



Tool path optimization of selective laser sintering processes using deep learning

D. H. Kim¹ · T. I. Zohdi¹

Received: 10 May 2021 / Accepted: 27 July 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Advancements in additive manufacturing (3D printing) have enabled researchers to create complex structures, offering a new class of materials that can surpass their individual constituent properties. Selective laser sintering (SLS) is one of the most popular additive manufacturing techniques and uses laser power to bond powdered material into intricate structures. It is one of the fastest additive manufacturing processes for printing functional, durable prototypes, or end-user parts. It is also widely used in many industries, due to its ability to easily make complex geometries with little to no additional manufacturing effort. In the SLS process, tool path selection is important because it is directly related to the integrity of a 3D printed structure. In this research, we focus on how to obtain an optimal tool path for the SLS process from a numerical simulation. Also, we apply a deep learning technique to accelerate the simulation of the SLS processes, while obtaining accurate numerical results.

Keywords Additive manufacturing · Deep learning · Optimization · Machine learning · Simulation · 3D printing

1 Introduction

Selective laser sintering (SLS) is an additive manufacturing technique that uses a laser as a power source to sinter powdered material,¹ by directing the laser automatically at points in space, dictated by a 3D model. The original SLS process was invented in 1986 at the University of Texas at Austin, with increasing applications as the technology matures [2]. Many researchers have proposed various ways to simulate the SLS process, to ascertain the mechanical/thermal behavior of the sintered material [3]. Dong et al. [4] developed a transient three-dimensional finite element (FE) model to simulate the temperature evolution during the SLS process. Kolossov et al. [5] created a three-dimensional finite element model for SLS processes, considering the non-linear behavior of thermal conductivity and specific heat due to temperature changes and phase transformations. Matsumoto et al. [6] proposed an FE model for calculating the temperature

and stress distribution in a single layer of metallic powder in selective laser processing. Simchi [7] and Simchi and Pohl [8] used experimental results to observe microstructural evolution and densification during laser sintering of metal powders. Gusarov and Kruth [9] employed a radiation transfer model to calculate absorptances and deposited energy profiles while processing thin layers of metallic powder, and provided an analytical equation of laser penetration as a function of particle size and powder bed density. This work was followed by a finite difference simulation of heat transfer during selective laser melting [10].² There also have been discrete element models, which are a good option for simulation of additive manufacturing processes, for modeling and simulation of the laser processing of the powdered particles [11–15]. That work developed a coupled discrete element-finite difference model of SLS process [16].

As mentioned, tool path selection in additive manufacturing is important because it is directly related to the durability of the 3D printed structure [17]. This research aims to obtain an optimal tool path for the SLS process from numerical simulations. During laser processing, temperature gradients

¹ PA12 is a widely used thermoplastic material for selective laser sintering which is suitable for many applications [1].

✉ D. H. Kim
donghoon_kim@berkeley.edu

¹ Department of Mechanical Engineering, 6195 Etcheverry Hall, University of California, Berkeley, CA 94720-1740, USA

² Selective laser sintering (SLS) processes heat the powdered material below its melting point so that the powder can fuse together at a particle level. In selective laser melting (SLM), however, the powdered material is not merely fused together but is fully melted. While the SLM normally works with metals, the SLS generally works with plastics and ceramics.

occur within a 3D printed structure, which can cause unintended residual stresses. This is one of the major reasons for premature failure in parts printed by additive manufacturing. In order to find an optimal laser path which minimizes temperature gradients among a variety of tool paths, we first need to solve a dynamic programming problem to find all the possible laser paths for the laser grid configuration of a given geometry. After obtaining all possible laser paths, one can use numerical methods, such as the finite difference method, the discrete element method, the finite element method, etc., to calculate the thermal gradients or residual stresses of a printed geometry.

However, when the simulation system becomes excessively large, it can become computationally expensive to perform exhaustive simulations to calculate the temperature gradients for large numbers of possible paths. Accordingly, in order to efficiently obtain an optimal laser path for SLS processes, we apply Deep Learning techniques, which allow computers to learn and detect patterns from noisy/complex data sets, and then to extract discovered patterns to make predictions for future unknown data. There have been some previous studies that used deep learning to solve partial differential equations (PDEs) or to develop deep learning-based reduced order models (ROMs). [18–20] propose deep learning-based numerical methods for solving PDEs. Also, [21,22] propose deep learning-based ROMs, having the potential to simulate physical and dynamic systems with increased computational efficiency while maintaining reasonable accuracy. Applying the deep learning technique, we accelerate the numerical simulations to obtain the optimal laser path with high accuracy, as well as reduced computational costs. In order to efficiently process multi-dimensional data, we employ Convolutional neural networks (CNNs) to solve this manufacturing problem.

The main objective of this work is to develop a deep learning model to efficiently predict the optimal tool path of the SLS process, which minimizes the average thermal gradient. In order to achieve this, we first obtain all the possible laser paths needed to print the objective geometry. We then solve a dynamic programming problem (which will be explained further in the next section). After that, we model and simulate the SLS process in order to calculate the average thermal gradient. Finally, based on the obtained simulation data, we construct a deep learning model to predict the optimal laser path for the SLS process. The overall flowchart is shown in Fig. 1 and Algorithm 1.

2 Tool path generation using dynamic programming

In this section, we will focus on finding all the possible laser paths within the discretized domain. Examples of the possible

Algorithm 1: Overall Algorithm

STEP 1: Split all the laser paths (obtained by dynamic programming) into the *training data* and the *test data*.

STEP 2: Finite difference simulations
foreach $path \in training\ data$ **do**
 | Finite difference simulation to obtain the thermal gradients.
end

STEP 3: Sort the paths in training data based on the finite difference simulation results.

STEP 4: Label the top half ‘Good’, and the bottom half ‘bad’.

STEP 5: Convert all the paths into gray-scale images (preprocessing for the deep learning).

STEP 6: Train the deep learning predictive model with the gray-scale images and corresponding labels of the training data.

STEP 7: Predict the rankings of all the data (the *training data* and the *test data*).

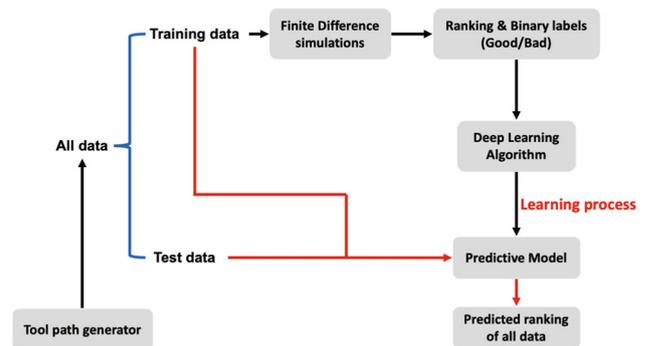


Fig. 1 The overall algorithm flowchart

laser paths for a simple 4 by 4 grid is shown in Fig. 2. After discretizing the domain into a structured grid, we can set the starting point of the laser from any of the nodes in the grid. In order to simplify the problem, we set the following restrictions on the laser path:

- The laser should visit all the nodes in the grid.
- The laser should visit each node only once.
- The laser cannot jump over nodes.

The main problem we encounter is that, as the laser moves from node to node, every single path obtained so far branches off to the new points, as shown in Fig. 3. For example, if we have connected the points (0, 4, 8, 12, 13, 14, 10) so far, then there are three possible next points 9,6, and 11. Therefore, (0, 4, 8, 12, 13, 14, 10) branches off to (0, 4, 8, 12, 13, 14, 10, 9), (0, 4, 8, 12, 13, 14, 10, 6), and (0, 4, 8, 12, 13, 14, 10, 11) in the next step. We have to accumulate the cases to search for all the possible tool paths.

For this reason, we apply a dynamic programming technique to efficiently solve for the laser path. Dynamic

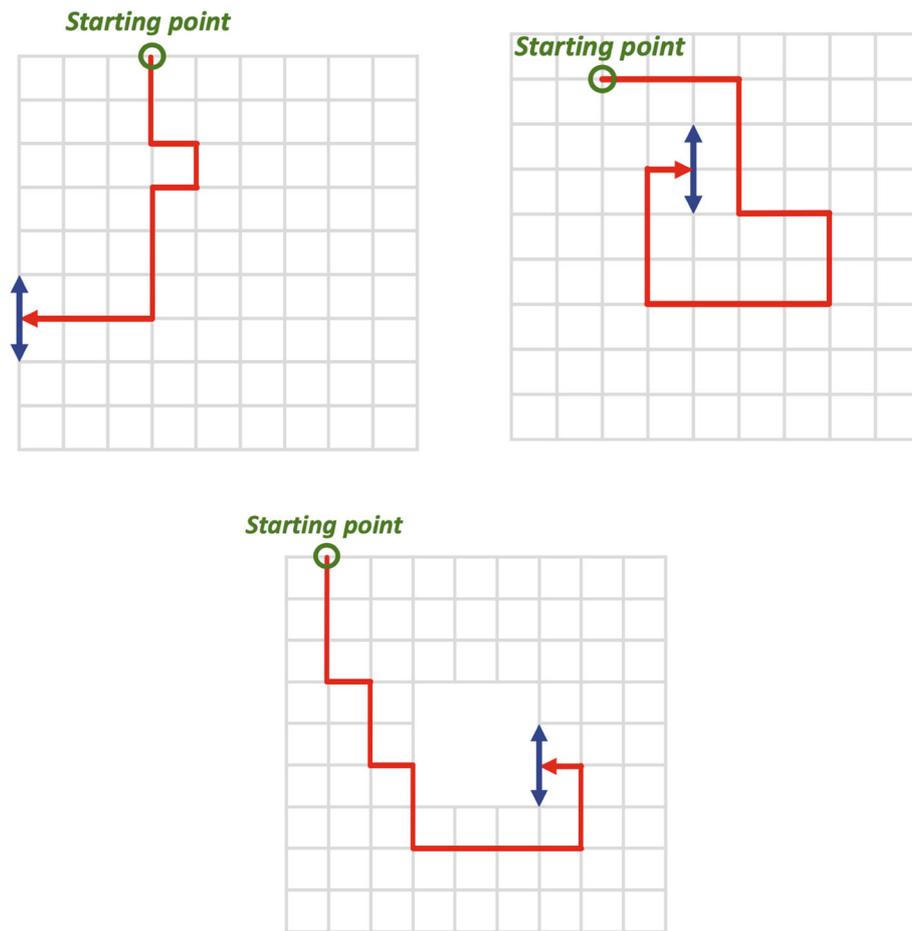


Fig. 5 Failed paths to be removed from the candidate path set

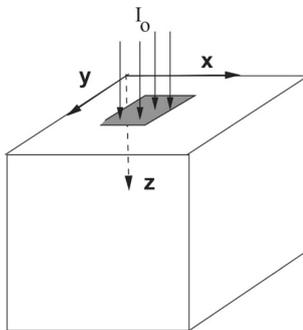


Fig. 6 A schematic and the coordinate system of the laser processing

grid points are the points where numerical simulations are actually performed to evaluate the thermal gradients. The simulation parameters are shown in Table 1.

The governing equation is as follows:

$$\rho C \frac{\partial \theta}{\partial t} = \nabla \cdot (K \nabla \theta) + I_{abs}, \quad (3.1)$$

where

- θ : Temperature (K)

- ρ : Density (kg/m^3)
- C : Specific heat capacity ($\text{J}/(\text{kgK})$)
- K : Thermal conductivity (W/mK)
- I_{abs} : Absorbed laser energy within the laser zone (W/m^3)

Assuming that we have a constant thermal conductivity k for simplicity, the boundary conditions are as follows:

- Top surface: Convection

$$-k \frac{d\theta}{dz} = h(\theta_\infty - \theta) \quad (3.2)$$

- Other surfaces: Adiabatic

$$q_s = -k \frac{d\theta}{dx} = 0 \quad (3.3)$$

$$q_s = -k \frac{d\theta}{dy} = 0 \quad (3.4)$$

$$q_s = -k \frac{d\theta}{dz} = 0, \quad (3.5)$$

where θ_∞ is the ambient temperature (K), and q_s is heat flux at the surfaces (W/m^2). As for energy absorption I_{abs} , we

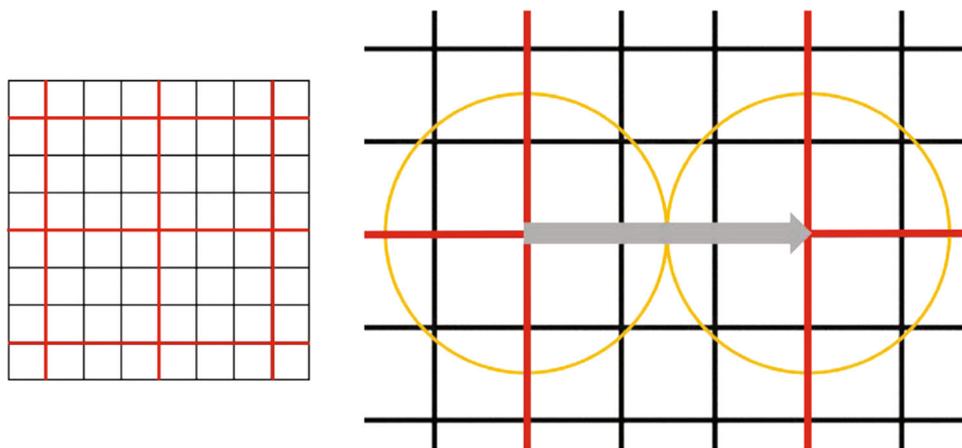


Fig. 7 Grid configuration (RED: the laser grid, BLACK: the material grid). (Color figure online)

Algorithm 2: Dynamic Programming

Initial step: Obtain the starting points of the laser, based on the symmetry of the laser grid.

```

foreach initial_point ∈ starting_points do
    all_the_paths_so_far = [initial_point]
    count = 0
    while count < the_number_of_all_the_laser_grid_points do
        next_paths = []
        foreach path ∈ all_the_paths_so_far do
            STEP 1: Obtain next_possible_points for path:
            NEITHER Already passed points NOR Self-collision
            NOR
            Interval between boundary flags.

            STEP 2:
            if next_possible_points is NOT empty then
                foreach pt ∈ next_possible_points do
                    Append pt to path
                    Append path to next_paths
                end
            end
        end
        all_the_paths_so_far = next_paths
        count = count + 1
    end
    Store all_the_paths_so_far as laser paths of the starting
    point (initial_point).
end
    
```

Table 1 Simulation parameters

Symbol	Units	Value	Description
θ_∞	K	300	Ambient temperature
θ_0	K	300	Initial temperature
h	W/m ² K	10	Convection coefficient
k	W/(mK)	0.22	Thermal conductivity of PA12
ρ	kg/m ³	1100	Density of PA12
C	J/(kgK)	1590	Specific heat capacity of PA12
w	m	0.0025	Laser radius
v	mm/s	10	Laser scanning speed
P	W	200	Power term in I_{abs}
β	1/m	80	Optical extinction coefficient
ΔL	m	0.00167	Material grid gap size

with

$$I_0 = \frac{2P}{\pi w^2}, \tag{3.7}$$

where β is an optical extinction coefficient [27], P is the power of the laser, and w is the laser radius. According to Gusarov et al. [10], the extinction coefficient β is dependent on the size and the morphology of micro-scale particles in the micrograph. This could be determined by experimental work as shown in [28]. In the numerical analysis we performed, we used the finite difference method using a constant optical extinction coefficient to calculate the temperature values.³ Also, we can rewrite Eq. 3.1 as follows:

use the Beer-Lambert penetration model for a Gaussian laser [16], where

$$I_{abs}(r, z) = I_0 e^{-\beta z} e^{-\frac{r^2}{w^2}} \tag{3.6}$$

$$\rho C \frac{\partial \theta}{\partial t} = k \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + \frac{\partial^2 \theta}{\partial z^2} \right) + I_{abs}(r, z). \tag{3.8}$$

³ One could use a micro-scale particle-based model to include the expression for the estimated extinction coefficient.

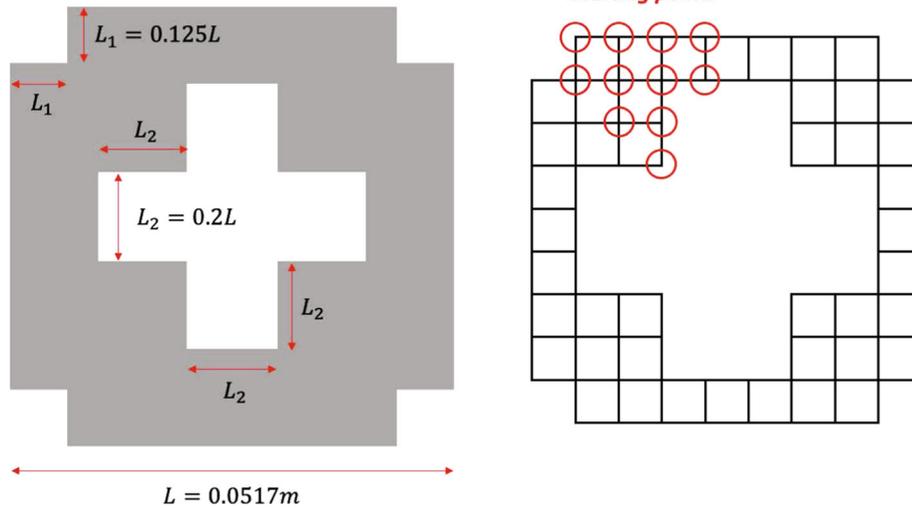


Fig. 8 LEFT: example geometry, RIGHT: corresponding laser grid

When we discretize Eq. 3.8 using finite difference method with a forward Euler scheme for time integration, we obtain:

$$\theta(t + \Delta t, x, y, z) = \theta(t, x, y, z) + \Delta t \left[\frac{k}{\rho C} (A_x + A_y + A_z) + \frac{1}{\rho C} I_{abs}(r, z) \right], \quad (3.9)$$

where

$$A_x = \frac{\partial^2 \theta}{\partial x^2} = \frac{\theta(t, x + \Delta x, y, z) - 2\theta(t, x, y, z) + \theta(t, x - \Delta x, y, z)}{\Delta x^2} \quad (3.10)$$

and

$$A_y = \frac{\partial^2 \theta}{\partial y^2} = \frac{\theta(t, x, y + \Delta y, z) - 2\theta(t, x, y, z) + \theta(t, x, y - \Delta y, z)}{\Delta y^2} \quad (3.11)$$

and

$$A_z = \frac{\partial^2 \theta}{\partial z^2} = \frac{\theta(t, x, y, z + \Delta z) - 2\theta(t, x, y, z) + \theta(t, x, y, z - \Delta z)}{\Delta z^2}. \quad (3.12)$$

The convective boundary condition [3.2] on the top surface is discretized as follows

$$h(\theta_\infty - \theta) + k \frac{\theta_1 - \theta_0}{\Delta z} = 0, \quad (3.13)$$

where θ_1 is the temperature at the outermost inner node, which is adjacent to the boundary node, and θ_0 is the temperature at the boundary node.

After the laser visits all the nodes in the laser grid, we have the final temperature values for the material grid nodes throughout the domain that are obtained by solving the Eq. [3.1] numerically.

Using the final temperature values, we could calculate the temperature gradient values throughout the inner nodes of the material grid, taking a central difference from the adjacent nodes. Each inner node has its temperature gradient values for x , y , and z directions. After that, we could get the average thermal gradient throughout the domain of the 3D printed layer, as shown in Equation [3.14].

Average thermal gradient

$$= \frac{1}{N} \sum_{i \in Z} \sqrt{\left(\frac{\partial \theta}{\partial x} \right)_i^2 + \left(\frac{\partial \theta}{\partial y} \right)_i^2 + \left(\frac{\partial \theta}{\partial z} \right)_i^2}, \quad (3.14)$$

where Z is a printing zone sintered by laser, and N is the number of the inner nodes in the material grid included in the printing zone. The average temperature gradient values for the laser paths are then used for the labeling of the training/test data in Sect. 6.

4 Simulation process of the SLS process

We implemented the simulation of the SLS process to print the example geometry shown in Fig. 8. We considered two example laser paths, even though there could be numerous possible laser paths for this geometry and the corresponding laser grid. The possible starting points on the laser grid,

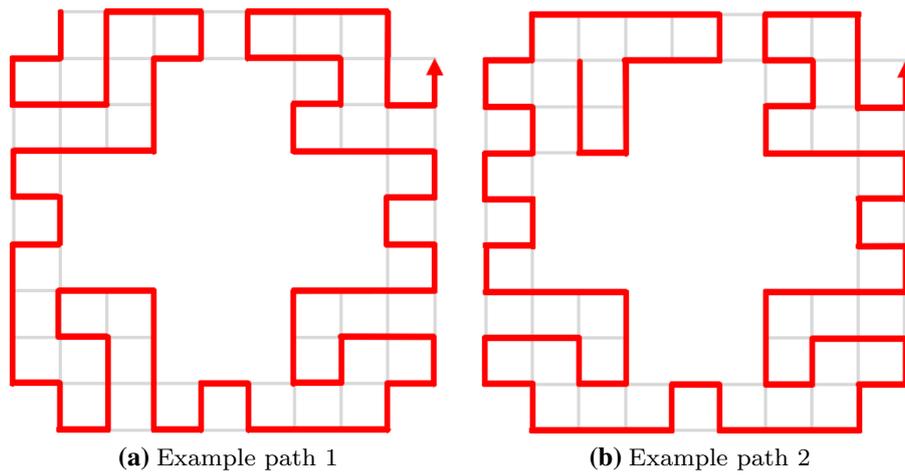


Fig. 9 Examples of the possible laser paths for the given geometry

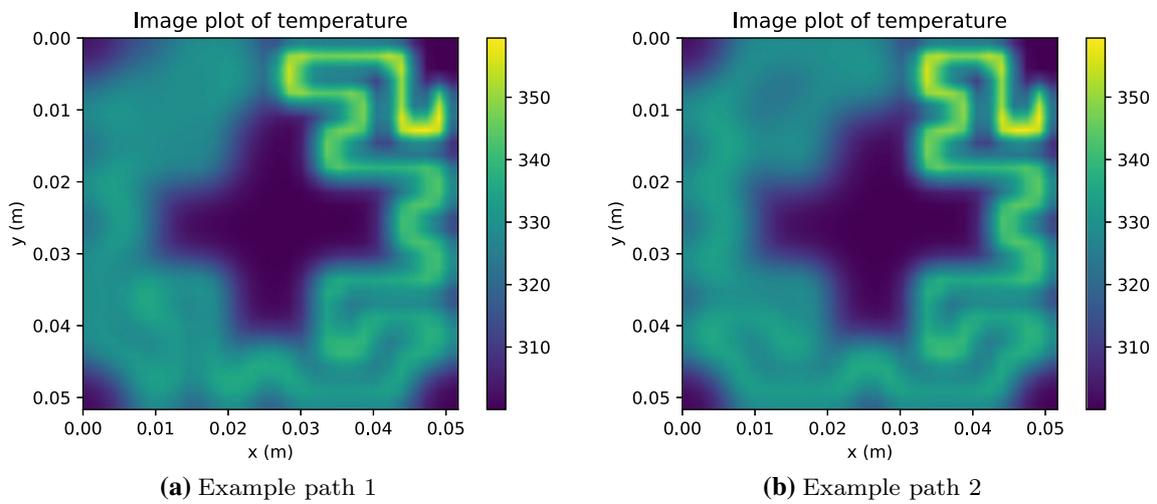


Fig. 10 Temperature plots of the top surfaces at the final processing time (37.5 (s))

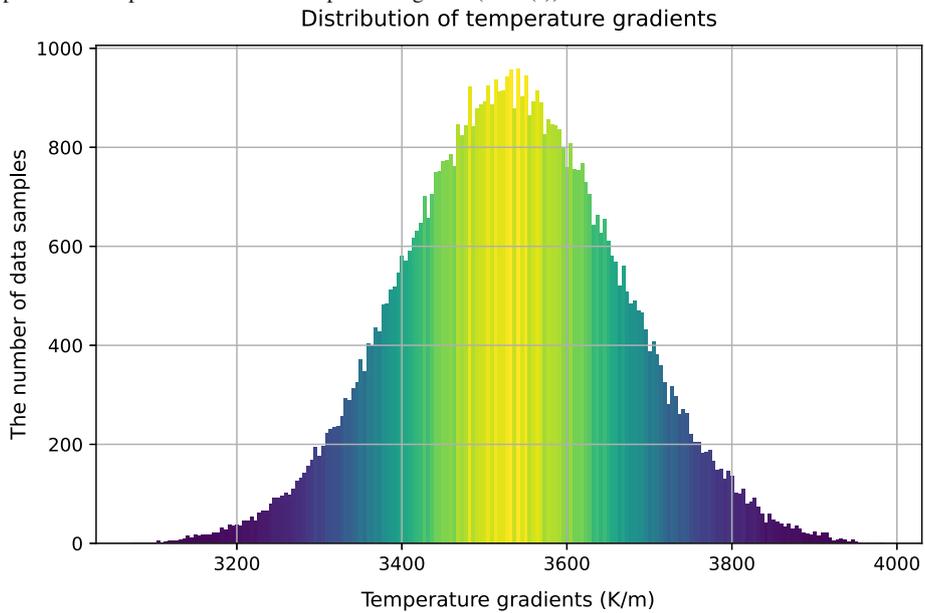


Fig. 11 Distribution of the temperature gradients

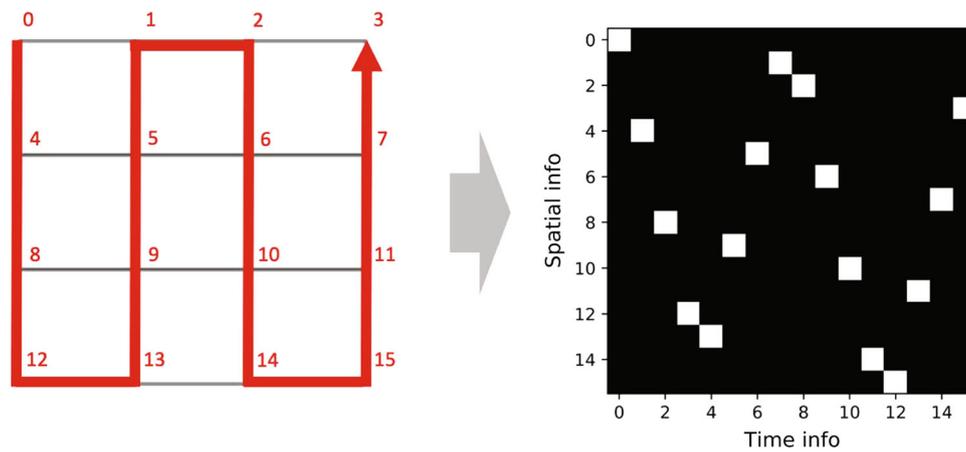


Fig. 12 Preprocessing of path data—the x-axis of the grayscale image corresponds to the time information, and the y-axis of the grayscale image corresponds to the spatial information. Since the laser grid on

the left subfigure has 16 grid nodes, the grayscale image on the right subfigure should have a size of 16 by 16

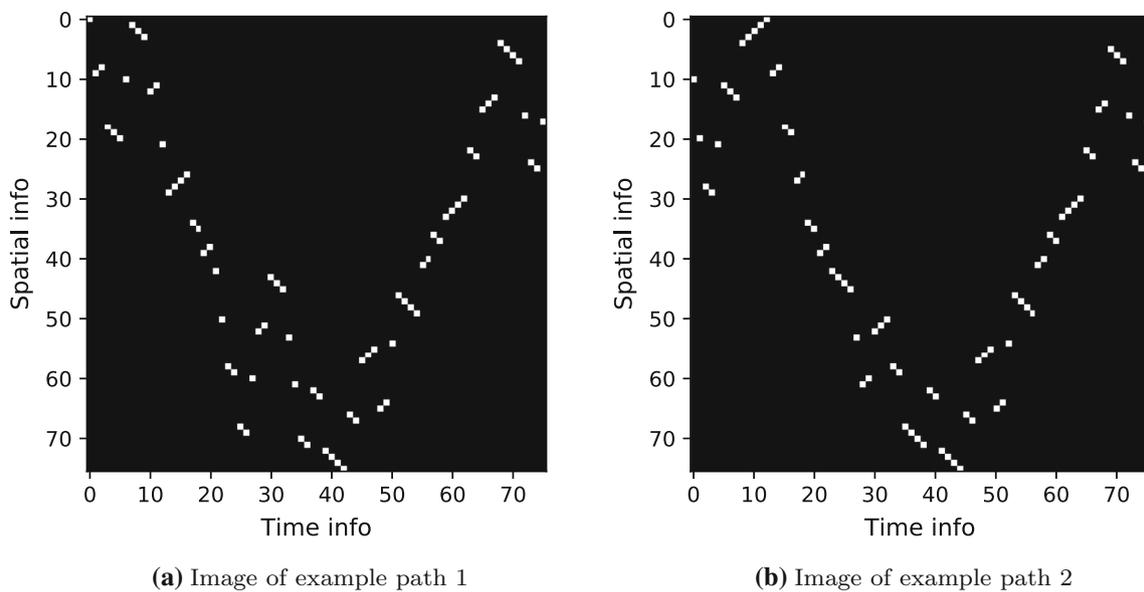


Fig. 13 Preprocessing of the paths into path map images

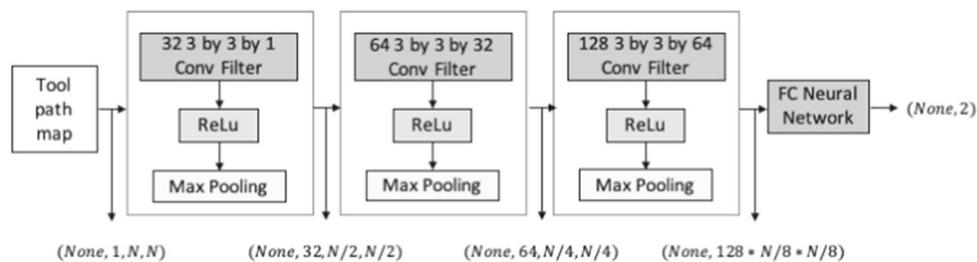


Fig. 14 CNN architecture

considering the symmetry of the geometry, are also shown in Fig. 8. There were 66,464 possible laser paths in total for this geometry and corresponding laser grid configuration, which were found by dynamic programming. The example

laser paths are shown in Fig. 9. Also, the temperature plots of the powder bed for the corresponding laser paths at the final processing time are shown in Fig. 10.

In the same way, we can perform the numerical simulations for all the other laser paths to obtain the corresponding average thermal gradients described in Eq. 3.14. The data distribution of the temperature gradients for 66464 laser paths is shown in Fig. 11.

5 Preprocessing of the laser paths

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to image classification and recognition [29–32]. The CNN processes data that has a known grid-like topology [33]. Our next step is to convert the laser paths we found into gray-scale path map images to train the deep learning model (CNN). The way to convert the laser paths (time data) into the image (space data) is shown in Fig. 12. The path maps (as shown in Fig. 12) also help to visually check that each node was visited only once by looking at each row. Node numbering goes from the left to the right, and from the top to the bottom, starting from zero. On the converted path map image, we identify the white points based on the laser path. In other words, the white squares correspond to the laser location at each time step. For example, if the second point of the laser path is node number 4, then the time-stamp identifier for this point is ‘1’ (since node numbering starts from zero), and the spatial identifier of this point is ‘4’.

Since the number of the laser grid nodes of the given geometry is 76, every single laser path is preprocessed into 76 by 76 gray-scale images. The preprocessing of the previous example paths for the given geometry is shown in Fig. 13.

6 A deep learning model to predict the optimal tool path

Deep learning is a branch of artificial intelligence based on a biologically-inspired learning process based on how neurons communicate and learn in living things, allowing computers to learn from the past data so that it could detect patterns and make predictions from noisy and complex data sets [33–37]. The deep learning approach deals with the design of algorithms to learn from machine-readable data. Also, there has been some research on generating predictive models to solve a variety of engineering problems such as material design, computer vision, pattern recognition, and spam filtering [38–45], including those in computational mechanics [46]. In our approach, we applied a deep learning algorithm to efficiently and accurately predict the optimal laser paths for SLS. We implemented our deep learning model with PyTorch (1.1.0 version), which is an efficient deep learning framework for Python and competent in both usability and speed [47].

In the previous section, we converted laser paths into grayscale images. Those images were processed in our deep

Table 2 Hyper-parameters (CNN model)

Hyper-parameter	Value
Convolution filter size	(3,3)
Max-pooling	(2,2)
Padding	1
Stride	1
Learning rate	0.001
Training epochs	25
Batch size	200
The number of hidden layers	1
The number of nodes in a hidden layer	800
Activation function	ReLU
Weight initialization	Xavier uniform
Loss function	Cross entropy loss
Optimization method	Adam optimizer

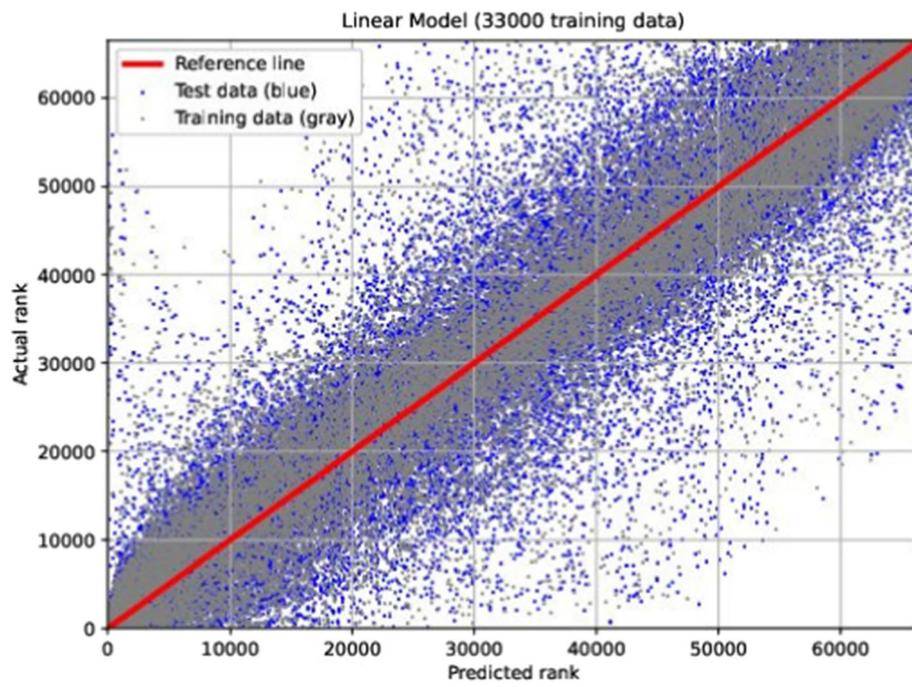
Table 3 Hyper-parameters (Linear model)

Hyper-parameter	Value
Learning rate	0.001
Training epochs	25
Batch size	200
Weight initialization	Xavier uniform
Loss function	Cross entropy loss
Optimization method	Adam optimizer

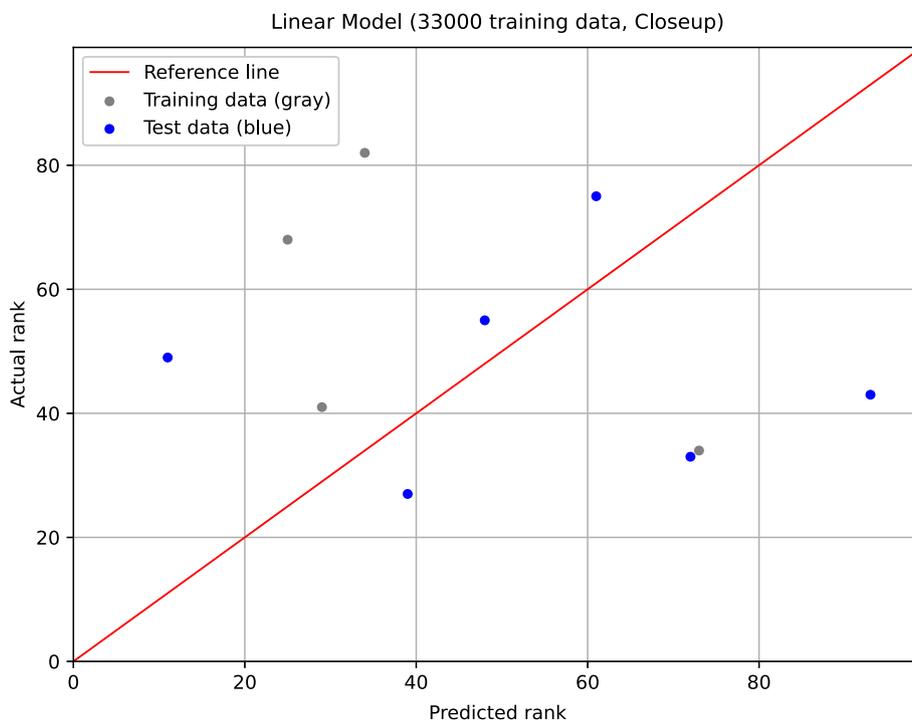
learning model. Also, every path map image was ranked before training with the deep learning model, based on the average thermal gradient described in Eq. 3.14. The lower the average thermal gradient, the better the path. After that, we gave each path two kinds of labels: half of the group was labeled as ‘good’ (a label of ‘1’) and the other half was labeled as ‘bad’ (a label of ‘0’). In other words, the samples having the temperature gradient higher than the median temperature gradient were labeled as ‘bad’, and the samples having the temperature gradient lower than the median were labeled as ‘good’.

The architecture of the CNN that we used for training and prediction is shown in Fig. 14. In order to reduce architectural complexity, we used a simple CNN structure having three convolutional layers, which was followed by a fully connected neural network. In Fig. 14, *None* represents the number of data samples, and *FC neural network* represents the fully connected neural network. For initialization of the parameters, Xavier initialization was used for the weight initialization, in order to obtain substantially faster convergence [48]. The configuration of the hyper-parameters for the CNN model is shown in Table 2.⁴

⁴ It is hard to say that the given configuration of the hyper-parameters is the optimal choice for the deep learning model.

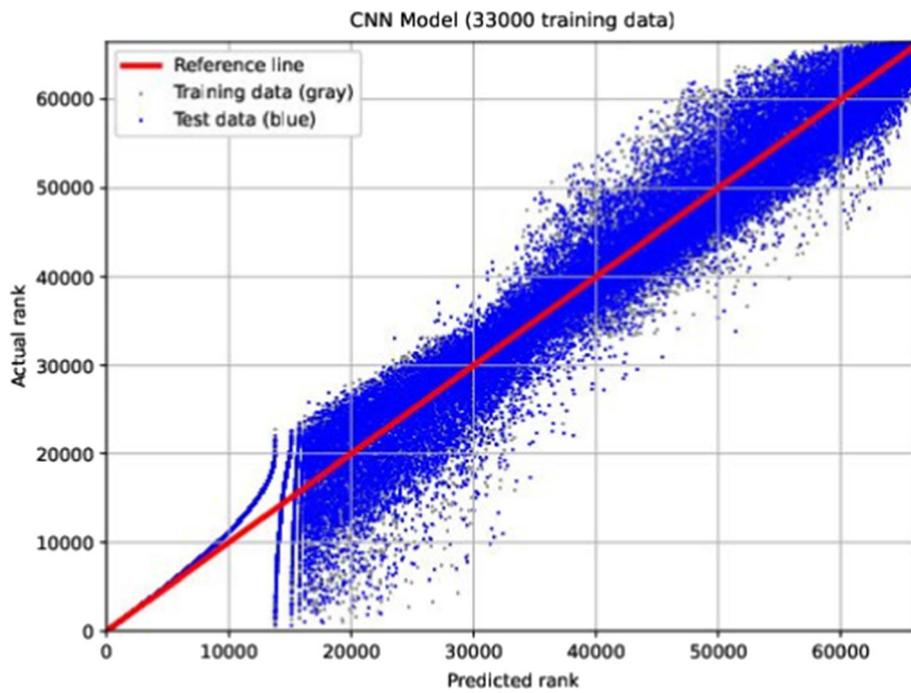


(a) Linear model (Full ranking)

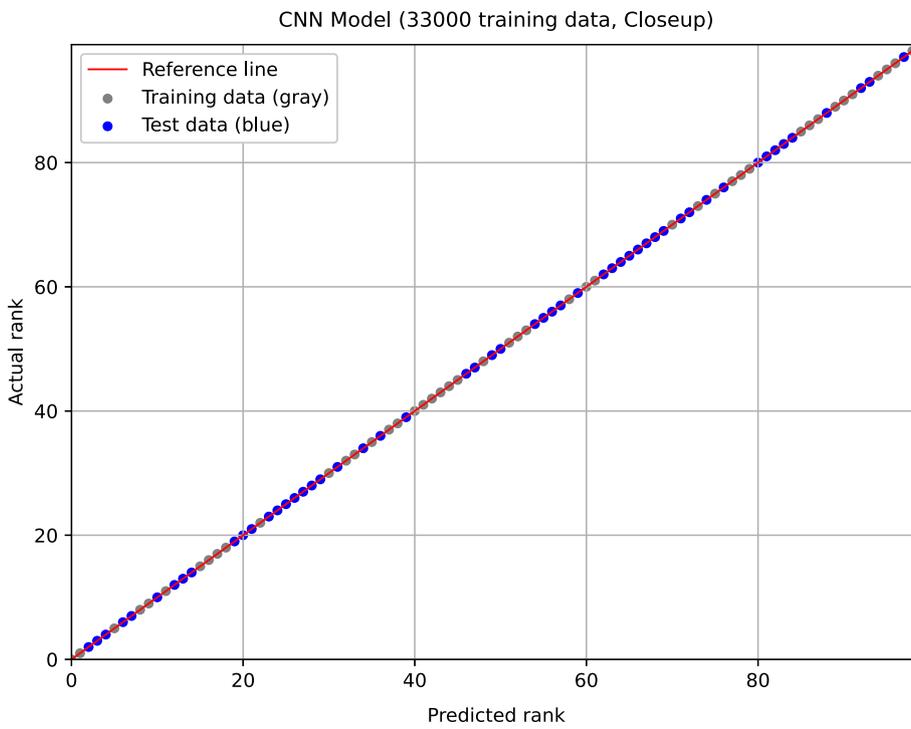


(b) Linear model (Closeup)

Fig. 15 Ranking prediction (linear model, 33000 training data)

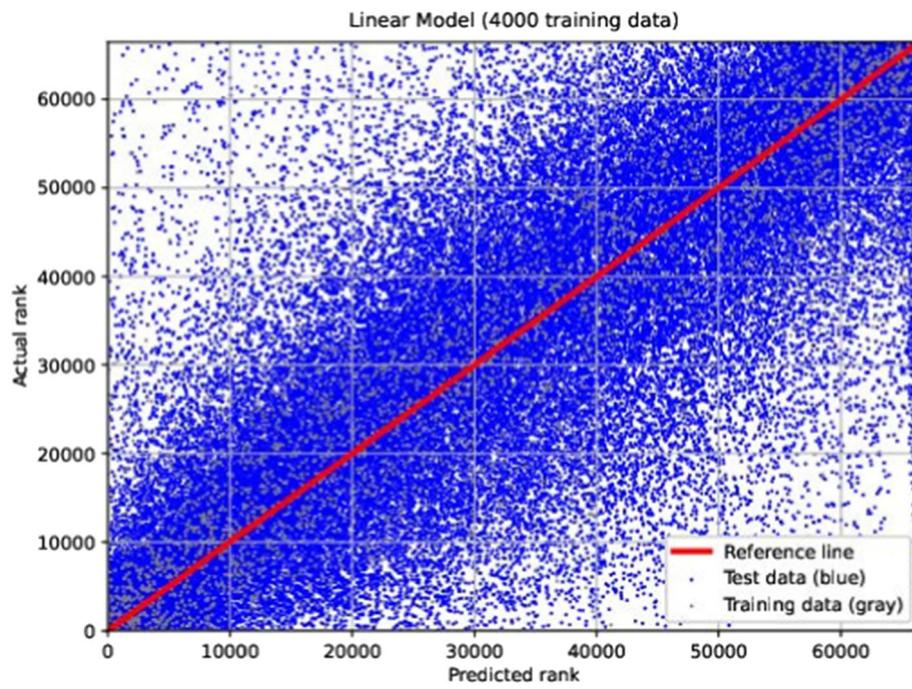


(a) CNN model (Full ranking)

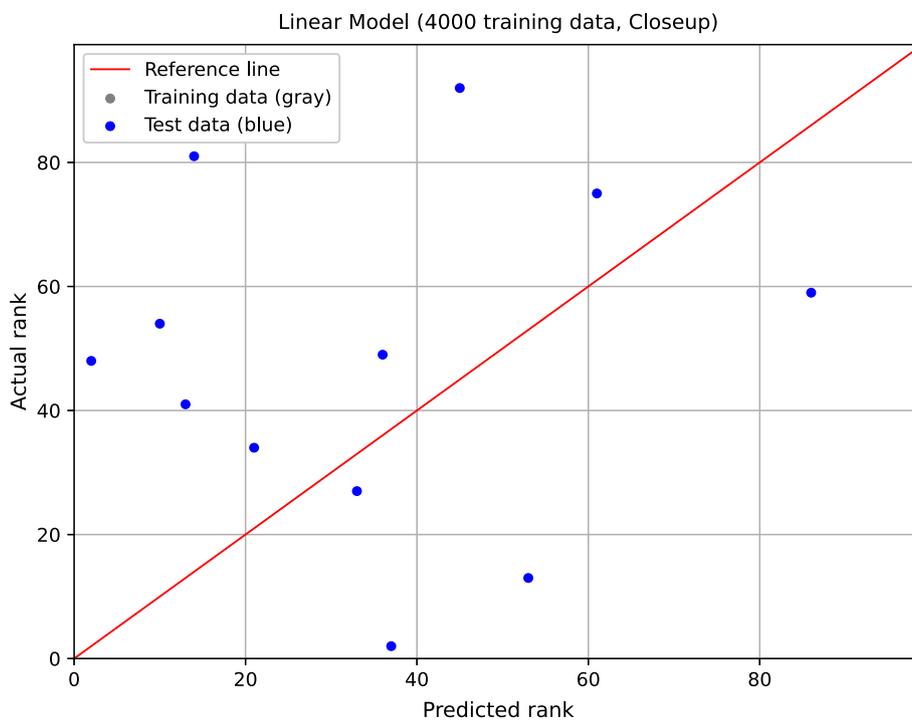


(b) CNN model (Closeup)

Fig. 16 Ranking prediction (CNN model, 33000 training data)

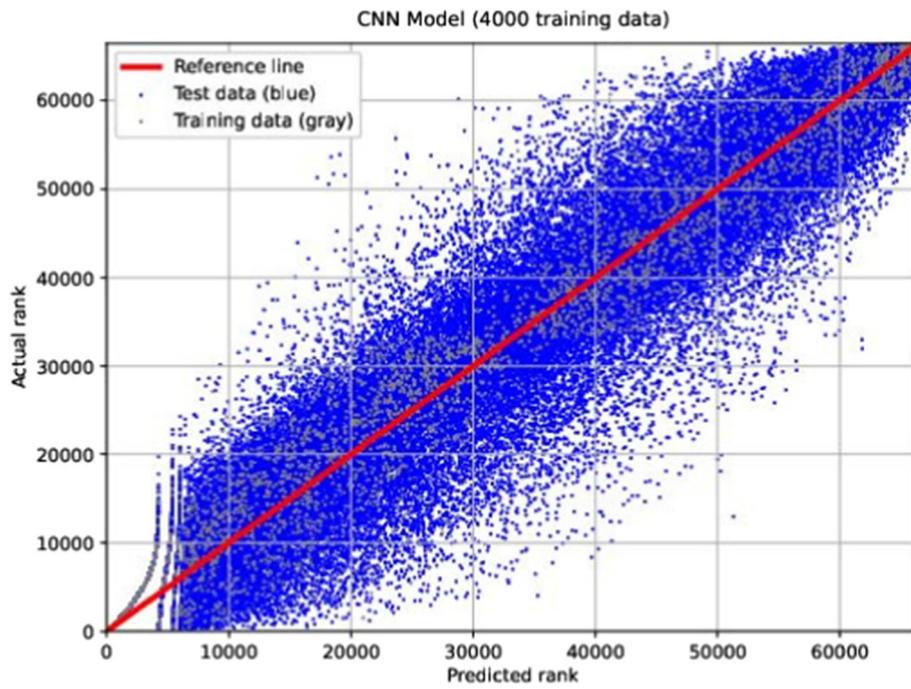


(a) Linear model (Full ranking)

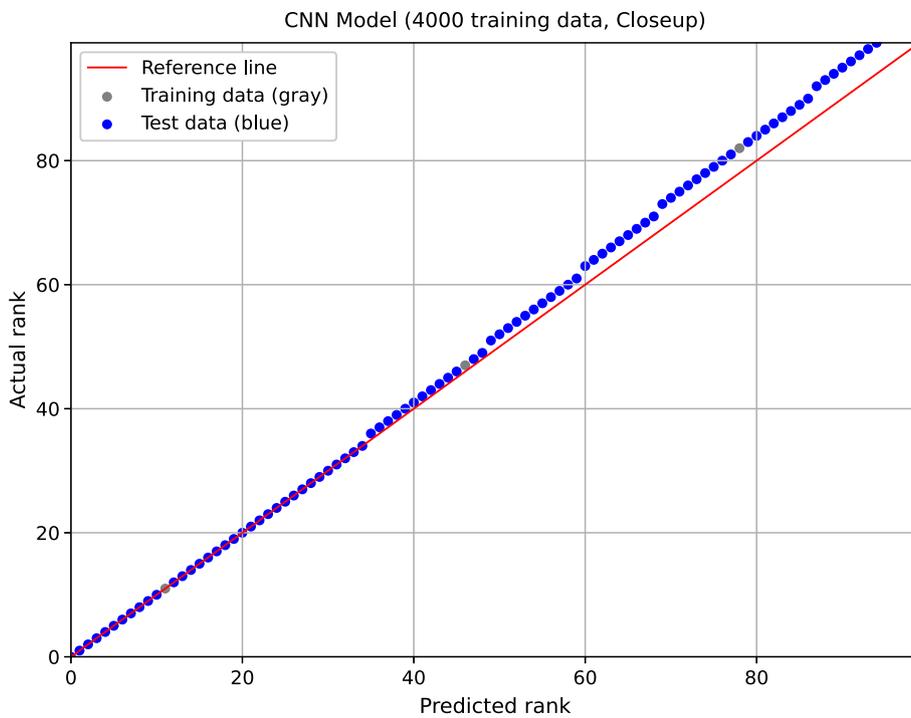


(b) Linear model (Closeup)

Fig. 17 Ranking prediction (Linear model, 4000 training data)



(a) CNN model (Full ranking)



(b) CNN model (Closeup)

Fig. 18 Ranking prediction (CNN model, 4000 training data)

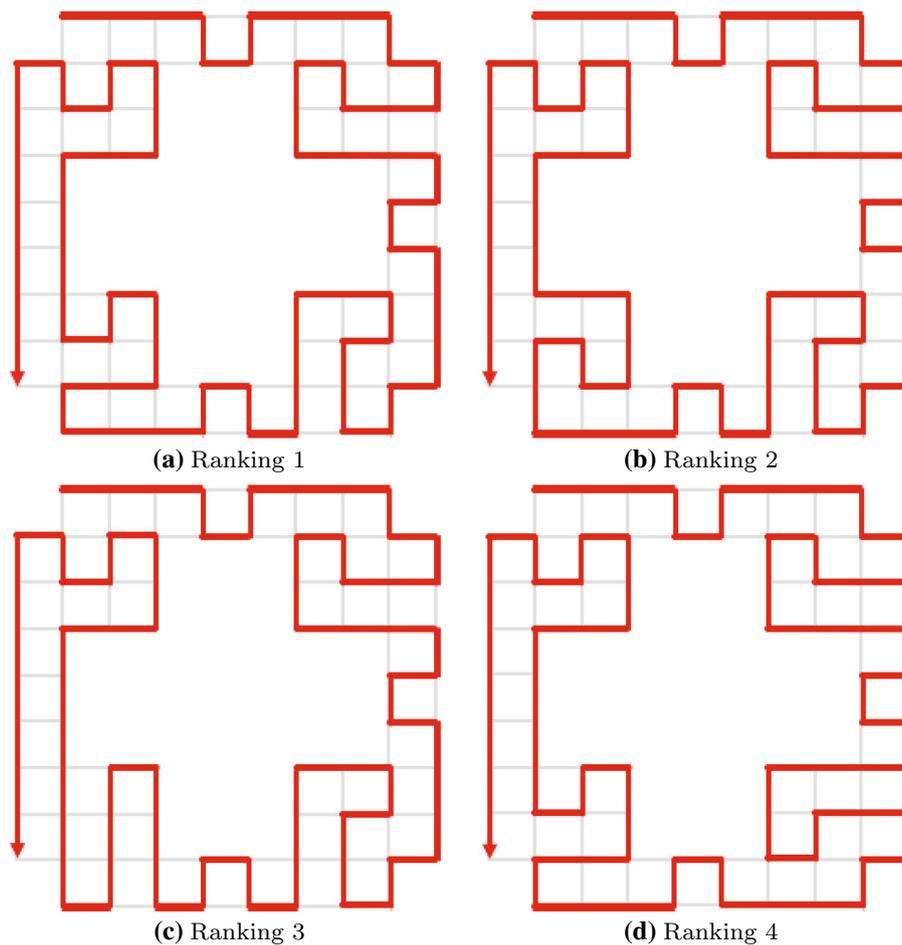


Fig. 19 Top four paths of the linear model

After passing through the CNN, we used the softmax function to convert neural network output numbers to probabilities (of being either a good path or a bad path) for each image. We not only used the softmax function to calculate the cross entropy loss function while training, but also used it as a probability extractor to predict the probability of being a good path for each path map image in a test data set. The softmax probability function is described in Eq. 6.1.

$$P_{\theta}(y^{(m)} = i) = \frac{e^{z_i^{(m)}}}{\sum_{k=0}^1 e^{z_k^{(m)}}}, \quad (6.1)$$

where i is the label of either 1 (good) or 0 (bad), $z_i^{(m)}$ is the output number of m^{th} image data for the label i from the CNN, and $P_{\theta}(y^{(m)} = i)$ is the predicted probability for m^{th}

image to have label i . Also, for multi-class classification, we used the cross entropy loss as in Eq. 6.2. We used this loss function as an error metric while training as well.

$$L = -\frac{1}{N} \sum_{m=1}^N \sum_{i=0}^1 P(y^{(m)} = i) \cdot \log P_{\theta}(y^{(m)} = i), \quad (6.2)$$

where N refers to the number of data samples (path map images), and $P(y^{(m)} = i)$ is the actual probability (either 0 or 1) for m^{th} image to have label i . In order to train the CNN with the cross entropy loss function, an Adam optimizer was used for the optimization, which is computationally efficient and has little memory requirements [49]. Also, note that the test data is brand new data for the trained model, which means they were completely isolated from the training data set and training process. The training set was **randomly** selected from all data sets. In addition to the CNN model, we also trained and tested the linear model to compare the predictive power of both models. The configuration of the hyper-parameters for the linear model is shown in Table 3.

Footnote 4 continued

One could try optimizing hyper-parameters by using optimization techniques such as Bayesian optimization, evolutionary algorithms, or gradient-based optimization.

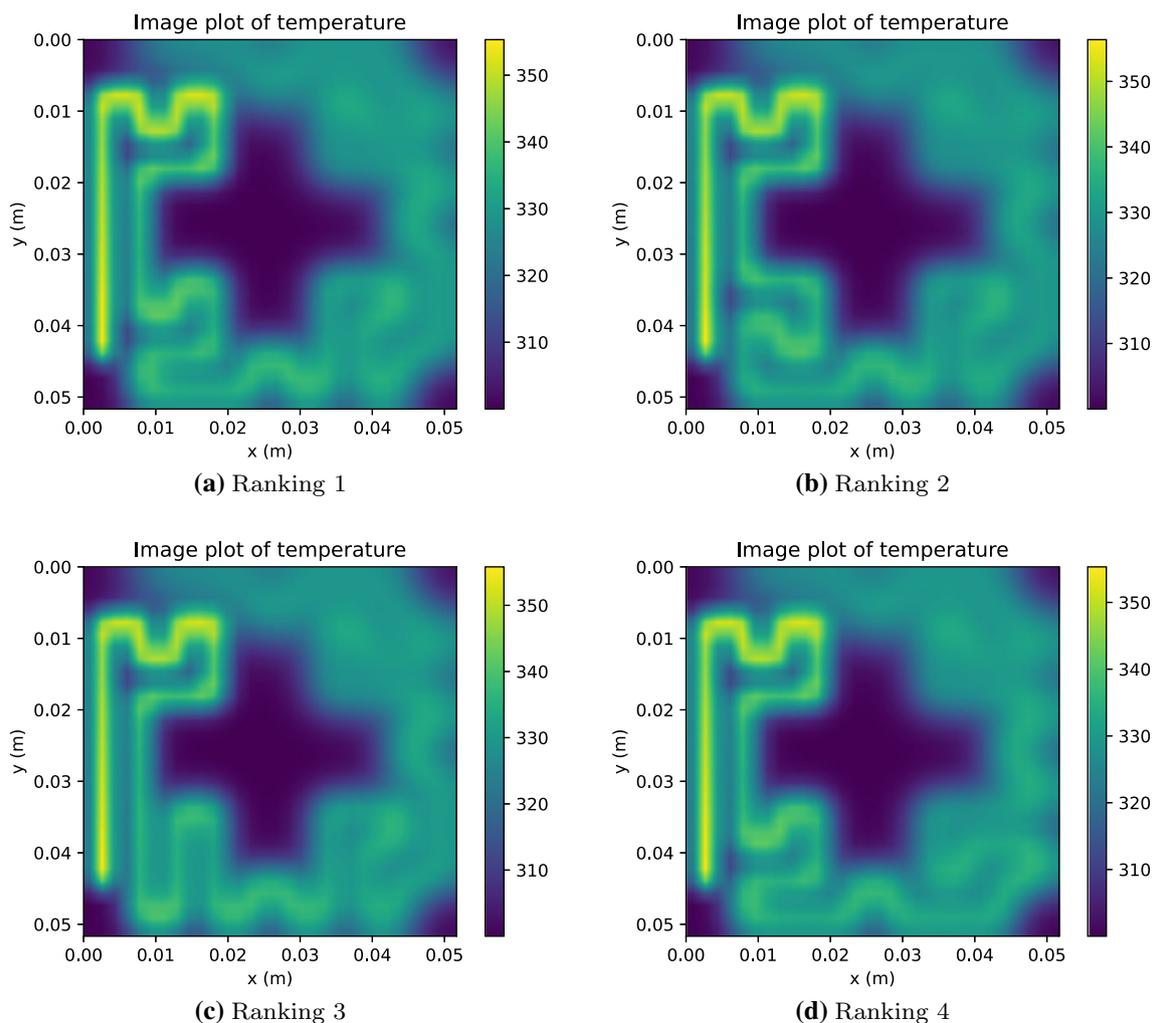


Fig. 20 Temperature plots at the top surfaces (Top 4, linear model)

7 Simulation results

With the deep learning model described above, we predicted what the optimal laser path should be, by ranking them based on the probability of being a good path, as obtained by the softmax probability extractor in Eq. 6.1. After that, we compared the predicted results from the CNN model with that of the linear model.

We first trained both the CNN model and the linear model with 33,000 point training data set (49.7% of 66464 total data points). In this case, the rest of the data (33464 paths) are what we actually make the predictions for. The full rankings of the predicted results for both models are shown in Figs. 15a and 16a. Also, closeups on the top-ranked path for Figs. 15a and 16a are shown in Figs. 15b and 16b, respectively. We observe that the linear model could not capture the actual optimal laser path well, as shown in Fig. 15. That is because the system is highly nonlinear. Even though the linear model is the least expensive machine learning model with the lowest

computational cost, we need to account for the accuracy. However, we can observe that the CNN model captures the highest-ranked laser path quite well, as shown in Fig. 16. We also attempted to train both the CNN model and the linear model with quite small data sets: 4000 training data points (only 6.02% of 66464 total data points). In this case, the rest of the data (62464 paths, 93.98%) are used to make the predictions. The full rankings of the predicted results for both models are shown in Figs. 17a and 18a. Also, closeups on the top-ranked path for Figs. 17a and 18a are shown in Figs. 17b and 18b, respectively.

We clearly observe that the linear model could not capture the actual optimal laser path pattern, as shown in Fig. 17. This implies that the linear model becomes less useful when the data set is even smaller. However, the CNN model still captures the high-ranked laser paths reasonably well, in spite of a much smaller training set, as shown in Fig. 18. The CNN model is quite successful in capturing the top 35 paths, even though 34 paths of them were from the test data set. The top

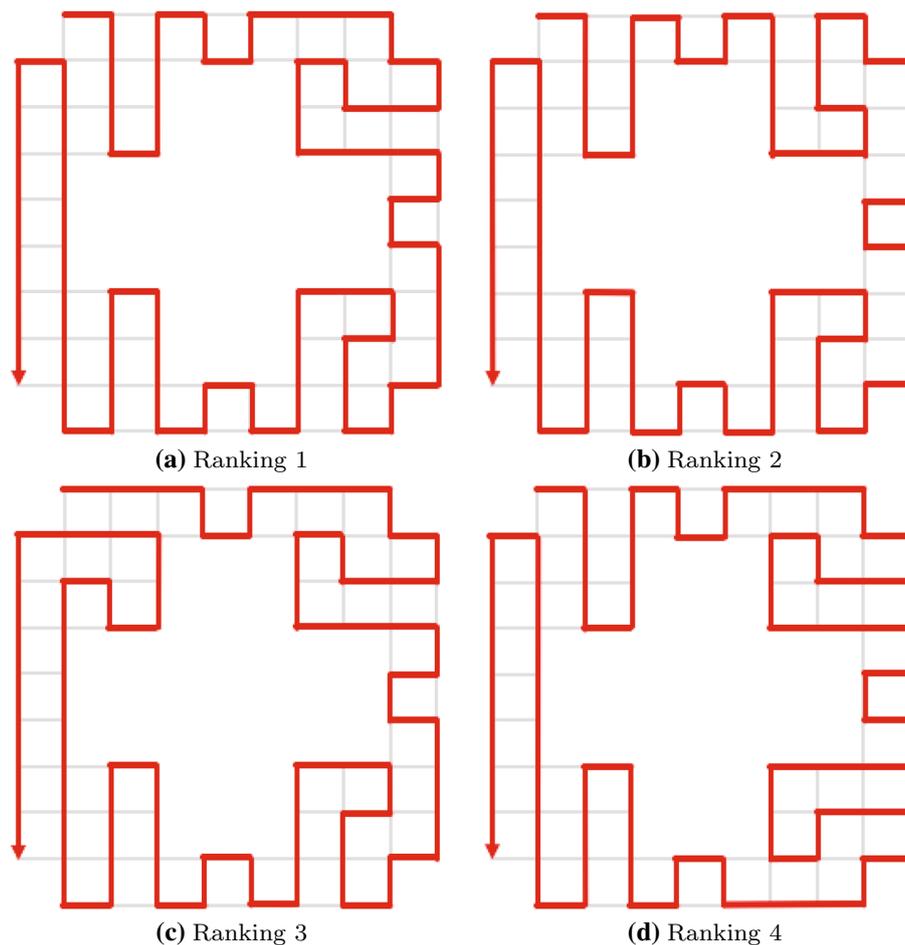


Fig. 21 Top 4 paths of the CNN model

4 paths which were chosen by the linear model are shown in Fig. 19, and the corresponding plots of the temperature at the top surface of the powder bed are shown in Fig. 20. Also, the top 4 paths which were chosen by the CNN model are shown in Fig. 21, and the corresponding plots of the temperature at the top surface of the powder bed are shown in Fig. 22.

The computing times for all the above models are shown in Table 4. The computer employed for the entire simulation was a *MacBook Pro (Retina, 15-inch, Mid 2014)*, and the corresponding CPU was a *2.8 GHz Intel Core i7*. The exhaustive simulation time for all the 66,464 laser paths was 38,109.53 (s). The total time represents the entire simulation time, including every single path simulation for the number of training data points, the training time, and the prediction time. The acceleration represents how many times faster it is than direct exhaustive simulations for 66,464 possible paths.

As we see in the table, the CNN model with 4000 training data (**only 6.02%** of 66464 total data) was more than 10 times faster than a brute force simulation for all the possible laser paths needed to calculate the thermal gradients, with the desired accuracy. This implies that the deep learning model

can learn the pattern of the preferable paths (which includes the process of solving differential equations) and predict the optimal laser paths accurately and efficiently, without knowing geometry and without having any mathematical or physical knowledge to solve differential equations and optimization problems.

8 Summary

As this work illustrated, optimal laser paths can be accurately predicted using a deep learning technique, even with a very small amount of training data and binary information (good/bad paths). Furthermore, a Deep Learning simulation using the CNN was significantly faster than a brute force simulation. These results illustrate the potential of Deep Learning for tool path optimization, in particular the ability to learn the patterns of tool paths and rebuild detailed path performances. In this work, we only considered the average thermal gradient at the final processing time as the cost of the laser path. In other words, we ranked the laser paths solely

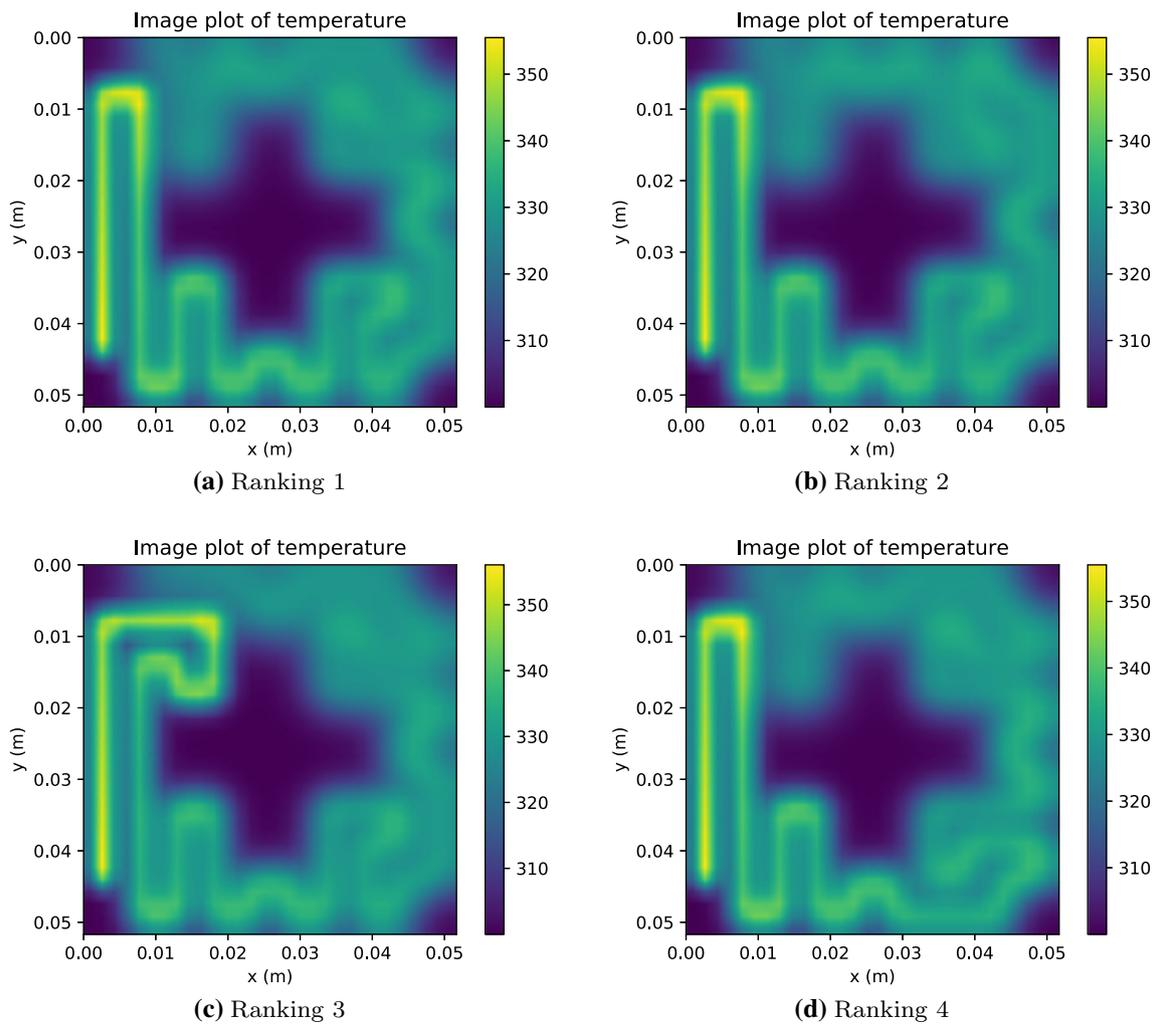


Fig. 22 Temperature plots at the top surfaces (Top 4, CNN model)

Table 4 Computation time comparison

Model	Training data	Training (s)	Prediction (s)	Total time (s)	Acceleration
CNN	33,000	3872.44	747.45	23,541.63	1.619
CNN	4000	473.13	752.91	3519.58	10.828
Linear	33,000	12.58	0.52	18,934.84	2.013
Linear	4000	1.98	0.47	2295.99	16.598

based on the average thermal gradient values. However, other operation parameters such as tool maneuverability, operating conditions could be included to calculate the cost or optimality of the laser paths.

Clearly, the idea of a physics-based deep learning approach for tool path optimization can be widely applied to many other additive techniques including fused deposition modeling, selective laser melting, and direct energy deposition. Also, deep learning techniques can be applied to many other fields of numerical simulations, well beyond simple tool path optimization for additive manufacturing, and deliver

reduced simulation costs while ensuring desirable accuracy [46,50,51]. However, there will always be trade-offs between computational costs and accuracy. More computational effort may be required if we use more sophisticated deep learning models, yielding good predictive capabilities, as opposed to simple deep learning models, with limited predictive power. Determining the optimal balance between these competing interests is an ongoing issue across this field.

Acknowledgements This research was supported by Samsung Scholarship. Also, the first author, Dong Hoon Kim, thanks Euihyun Choi for providing valuable thoughts on tool path generation.

Funding This research was supported by Samsung Scholarship.

Availability of data and material All the relevant data and the material is available at https://github.com/kdhoon2/SLS_DHKIM.

Declarations

Conflicts of interest/Competing interests Not applicable

Code availability The simulation code is available at https://github.com/kdhoon2/SLS_DHKIM.

References

- Aldahash SA (2018) Optimum manufacturing parameters in selective laser sintering of PA12 with white cement additives. *Int J Adv Manuf Technol* 96(1–4):257–270
- Steen WM, Mazumder J (2010) *Laser material processing*. Springer, Berlin
- Zeng K, Pal D, Stucker B (2012) A review of thermal analysis methods in laser sintering and selective laser melting. In: *Proceedings of solid freeform fabrication symposium Austin, TX, vol 60*, pp 796–814
- Dong L et al (2009) Three-dimensional transient finite element analysis of the selective laser sintering process. *J Mater Process Technol* 209(2):700–706
- Kolossov S et al (2004) 3D FE simulation for temperature evolution in the selective laser sintering process. *Int J Mach Tools Manuf* 44(2–3):117–123
- Matsumoto M et al (2002) Finite element analysis of single layer forming on metallic powder bed in rapid prototyping by selective laser processing. *Int J Mach Tools Manuf* 42(1):61–67
- Simchi A (2006) Direct laser sintering of metal powders: mechanism, kinetics and microstructural features. *Mater Sci Eng A* 428(1–2):148–158
- Simchi A, Pohl H (2003) Effects of laser sintering processing parameters on the microstructure and densification of iron powder. *Mater Sci Eng A* 359(1–2):119–128
- Gusarov AV, Kruth J-P (2005) Modelling of radiation transfer in metallic powders at laser treatment. *Int J Heat Mass Transf* 48(16):3423–3434
- Gusarov AV et al (2009) Model of radiation and heat transfer in laser-powder interaction zone at selective laser melting. *J Heat Transf* 131(7):072101
- Ganeriwala R, Zohdi TI (2014) Multiphysics modeling and simulation of selective laser sintering manufacturing processes. *Procedia CIRP* 14:299–304
- Zohdi TI (2013) Rapid simulation of laser processing of discrete particulate materials. *Arch Comput Methods Eng* 20(4):309–325
- Zohdi TI (2014) Additive particle deposition and selective laser processing—a computational manufacturing framework. *Comput Mech* 54(1):171–191
- Zohdi TI (2014) A direct particle-based computational framework for electrically enhanced thermo-mechanical sintering of powdered materials. *Math Mech Solids* 19(1):93–113
- Zohdi TI (2017) Modeling and simulation of laser processing of particulate-functionalized materials. *Arch Comput Methods Eng* 24(1):89–113
- Ganeriwala R, Zohdi TI (2016) A coupled discrete element-finite difference model of selective laser sintering. *Granular Matter* 18(2):21
- Zohdi TI (2018) Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning. *Comput Methods Appl Mech Eng* 331:343–362
- Weinan E, Han J, Jentzen A (2017) Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun Math Stat* 5(4):349–380
- Cai Z et al (2020) Deep least-squares methods: An unsupervised learning-based numerical method for solving elliptic PDEs. *J Comput Phys* 420:109707
- Weinan E, Yu B (2018) The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Commun Math Stat* 6(1):1–12
- Mohan AT, Gaitonde DV (2018) A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. *arXiv preprint arXiv:1804.09269*
- Wang Z et al (2018) Model identification of reduced order UID dynamics systems using deep learning. *Int J Numer Meth Fluids* 86(4):255–268
- Bertsekas DP et al (1995) *Dynamic programming and optimal control*. Athena Scientific, Belmont
- Bellman R (1966) *Dynamic programming*. *Science* 153(3731):34–37
- Howard RA (1960) *Dynamic programming and Markov processes*
- Bellman RE, Dreyfus SE (2015) *Applied dynamic programming*. Princeton University Press, Princeton
- Hecht J (2018) *Understanding lasers: an entry-level guide*. Wiley, Hoboken
- Rombouts M et al (2005) Light extinction in metallic powder beds: correlation with powder structure. *J Appl Phys* 98(1):013533
- Valueva MV et al (2020) Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. In: *Mathematics and computers in simulation*
- LeCun Y et al (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551
- Y LeCun, Y Bengio et al (1995) Convolutional networks for images, speech, and time series. In: *The handbook of brain theory and neural networks*, vol 3361, no 10
- LeCun Y et al (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
- Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*. MIT Press, Cambridge
- Mitchell RS, Michalski JG, Carbonell TM (2013) *An artificial intelligence approach*. Springer, Berlin
- Murphy KP (2012) *Machine learning: a probabilistic perspective*. MIT Press, Cambridge
- Langley P et al (1994) Selection of relevant features in machine learning. In: *Proceedings of the AAAI fall symposium on relevance*, vol 184, pp 245–271
- Kononenko I, Kukar M (2007) *Machine learning and data mining*. Horwood Publishing, Cambridge
- Gu GX, Chen CT, Buehler MJ (2018) De novo composite design based on machine learning algorithm. *Extreme Mech Lett* 18:19–28
- Chen C-T, Gu GX (2019) Machine learning for composite materials. *MRS Commun* 9(2):556–566
- Y Kim et al. (2020) Designing adhesive pillar shape with deep learning-based optimization. *ACS Appl Mater Interfaces*
- K He et al. (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
- Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. *Expert Syst Appl* 36(7):10206–10222

43. Santos I et al. (2009) Machine-learning-based mechanical properties prediction in foundry production. In: 2009 ICCAS-SICE. IEEE, 4536–4541
44. Mannodi-Kanakithodi A et al (2016) Machine learning strategy for accelerated design of polymer dielectrics. *Sci Rep* 6:20952
45. Pilania G et al (2020) Data-based methods for materials design and discovery: basic ideas and general methods. *Synth Lect Mater Opt* 1(1):1–188
46. Oishi A, Yagawa G (2017) Computational mechanics enhanced by deep learning. *Comput Methods Appl Mech Eng* 327:327–351
47. Paszke A et al (2019) Pytorch: an imperative style, high-performance deep learning library. In: *Advances in neural information processing systems*, pp 8026–8037
48. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp 249–256
49. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
50. Wilt JK, Yang C, Gu GX (2020) Accelerating auxetic metamaterial design with deep learning. *Adv Eng Mater* 22:1901266
51. Kim DH, Zohdi TI, Singh RP (2020) Modeling, simulation and machine learning for rapid process control of multiphase owing foods. *Comput Methods Appl Mech Eng* 371:113286

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.