_n(i) - Q_n(i) \Big)$. Therefore, a loss function is proposed:

$$\mathcal{L}_{\text{heat}} = \left(\sum_i \Big(T_{n+1}(i) - T_n(i) + Q_n(i) \Big) \right)^2 \tag{24}$$

Another well known property of the evolution of heat distribution is the maximum principle (see for example [9], §2.3.). For our purpose, it suggests that the temperature at a vertex is within the range given by the minimum and maximum over its previous temperature and the temperatures of connected vertices. This is expressed by the regularizing loss terms:

$$\mathcal{L}_{\text{max}} = \sum_i \max\Big(0, T_{n+1}(i) - \max(M)\Big)^2 \tag{25}$$

$$\mathcal{L}_{\text{min}} = \sum_i \max\Big(0, \min(M) - T_{n+1}(i)\Big)^2 \tag{26}$$

$$M = \big\{ T_n(i);\ T_{n+1}(j),\ j \sim i \big\}$$

Furthermore, a potential energy for the heat distribution can be defined. It is given by:

$$E(T,t) = \int_U \Big(T(x,t) - \overline{T}(t) \Big)^2 dx \tag{27}$$

For the time-differential of this energy, one can compute:

$$\begin{aligned} \dot{E}(T,t) =& 2 \int_U \Big(T(x,t) - \overline{T}(t) \Big) \dot{T}(x,t) dx \\ =& 2\alpha \int_U \Big(T(x,t) - \overline{T}(t) \Big) \Delta T(x,t) dx \\ =& 2\alpha \Bigg(\underbrace{\int_{\partial U} \Big(T(x,t) - \overline{T}(t) \Big) \big(\nabla T(x,t) \cdot \nu \big) dS}_{\text{energy from dissipation}} - \underbrace{\int_U |\nabla T(x,t)|^2 dx}_{\geq 0} \Bigg) \end{aligned} \tag{28}$$

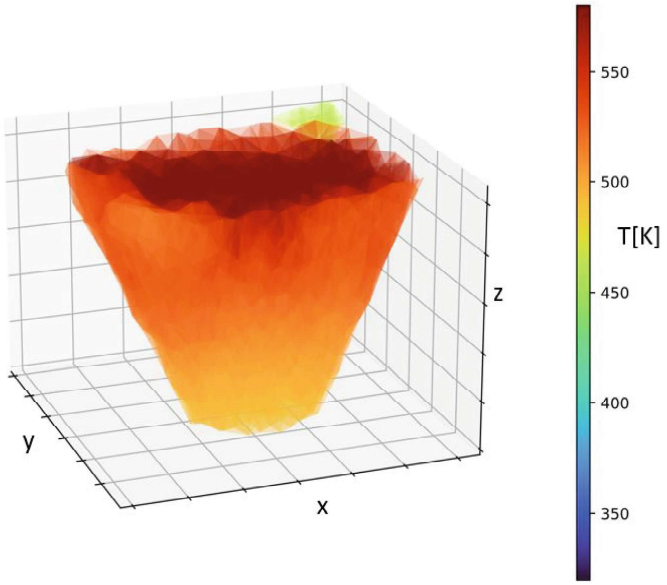

Fig. 5. Predicting heat transfer of a 350-layer printed inverted pyramid - Heat transfer can be predicted over a long period of time by integrating real measured data into a graph model

Assuming dissipation is relatively small, it should roughly hold that $\dot{E} \leq 0$, so $E(T_{n+1}) \leq E(T_n)$. Thus, we introduce another loss term:

$$\mathcal{L}_{\text{energy}} = \max\Big(0, E(T_{n+1}) - E(T_n)\Big) \tag{29}$$

For training the model, the regularizing loss functions $\mathcal{L}_\Theta, \mathcal{L}_\varphi, \mathcal{L}_\psi, \mathcal{L}_{\text{heat}}, \mathcal{L}_{\text{min}}, \mathcal{L}_{\text{max}}$, and $\mathcal{L}_{\text{energy}}$, as well as the prediction loss

$$\mathcal{L}_{\text{data}} = \sum_{i:\ C_i=\text{top}} \Big(T_{n+1}(i) - T_{n+1}^{(data)}(i)\Big)^2 \tag{30}$$

are added using appropriate weights, and the descent algorithm seeks to find a minimum for the sum, i.e. a model state that fits the data without violating known properties of heat diffusion. This restricts the admissible set of solutions, thus mitigating the problem of underdetermination, and the regularizing loss functions also regard vertices that are disconnected from the surface and therefore not captured by $\mathcal{L}_{\text{data}}$.

4.3 Results

The model is restricted for learning a consistent diffusion process, both with the laws of physics and the observations.

To generate the initial states for the training steps, the model training starts only for the first layer (where we can observe the complete state), and then gradually increase the number of layers used for training.

The model is trained on the first 100 layers of a print job for an inverted pyramid frustum to evaluate predictions for all 500 layers. The optimisation was performed with the ADAM optimizer. We used a learning rate of $\eta = 1 \times 10^{-5}$ and decay rates $\beta_1 = 0.5$ and $\beta_2 = 0.99$, which estimate the first and second moments of the gradient less than usual. Our model architecture contains the same number of layers as the thermal images are generated. The frame rate has been limited to 3 Hz in order to be in control of the amount of data. The internal state of the model after printing up to 350 layers is shown in Fig. 5. For now, it is not possible to verify these predictions, as the lower layers cannot be observed. However, the distributions seem to be consistent with the heat process in an object.

5 Conclusion and Discussion

In this paper, we have proposed an NDGCN approach for predicting heat transfer in SLM processes. First, we have given a benchmark for our model to reproduce synthetic data and discover physically interpretable parameters modeled as trainable weights of the network. This foundational step underscores the potential of our approach in capturing underlying physical processes. In the second part, a generated graph data structure is used to integrate thermal imaging data. In addition, we extend our model with several regularization principles to predict heat transfer using real measurement data. While this work represents an essential proof of concept with promising initial results, several avenues for future research and refinement are evident. First, the second part of the paper lacks a comprehensive evaluation of our results in comparison to baseline models, including other machine learning approaches. This comparative analysis will provide valuable insights into the effectiveness of our NDGCN in relation to existing methodologies. Additionally, we anticipate expanding our dataset to encompass a broader range of built geometries and scenarios, thereby enhancing the robustness and applicability of our model. This broader dataset will not only facilitate a more thorough evaluation but also enable us to explore diverse industrial applications. Furthermore, we acknowledge the importance of addressing the computational complexity of our NDGCN model relative to other techniques like PINNs or numerical analysis. Such an evaluation will help in understanding the computational trade-offs associated with our approach and guide its optimization for practical use cases.

Acknowledgments. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft Kultur und Tourismus in the program Center of Excellence for AI-research “Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig”, project identification number: ScaDS.AI.

References

1. Alt, T., Schrader, K., Augustin, M., Peter, P., Weickert, J.: Connections between numerical algorithms for PDEs and neural networks. J. Math. Imaging Vision **65**(1), 185–208 (2023). https://doi.org/10.1007/s10851-022-01106-x
2. Bauer, M., Uhrich, B., Schäfer, M., Theile, O., Augenstein, C., Rahm, E.: Multi-modal artificial intelligence in additive manufacturing: combining thermal and camera images for 3D-print quality monitoring. In: Proceedings of the 25th International Conference on Enterprise Information Systems, pp. 539–546. SCITEPRESS - Science and Technology Publications (2023). https://doi.org/10.5220/0011967500003467
3. Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks for heat transfer problems. J. Heat Transfer **143**(6) (2021). https://doi.org/10.1115/1.4050542
4. Chamberlain, B., Rowbottom, J., Gorinova, M.I., Bronstein, M., Webb, S., Rossi, E.: Grand: graph neural diffusion. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 1407–1418. PMLR (2021). https://proceedings.mlr.press/v139/chamberlain21a.html
5. Chen, L., Xu, J.C.: Optimal delaunay triangulations. J. Comput. Math. 299–308 (2004)
6. Choi, H., Choi, J., Hwang, J., Lee, K., Lee, D., Park, N.: Climate modeling with neural advection-diffusion equation. Knowl. Inf. Syst. **65**(6), 2403–2427 (2023). https://doi.org/10.1007/s10115-023-01829-2
7. E, W.: A proposal on machine learning via dynamical systems. Commun. Math. Stat. **5**(1), 1–11 (2017). https://doi.org/10.1007/s40304-017-0103-z
8. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Trans. Graph. (TOG) **13**(1), 43–72 (1994)
9. Evans, L.C.: Partial Differential Equations, vol. 19. American Mathematical Society (2022)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
11. He, X., Mo, Z., Wang, P., Liu, Y., Yang, M., Cheng, J.: Ode-inspired network design for single image super-resolution. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1732–1741. IEEE (2019). https://doi.org/10.1109/CVPR.2019.00183
12. Jia, X., Chen, S., Zheng, C., Xie, Y., Jiang, Z., Kalanat, N.: Physics-guided graph diffusion network for combining heterogeneous simulated data: an application in predicting stream water temperature. In: Shekhar, S., Zhou, Z.H., Chiang, Y.Y., Stiglic, G. (eds.) Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 361–369. Society for Industrial and Applied Mathematics, Philadelphia (2023). https://doi.org/10.1137/1.9781611977653.ch41
13. Khoshsirat, S., Kambhamettu, C.: A transformer-based neural ode for dense prediction. Mach. Vis. Appl. **34**(6) (2023). https://doi.org/10.1007/s00138-023-01465-4
14. Kurz, G., Holoch, M., Biber, P.: Geometry-based graph pruning for lifelong slam. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3313–3320. IEEE (2021)
15. Mukherjee, T., Wei, H.L., De, A., DebRoy, T.: Heat and fluid flow in additive manufacturing – part ii: powder bed fusion of stainless steel, and titanium, nickel and aluminum base alloys. Comput. Mater. Sci. **150**, 369–380 (2018). https://doi.org/10.1016/j.commatsci.2018.04.027
16. Mukherjee, T., Wei, H.L., De, A., DebRoy, T.: Heat and fluid flow in additive manufacturing–part i: modeling of powder bed fusion. Comput. Mater. Sci. **150**, 304–313 (2018). https://doi.org/10.1016/j.commatsci.2018.04.022

17. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019). https://doi.org/10.1016/j.jcp.2018.10.045
18. Ruthotto, L., Haber, E.: Deep neural networks motivated by partial differential equations. J. Math. Imaging Vision **62**(3), 352–364 (2020). https://doi.org/10.1007/s10851-019-00903-1
19. Shen, J., Li, Z., Yu, L., Xia, G.S., Yang, W.: Implicit euler ode networks for single-image dehazing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 877–886. IEEE (2020). https://doi.org/10.1109/CVPRW50498.2020.00117
20. Uhrich, B., Hlubek, N., Häntschel, T., Rahm, E.: Using differential equation inspired machine learning for valve faults prediction. In: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), pp. 1–8. IEEE (2023). https://doi.org/10.1109/INDIN51400.2023.10217897
21. Uhrich, B., Schäfer, M., Theile, O., Rahm, E.: Using physics-informed machine learning to optimize 3D printing processes. In: Correia Vasco, J.O., et al. (eds.) ProDPM 2021. Springer Tracts in Additive Manufacturing, pp. 206–221. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-33890-8_18
22. Uhrich, B., Pfeifer, N., Schäfer, M., et al.: Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing. Appl. Intell. **54**(6), 4736–4755 (2024). https://doi.org/10.1007/s10489-024-05402-4. Springer