Hybrid sparse convolutional neural networks for predicting manufacturability of visual defects of laser powder bed fusion processes

Ying Zhang, Yaoyao Zhao

Hybrid Sparse Convolutional Neural Networks for Predicting Manufacturability of Visual Defects of Laser Powder Bed Fusion Processes

Ying Zhang, Yaoyao Fiona Zhao*

Department of Mechanical Engineering,

McGill University, Montreal, Quebec H3A0G4

Email: ying.zhang8@mail.mcgill.ca, yaoyao.zhao@mcgill.ca

* Corresponding author

Abstract

This paper presents a novel solution to predict the visual defects, one of the major criteria for analyzing manufacturability for the Laser-based Powder Bed Fusion (LPBF) process. For the existing manufacturability investigations, the key challenge is how to model the complex interrelationships among the design, process, and final quality. The recent research proposed machine learning methods to model the manufacturability analysis. Voxel-based Convolutional Neural Network (CNN) has been investigated as one potential solution for design shape analysis. Those approaches are limited by the computational capability available and only lower resolution was performed. However, low resolution is not enough for analyzing the LPBF manufacturing process precisely. Some detailed features may be omitted through the voxelization process. To solve this issue, a more efficient CNN is proposed in this paper. Design data is stored in a sparse matrix so that only the occupied voxels are trained by CNN operations. It joins with the process data, which is trained by a Neural Network (NN) model to make the prediction of manufacturability. By performing the generalized convolutions, the computational costs decrease sharply compared to voxel-based CNN, which offers the advantage of performing with high resolutions. The approach is validated in terms of effectiveness and efficiency on the manufacturability prediction for the LPBF process.

1. Introduction

In the past decade years, Additive Manufacturing (AM) has rapidly expanded in various areas such as aerospace, automotive, tooling, medicals, etc. [1-3] It opens new opportunities in both design and manufacturing by removing the constraints imposed by conventional manufacturing (CM) such as machining or casting. However, not all designs can be fabricated by AM. There are still some limitations that need to be considered before fabrication to be carried out. These constraints include but are not limited to the narrow printable material selections, poor surface roughness, pores, and unfeasible features (e.g., overhangs) [4-11]. Failure in identifying the design features or printing process parameter setups often leads to printing failure, which eventually leads to increased manufacturing costs and a slower product development process. To solve this issue, manufacturability analysis research has been conducted to develop methods or tools to ensure the design could be successfully printed at the early design stage. In this paper, the manufacturability for the Laser Powder Bed Fusion (LPBF) process - one of the AM techniques - is the focus. LPBF process has demonstrated its capability of producing metal or

polymer parts for a wide range of applications. The manufacturing failures this research aims to detect are visual defects such as geometric incompleteness, cracking, and warping.

Existing main strategies for analyzing manufacturability in the LPBF process have been identified as design guidelines, real-time process monitoring, manufacturing feature recognitions, and integrated modeling [12]. These approaches provide some basic functions of quality check, reparation of STL meshes, slicing, support structure generation, and limited functionality of overhang and thin feature detection. However, none of them offers an automated and well-modeled manufacturability analysis for a given design to be fabricated via the LPBF process. There has been an increasing demand for automated computational applications for LPBF, on which the tool helps to visualize the troublesome areas and give feedbacks for modification. Recently, a new approach of applying the Machine Learning (ML) method to analyze manufacturability has been widely investigated [13]. Comparing to conventional methods, ML has several advantages in solving manufacturability issues. When evaluating manufacturability, the key challenge is the variation of each design and LPBF machine plus the compounded complex interrelationship among the LPBF process, design, and qualities of the final products. The ML method has the advantages of solving these high complexity issues via the data-driven approach. The ML model has demonstrated its capability to solve complex and uncertain problems in many applications in the past. With a well-trained ML model, it is also expected to provide a better result than humans. Also, once the model has been trained, the prediction only takes less than a minute, which fits the scope of the initial manufacturability check at the early design stage. Moreover, with more and more data gathered through time, the performance of the model can be improved without changing the algorithms. The model can be easily updated with incoming data to make ML as the most suitable potential solution to model and analyze manufacturability.

In the existing approach on applying ML in AM manufacturability analysis, one typical input features for ML models are values such as critical feature parameters of the design and process information [14]. Typical process information used in ML includes temperature, layer thickness, scanning speed, etc. The critical design feature parameters can be overall size, minimum feature size, topology types, etc. Specifically, for the lattice structure, the critical features can be lattice topology type, strut diameter, overall dimension, etc. However, using critical values to represent the design is highly limited. Not all the designs can be represented in the form of critical design values. A better approach to using layer-wise images and voxelization of the three-dimensional (3D) design model has been proposed to solve this issue [15-19]. However, this approach is limited by the computational capability which only allows low-resolution images and 3D models to be used in the existing manufacturability studies. Low voxelization resolution does not provide enough detail to accurately analyze the manufacturability of a LPBF process. Some detailed features are omitted through the voxelization process. To solve this issue, a hybrid ML model with sparse Convolutional Neural Network (CNN) is proposed in this paper. The design data is stored in a sparse data structure to be efficiently trained by CNN operations and joins with the process data, which is trained by a Neural Network (NN) model to make the prediction of manufacturability. The results demonstrate that the hybrid sparse CNN model offers better

results and lower computational costs than the voxel-based CNN approach and conventional manufacturability analysis which provides the benefits of performing high-resolution data.

To summarize, the contributions of this paper include, firstly, it provides an efficient ML model to predict whether the given design can successfully print without visual defects. Secondly, in 3D convolutional neural network research, the resolution used most frequently is 64 x 64 x 64. Thus, our point view is that if the resolution of input data is higher than 64 x 64 x 64, it is considered high resolution. In our approach, we can reach up to 256 x 256 x 256. The proposed sparse CNN approach takes much less computational power than the traditional voxel-based CNN approach with better performance. To the best knowledge of the author, this is the first paper to use the sparse matrix as the shape representative for the design model in an ML-assisted manufacturability assessment for the LPBF process.

In the rest of the paper, the background will be introduced, and the related works will be reviewed and discussed in Section 2. In Section 3, the framework of the ML-assisted manufacturability analysis is depicted. The details of the data generation, ML operations, network structures, and learning parameters are explained in Section 3 as well. Section 4 provides the results for two applications: a two-class manufacturability classification to determine whether the given design is printable or not and a semantic manufacturability segmentation to determine the potential failure area. Finally, Section 5 concludes the research work and proposes future perspectives.

2. Background & Related Works

2.1. Manufacturability of LPBF

There is no clear definition for the manufacturability of the LPBF process in the literature. The authors defined the manufacturability of the LPBF process in their earlier publication [12] as the design characteristics which indicate how difficult or easy the design is from a manufacturing perspective. Manufacturability analysis is to ensure the part can be well fabricated based on defined design, materials, and manufacturing processes. It introduces three manufacturability levels to define what is "well fabricated". The first level is to make sure the printed part is free of visual defects such as geometric incompleteness and warping. The second level is to determine whether the printed part meets the requirements of dimensional accuracy. The third level is to achieve the requirements of the density, porosity, and cracks in the parts. Those defects can be caused by the limitation of the LPBF process, such as long unsupported bridges, clearance, wall thickness, which are under the resolution of the printer, etc. Another main reason to cause those defects is the residual stress generated by the uneven temperature distribution. Numerous studies [20-23] have been conducted to investigate how the residual stress affects the final quality of the printed parts.

In this paper, we focus on analyzing manufacturability level one, which is the visual defects including geometric incompleteness, and warping. It is the foundation of analyzing manufacturability on the other two levels. The manufacturability analysis refers to the analysis to check whether the given design is free of visual defects. If the given design can be printed without the visual defects, it is considered manufacturable through the LPBF process.

2.2. Fundamentals of ML

ML systems automatically learn trends from data that allow them to make generalizations about instances they have not seen before. ML lies at the intersection of computer science and statistics and has been applied to a wide variety of problems where human intuition is not sufficient. Supervised, unsupervised, and reinforcement learning are three major categories in ML studies [14, 24]. This research focuses on supervised learning. Supervised learning takes labeled input data and creates a model to generalize on unlabeled data of the same format. Supervised learning can also be further broken down into regression and classification problems. Regression denotes that the output of the model will be continuous values such as print time or component cost. Statistical classification is the most mature and widespread application of ML. Classifiers typically input a vector of feature values and assign them to a discrete class. This is usually accomplished using decision boundaries to divide the input/feature space into regions, each representing a class, and observing where new data lies. In this paper, manufacturability analysis is treated as a classification problem to determine whether the entire part or each voxel of the part is fabricated or not.

There are many ML algorithms investigated in the literature [14, 24]. The most common algorithms include decision tree, support vector machine (SVM), Naïve Bayes, random forest, neural networks, etc. Neural networks, including classic feedforward NN and CNN, are the main algorithms applied in this paper. In the feedforward NN, every input for the current layer is connected to form every output by the weights. CNN is frequently used to deal with images or 3D objects which are multi-dimensional. In the convolutional layers, a set of "filters" are applied to a subset of the input variables at a time and swept over the entire input, so only nearby inputs are connected to each other, which results in far fewer weights than the fully connected layer. The objective of training a NN model is to find all the corresponding coefficients to minimize the loss function, which varies among models. The loss function is computed as the difference between the prediction and the ground truth. The selection of the loss function depends on the objective of the model. Popular loss functions include hinge loss, binary cross-entropy, mean absolute error, categorical cross-entropy, etc. The loss function can also be self-defined. To reach the minimum of the error function, there are a number of learning algorithms in the literature, such as quasi-Newton methods and stochastic gradient descent [24]. The process to minimize the loss cost in the model's prediction and compute all the corresponding coefficients is called backpropagation [25]. In this paper, it provides a hybrid machine learning model integrating a NN and generalized CNN to make the prediction of manufacturability for a given design with selected LPBF process parameters.

2.3. ML in AM Manufacturability Analysis

Recently, attempts have been made to train the ML model to predict the manufacturability of the given design, which is targeted to be fabricated in AM processes. The existing approach on applying ML in AM manufacturability analysis can be categorized into two major categories: real-time detection and analysis at the design stage.

For real-time detection, there are two main approaches. The first one is to use layer-wise images from the in-situ sensor as input features to predict the defects. The prediction of the defects can

be an image or simple yes or no value. The research conducted by Snow's group [17] offers a supervised ML for detecting flaws during the LPBF using images. Their approach collects the layer-wise images using a digital camera as input features to both NNs and CNNs. The ground truths are labeled according to the XCT scan data. Their results demonstrate a greater than 87.3% accuracy for detecting flaws. Moreover, their research concludes that CNN has a significantly better performance than NNs across all the tasks. Liu et al. [18] proposes similar research on applying ML in quality control with layer-wise and real-time images. Their system also provides the ability to automatically adjust machine parameters with the feedback of the ML prediction. Recently, Yazdi et al. [19] proposes an advanced research on applying ML in porosity defect with layer-wise and real-time images. Different than other approaches, their study presents the hybrid ML structure with two types of input data to monitor the process parameters. They extract the statistical wavelet decomposition coefficient from the images to go through a NN and merge with a CNN of the original images to make the prediction. Their approach demonstrates a satisfied performance with an F-score of 97.14%. The second approach is to use the process information such as temperature, layer thickness, scanning speed as the input to make the prediction of some characteristics such as roughness, tolerance, printability, etc. Such studies include the research proposed by Li's group [20] on using ML techniques to predict surface roughness with a set of input features in time and frequency domains, which were obtained from the raw sensor signals. Cerda-Avila et al. [21] recently proposed a similar approach by using ML techniques to predict the structural performance with the input features such as layer thickness, infill pattern, build orientation.

For the analysis at the design stage, there are two main approaches. The first one is to use the voxelization of the design model as the input to make the prediction of a single value such as printability, which is a yes or no question. Guo et al. [22] proposed a deep-learning-based framework for assessing the manufacturability of cellular structures in the LPBF process. He takes the voxelization of the design model as the input to predict manufacturability. The results demonstrate the capability of his model on the manufacturability analysis even with a small number of data. A similar idea conducted by Mycroft et al. [23] has been published as well. The second approach is to analyze the critical feature parameters of the design model (e.x. lattice type, strut diameters for the lattice structure) to make the prediction of the single values such as printability, ultimate strength, elastic modulus, etc. Sample research includes the study conducted by Hassanin et al. [26]. In his study, he chooses the strut length, strut diameter, and strut orientation angle as the input features to the ML model to predict the properties of the printed cellular structures.

To summarize, the real-time detection approach could not offer any help before the real fabrication. For the analysis at the design stage, the existing studies mainly use low-resolution images or voxelizations, and only consider the single aspect, either the design alone or just the process. There is no existing research to apply ML in AM manufacturability analysis at the design stage by using the sparse 3D structures with the consideration of the effects from both design and process aspects, which leads to the study proposed in this paper.

3. Methodology

Fig. 1 shows the general framework of ML-assisted manufacturability prediction system for a given design to be fabricated via the LPBF process, which was proposed by the authors in [27]. Instead of using a voxelized 3D geometric model directly as the design input to the ML models, the voxelized 3D geometric model is preprocessed into a sparse matrix, which is combined with a coordinate matrix and a feature matrix. Only occupied voxels are stored in the sparse matrix, and the ML operation is only applied in the sparse matrix, which decreases the computational cost sharply, and it also improves the performance of the models. The integration of the input features for design and material and processing information is first sent to a two-class manufacturability classifier to determine whether the given design is manufacturable or not with the given material and processing information. If it is printable, the given design will be ready for printing. If it is not printable, the input features will be sent to a semantic manufacturability segmentation step to determine which voxels are not printable in order to provide a printability map to designers. With the printability map, the potential failure area can be visualized to help designers for future design improvement. Supervised learning is chosen in this research as the starting point of the ML approach. Comparing to other two types of machine learning algorithms: unsupervised learning and reinforced learning, supervised learning has been used more widely demonstrating its feasibility and effectiveness in AM applications. In recent years, supervised learning has shown a high success rate in AM applications including candidate selections [28], surrogate models in design optimization and simulation models [29-31], and predictions on the quality of fabrications [20, 32, 33]. Supervised learning learns from labeled training data to predict unforeseen data. It takes the benefits of the existing experience, while in unsupervised learning, learning relies on the algorithms to find structured patterns [34]. Furthermore, reinforcement learning requires more data and more computations in order to achieve the prediction. This technique is preferred to achieve long-term results [35]. Since the proposed approach is part of a closed-loop manufacturability analysis system, the effectiveness of the machine learning model is important so that subsequent design modification recommendation research can be carried out. For our research, the prediction (output of the ML model) can be clearly classified and labeled. It is easy and beneficial to start developing the model with supervised learning.

In the rest of the section, the methodology proposed in this paper will be fully described. Section 3.1 will provide a clear description of the dataset as well as how the dataset is generated as the input features of the ML model. Section 3.2 will depict the methodology on how ground truth labeling is manually created. The generalized ML operations on the sparse matrix will be explained in Section 3.3. Section 3.4 will describe the network structures and hyperparameters to predict the manufacturability of the given design and processing information.

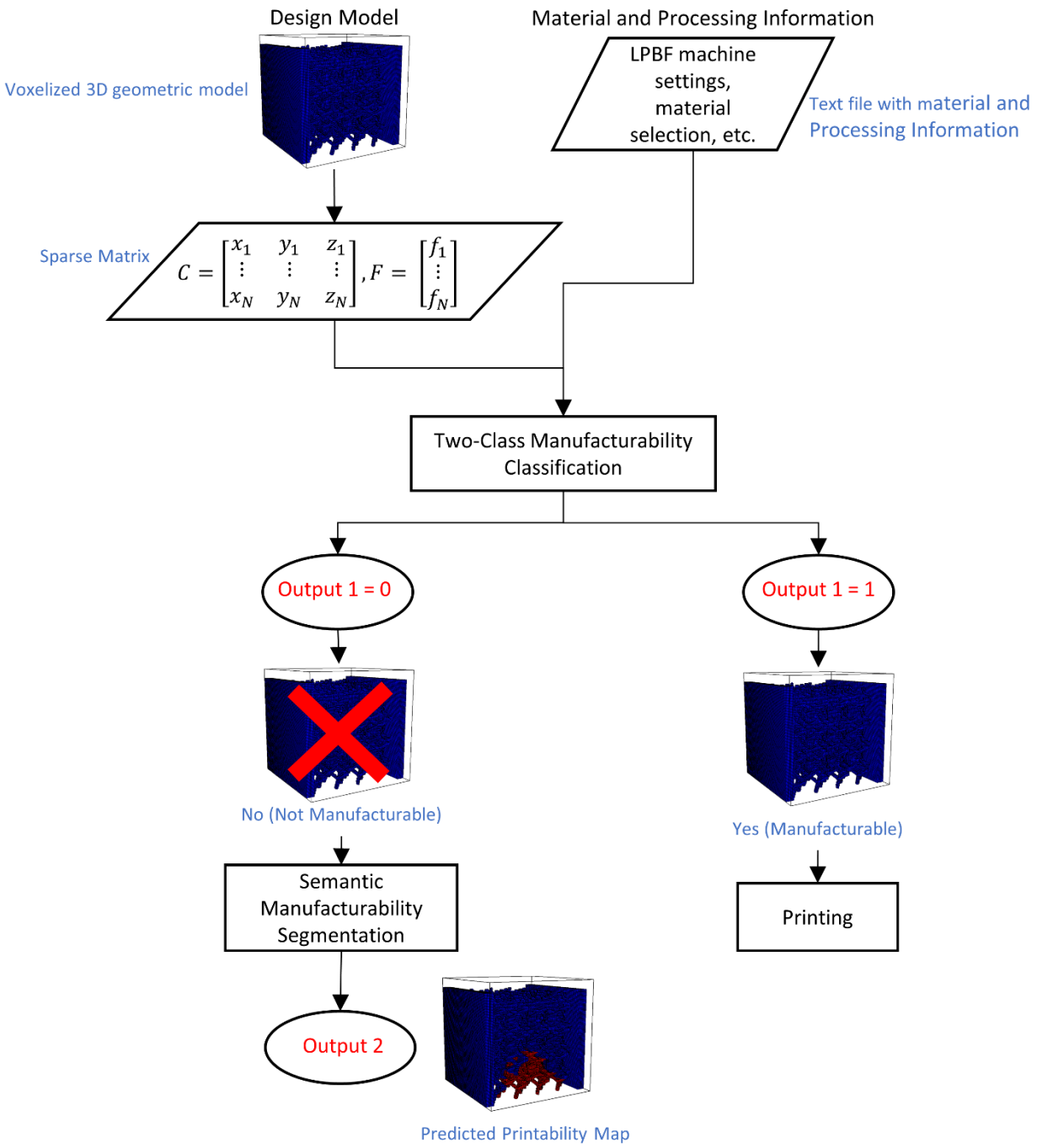

Fig. 1: Framework of the ML-assisted manufacturability analysis

3.1. Input Data Generation

In this paper, a collection of Computer-Aided Design (CAD) models are prepared for the ML prediction. The CAD models are originally saved as STL files. After that, those designs are fabricated by an LPBF machine. The results of the printed part are used as the ground truth of the ML model to optimize the manufacturability issues, which will be further explained in Section 3.2. The processing information such as material selections, machine selections, parameter

settings, etc. is applied as the input data as well. There is no specific requirement for the dataset. All the printed parts fabricated by any LPBF machine can be updated to the dataset. In this paper, 245 printed samples are collected in the dataset, and they are randomly split into a training dataset and a validation dataset to train and validate the ML model. The dataset includes samples such as different types of lattice structures, benchmark, tensile bars, cylinder channels, special design for automotive and tooling, etc.

To preprocess the design models, the first step is to voxelize the STL files. Binvox [36], which is a 3D mesh voxelizer, is used here to help with voxelization. After parsing, the result of the voxelization is turned into a 3D matrix. 1 is used here standing for occupied voxels, which are the main body of the designs, and 0 stands for the non-occupied voxels, which is the background. For most existing studies, the resolution of voxelization is low. Higher voxelization resolutions cost more computational power, which leads to a slow training process, and it may run out of computational capability. When observing the 3D design matrix, it is found that most of the elements in the design matrix are zero, which leads to a super sparse matrix. Especially for the high-resolution voxelized matrix, the sparsity is even higher. Fig. 2 shows an example comparison between the different voxel sizes for the same design model. For conventional CNN, the operation is applied to every element in the matrix. However, it is not necessary to apply the operations to all the zero elements, especially the zero-elements that are far from the main body of the design. The redundancy will waste the computational memory and slow down the training process. To solve these issues and improve the performance, the design matrix is preprocessed and stored into a sparse matrix, and ML operations are only applied to those occupied voxels, which will be explained in Section 3.3. The sparse matrix is stored as the combination of a coordinate matrix, C, and a feature matrix, F which can be represented as the following:

$$C = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix}, \; F = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \tag{1}$$

where x_i, y_i, and z_i are the coordinates, and N is the number of the non-zero elements. f_i is the associated feature values. For the input data in this study, feature values are uniform, which is 1 standing for occupied voxels.

Voxel Size	Sparsity
32x32x32	95.24%
64x64x64	96.60%
128x128x128	98.20%
256x256x256	98.16%
512x512x512	98.20%

Fig. 2: Comparison between different voxel sizes for the given design

To preprocess the material and processing information data, the same strategy is taken from the previous study [26]. One-hot encoding is applied to convert the categorical text variables into numerical variables to be provided to the ML model to do a better job in prediction. Such material and processing information includes material types and brand, material density in loose form, and machine brand and type. Machine settings such as laser power, layer thickness, printing speed, etc., can be further investigated in future work.

3.2. Data Labelling

To generate the ground truth for the ML model, the corresponding printed result to the given design is obtained. If the design is successfully printed with associated processing information, and the printed result is out of visual defects, the ground truth will be given a value of 1 as printable for the initial prediction. If the design is not printable, the truth will be given a value of 0 as non-printable. Moreover, if the design is not printable, the printability map will be generated. The original voxelized models are sent to an annotation tool to label the non-printable voxels. The example of the annotation tool is shown in Fig. 3. The tool is developed based on VoxCad [37]. The non-printed voxels are manually labeled as red layer by layer according to the printed result. Then the labeled results are transferred to the sparse matrix with coordinates and features as Equation 1. Note that the coordinate matrix should be the same as the one in the input. For the feature values in the ground truth, 1 stands for the printable voxels, and 0 stands for the non-printable voxels. Fig. 4 shows an example of the labeling process.

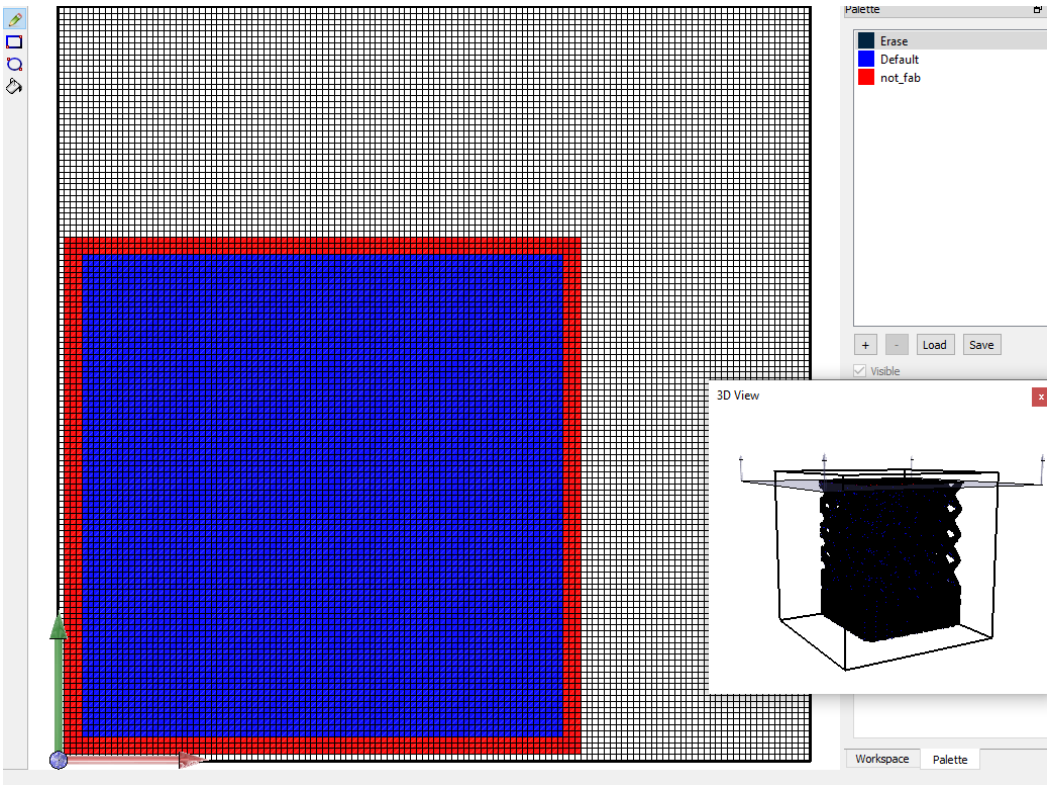

Fig. 3: Interface of the annotation tool

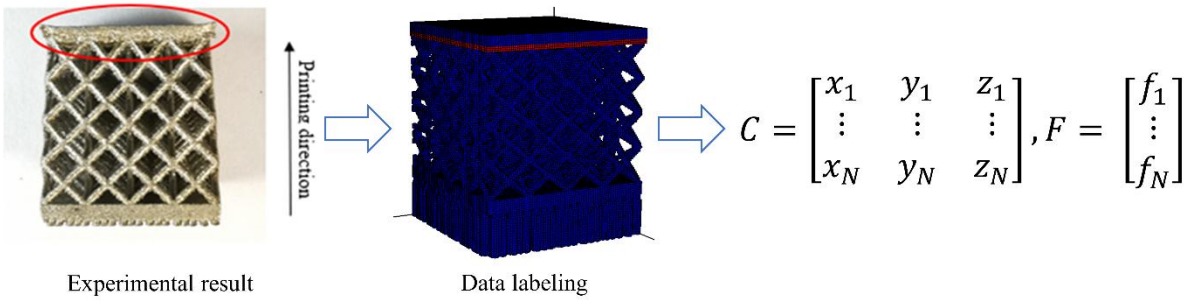

Fig. 4: Examples of data labeling

3.3. Network Operations on Sparse Data Structure

The convolution is the most important operation in CNN. For the conventional convolution operation, it is frequently used to deal with images or 3D objects which are multi-dimensional. In the conventional convolutional layers, a set of "filters" applied to a subset of the input variables at a time and swept over the entire input, so only nearby inputs are connected to each other, which results in far fewer weights than the fully connected layer. The convolution operation calculates the sum of the element-wise multiplication between the input matrix and filter matrix, so it performs a many-to-one relationship. Although it costs much less than the fully connected layer, it is still a very expensive operation in deep convolutional networks. In our generalized convolution, the operation only applies to the occupied voxels so that it requires much less

computational power than the standard convolution operations. It can be represented as the following equation:

$$O^{(n)}{}_{i,j,k} = \sum W^{(n)} I^{(n)}{}_{(i,j,k)} \ , (i,j,k) \in C \tag{2}$$

where $O^{(n)}{}_{i,j,k}$ is the n-th channel of the output value on voxel (i, j, k). $W^{(n)}$ represents the weights of the operation, which is the filter describing above. $I^{(n)}{}_{(i,j,k)}$ represents the n-th channel of the input features on voxel (i, j, k). C is the coordinate matrix generated in Section 3.1. To better illustrate the generalized operation, a graphic explanation is shown in Fig. 5. It provides a comparison between the standard convolution operation and the generalized convolution operation. On the left, the standard convolution operation swept over the entire input; however, for the convolution on the sparse matrix, it only swept over the occupied elements. When calculating the output value on the selected elements, it first gets the values from all the neighbors for the selected elements based on the stored coordinates. Then the "filters" are applied to the pattern, which is consisting of the selected point and surrounding neighbors to compute the output.

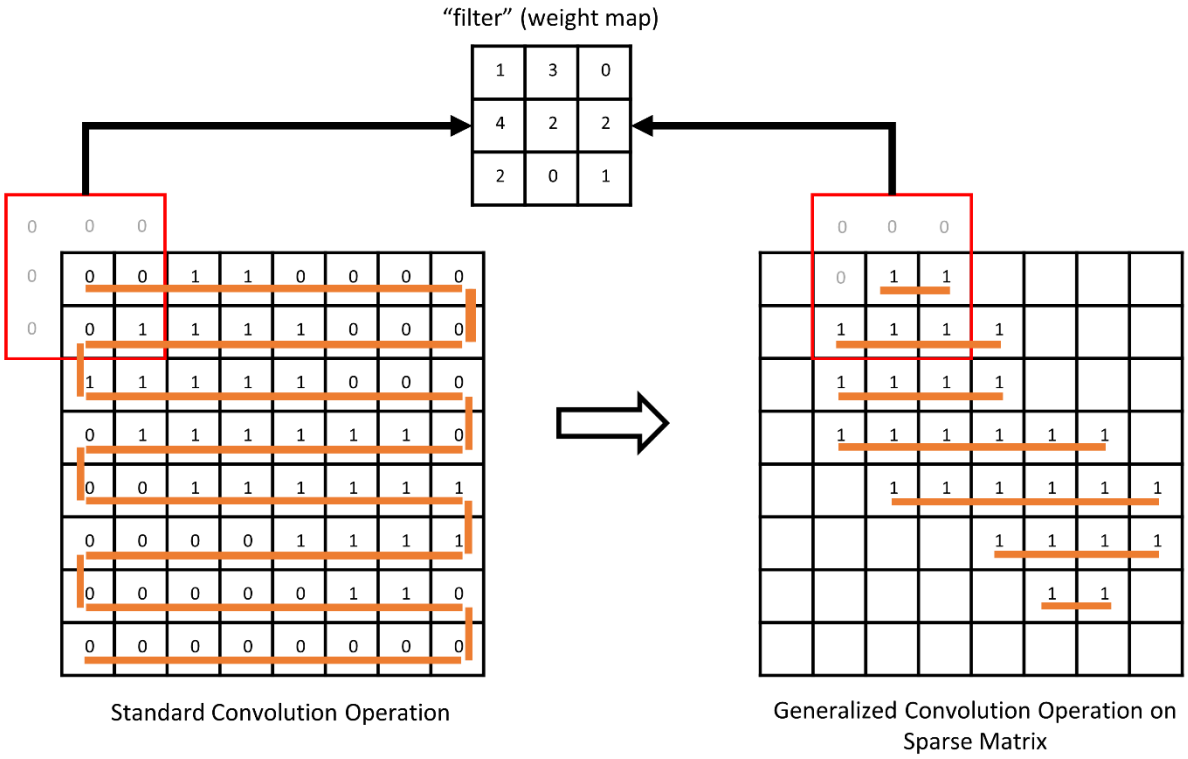

Fig. 5: Graphic explanation of convolution on sparse matrix

Pooling is the operation of downsampling the input variables by summarizing the presence of features in each patch [38, 39]. There are two common types of pooling operation. One is average pooling, which summarizes the average presence of the patch. Another one is the max pooling, which takes the most activated presence of the input patch. Pooling for sparse data has a similar idea to the conventional pooling operation. The difference is the pooling for the sparse

data only applies to the occupied voxels instead of the entire 3D matrix. The max pooling is used in this research, and it can be represented numerically as:

$$O_{i,j,k} = \max I^{(n)}{}_{(i,j,k)}, (i,j,k) \in C \quad (3)$$

where $O^{(n)}{}_{i,j,k}$ is the n-th channel of the output value of on voxel (i, j, k) and $I^{(n)}{}_{(i,j,k)}$ represents the n-th channel of the input features on voxel (i, j, k).

Transpose convolution on the sparse matrix takes the same idea as the standard transpose convolution. It is the opposite of the convolutional operation on the sparse matrix, which is also well known as deconvolution [40, 41]. It goes the backward direction of the convolution and performs a one-to-many relationship. For the convolutional operation, the output is always downsizing, but for the transpose convolution, a higher resolution output will be gotten in the end. Another new operation used in our network structure is broadcast, which is proposed by Choy [42]. It can be represented as the following equation:

$$F_u = x_2 \; for \; u \in \; C^{in} \quad (4)$$

It is the operation that copy value x_2 for all input coordinates, and F_u is the new feature values for the input coordinate.

3.4. Network Structure and Learning Parameters

Fig. 6 shows the network structure and major-related hyperparameters for the two-class manufacturability classification. As mentioned above, the design inputs are mainly treated by sparse convolution operation, and the material and processing inputs go through a feedforward NN. The two models are integrated and go through another two fully connected layers to make the predictions of the printability. Major hyperparameters include the number of filters or neural units, activation functions, kernel size, stride size, etc.

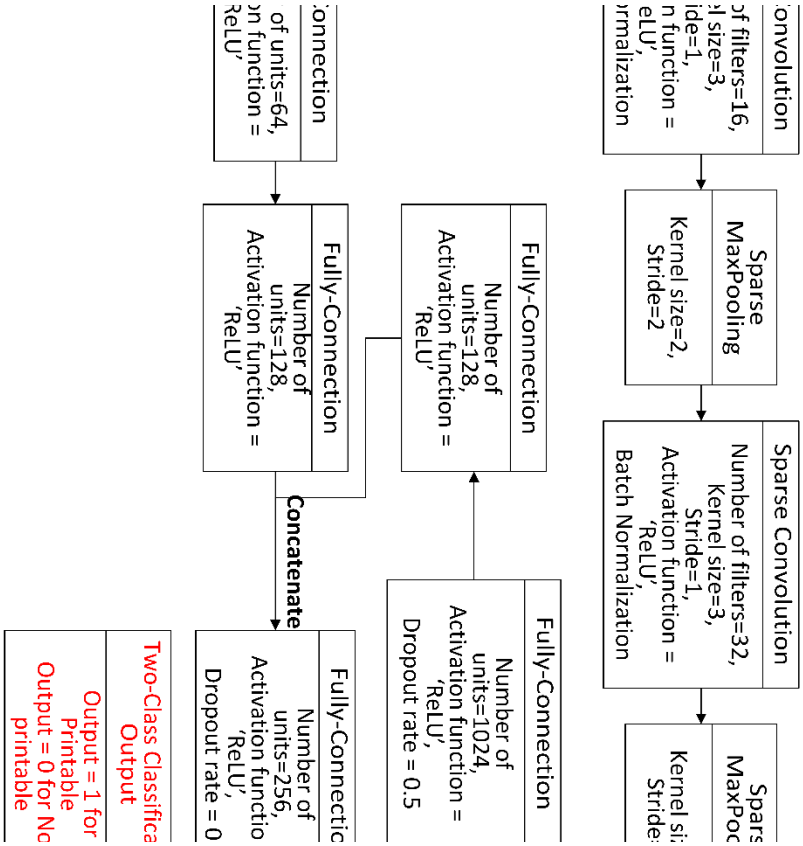

Fig. 6: Network Structure for two-class manufacturability classification

Fig. 7 shows the network structure and major-related hyperparameters for the semantic manufacturability segmentation. The network structure is inspired by the popular ML model, 3D U-net[43]. Different from the network structure in 3D U-net, our model has two different inputs: design aspects and material and processing aspects. The two inputs are firstly treated by separatable ML operations and then join together to make the prediction of the printability map.

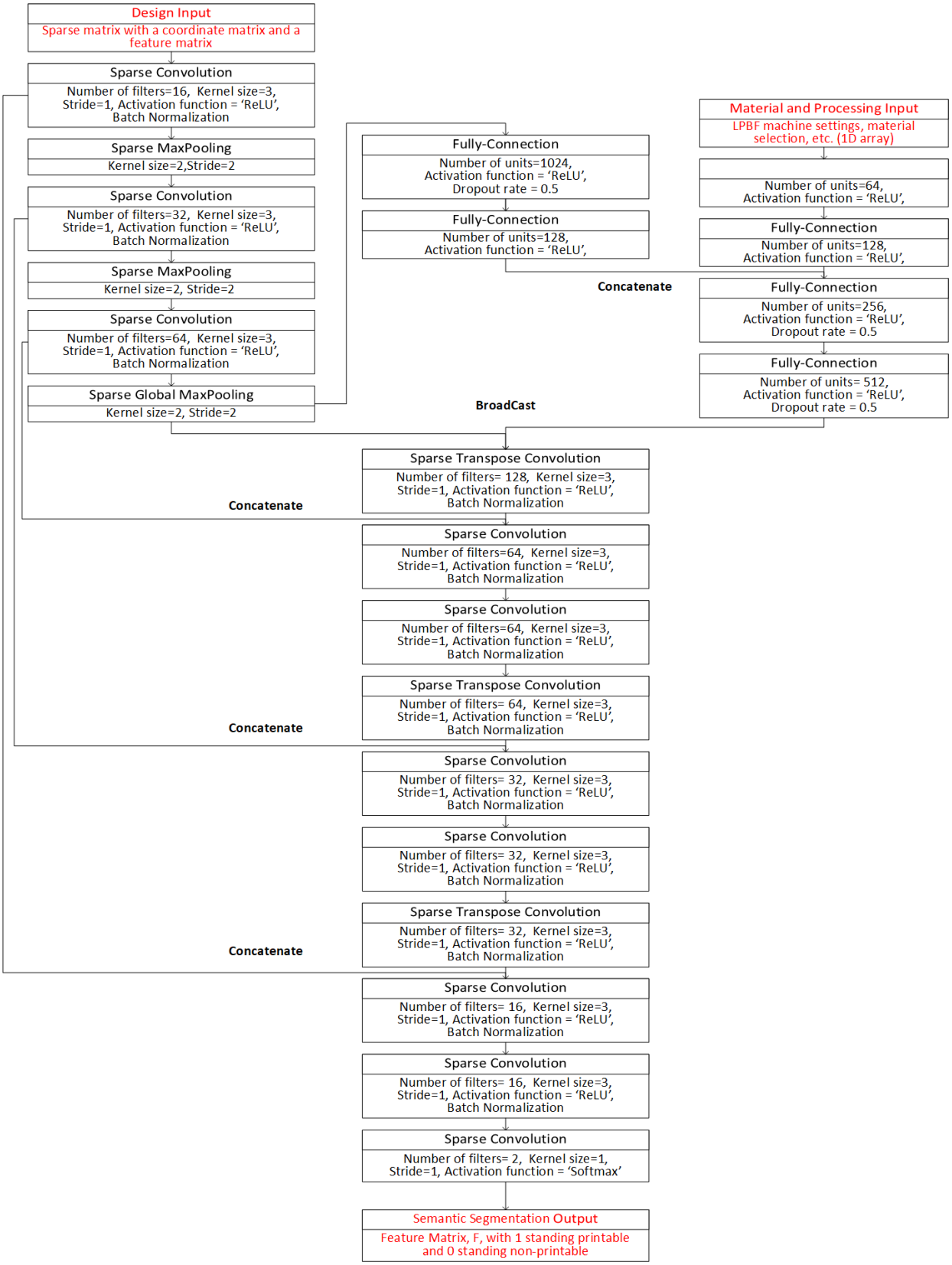

Fig. 7: Network Structure for semantic manufacturability segmentation

In this study, the learning rate is set to be 1e-4, the batch size is set to be 4, and the epoch number is set to be 40 in order to converge the loss function. Those learning parameters are manually tuned to find the best performance for the current dataset.

4. Results and Discussions

The dataset used in this research is split into a 4:1 ratio as a training dataset and validation dataset. The training dataset is used to train the ML model, and the validation dataset is used to validate the ML model. The five-fold cross-validation is used for each training to avoid the potential overfitting issue. In the rest of this section, results in predicting whether the given design is printable or not are discussed along with the generated printability map. It demonstrates that the developed ML model is capable to analyze manufacturability accurately with better performance compared to the literature.

The developed models are implemented in Python 3.7 on an NVIDIA GeForce RTX 2080 Ti. Relevant libraries include Scikit-learn [44], PyTorch, and MinkowskiEngine [42].

4.1. Two-Class Manufacturability Classification

Table 1 shows the cross-validation result for the two-class manufacturability classification at the resolution of 128^3. The model performance is evaluated on its accuracy, which is represented as $(TP + TN)/(TP + TN + FP + FN)$, where *TP* indicates true positive, which the prediction is printable and the truth is also printable. *TN* indicates true negative, in which the prediction is non-printable, and the truth is also non-printable. *FP* indicates a false positive, in which the prediction is printable, but the truth is non-printable. FN indicates a false negative, which depicts the opposite results of the *FP*. The loss function is selected as cross-entropy. Cross-entropy is commonly used in ML applications, and it offers the best performance in accuracy for the two-class manufacturability classification. After calculating the average of the cross-validation cases, the accuracy is about 0.9714. In the literature, the researcher did the same investigation in the dense matrix [27], which applies convolution to every element in the matrix. The same dataset is used in both studies. The sparse matrix solution shows a better performance with 0.9714 in accuracy than the dense matrix with 0.8404 in accuracy. Moreover, as the cross-validation shows, the accuracy of each case does not fluctuate a lot, which shows the stability of the proposed model, and the model does not suffer the overfitting issue in the current dataset.

Table 1: Cross-validation result of the two-class manufacturability classification

Cross-validation Case	Accuracy in sparse matrix	Accuracy in dense matrix
1	0.9592	/
2	0.9796	/
3	0.9592	/
4	0.9796	/
5	0.9796	/
Average	0.9714	0.8408

A memory cost comparison between the sparse matrix and dense matrix is demonstrated in Fig. 8 as well. The sparse matrix requires much less memory than the dense matrix for all the voxelization sizes. Note that the sparse matrix can be easily fit into a GPU's memory (12GB) for

the resolution of 128. In contrast, the same-size dense matrix is not able to fit in a single GPU. Computing in the CPU will take more running time.

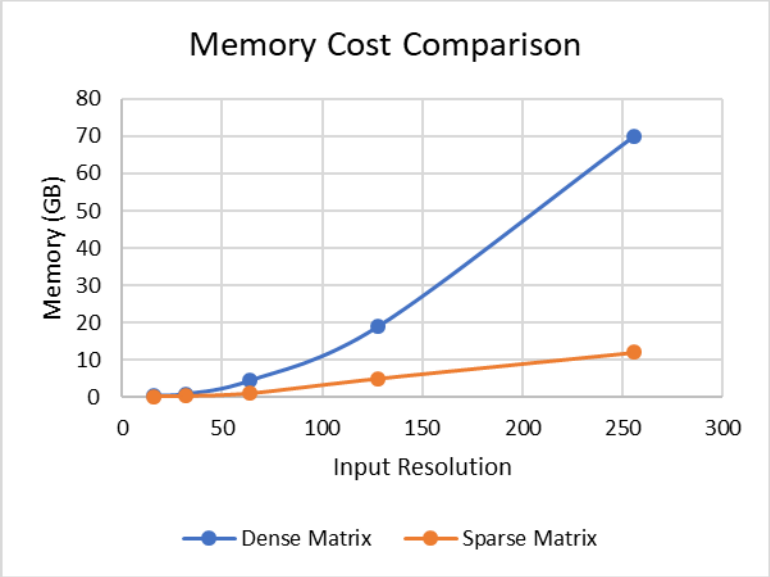

Fig. 8: Memory cost comparison between dense matrix and sparse matrix

Overall, using the sparse matrix input data structures sharply decrease the computational cost and improve the model performance. By only considering the occupied voxels and removing the most useless elements which are background, the model demonstrates its capability in predicting the manufacturability of the given design.

4.2. Semantic Manufacturability Segmentation

The cross-validations are conducted for the semantic manufacturability segmentation as well. For semantic manufacturability segmentation, the mean intersection of union (IoU) is chosen to evaluate the performance. In the semantic manufacturability segmentation, the non-printable class is much less than the printable class, which suffers an imbalance class issue. To solve the imbalance class issue, a loss function of the weighted dice coefficient loss proposed by Zhang et al. [27] is used in this study. For convenience, the weighted dice coefficient is represented as follows:

$$loss = \alpha DC_0 + (1 - \alpha) DC_1 \tag{5}$$

$$DC = -\frac{2TP}{2TP+FP+FN} \tag{6}$$

where DC_i is the dice coefficient for the i-th class. α is the weight coefficient to balance two classes. α is set to be 0.6 in this study to emphasize the importance of the non-printable class. A set of selections for α includes 0.55, 0.7, 0.8 has been tested as well. 0.6 is chosen because of the best performance. The background, which is the empty class, is not stored in the sparse matrix so that it is not trained in the sparse convolutional neural network. Therefore, the background keeps the same as the original, which is counted as 100% correct to compute the mean IoU. Table 2

shows the cross-validation result of IoU for each class, as well as the mean IoU, on the resolution of 128.

Table 2: Cross-validation result of the semantic manufacturability segmentation

Iteration	IoU_mean	IoU_empty	IoU_print	IoU_non_print
1	0.8386	1	0.7964	0.7195
2	0.8366	1	0.8217	0.6882
3	0.8359	1	0.8105	0.6972
4	0.8293	1	0.7865	0.7013
5	0.8382	1	0.8153	0.6994
Average	0.8357	1	0.8061	0.7011

Table 3 compares the proposed ML model to the literature approach on the same dataset. Our result clearly shows a better performance than the dense matrix. Especially for the non-printable class, the sparse CNN demonstrates a better predictive result than the convolution on dense matrix.

Table 3: Comparison between the dense matrix and sparse matrix

	Average IoU_mean
Sparse Matrix	0.8357
Dense Matrix [27]	0.7951

Qualitative results for two samples are provided in Fig. 9 and Fig. 10 to demonstrate the capability of the proposed manufacturability prediction. Fig. 9 is an example of the solid lattice hybrid structure printed through a Renishaw machine with the material selection of AlSi10Mg. The original design is demonstrated in Fig. 9a. It is predicted as fully printable through the proposed ML model as shown in Fig. 9b. Fig. 9c shows the prediction from the commercial software Materialise Magics. Fig. 9d is the experimental result that validates the prediction of the proposed Model. Note that for the figure of the printed result, the support structure has been removed.

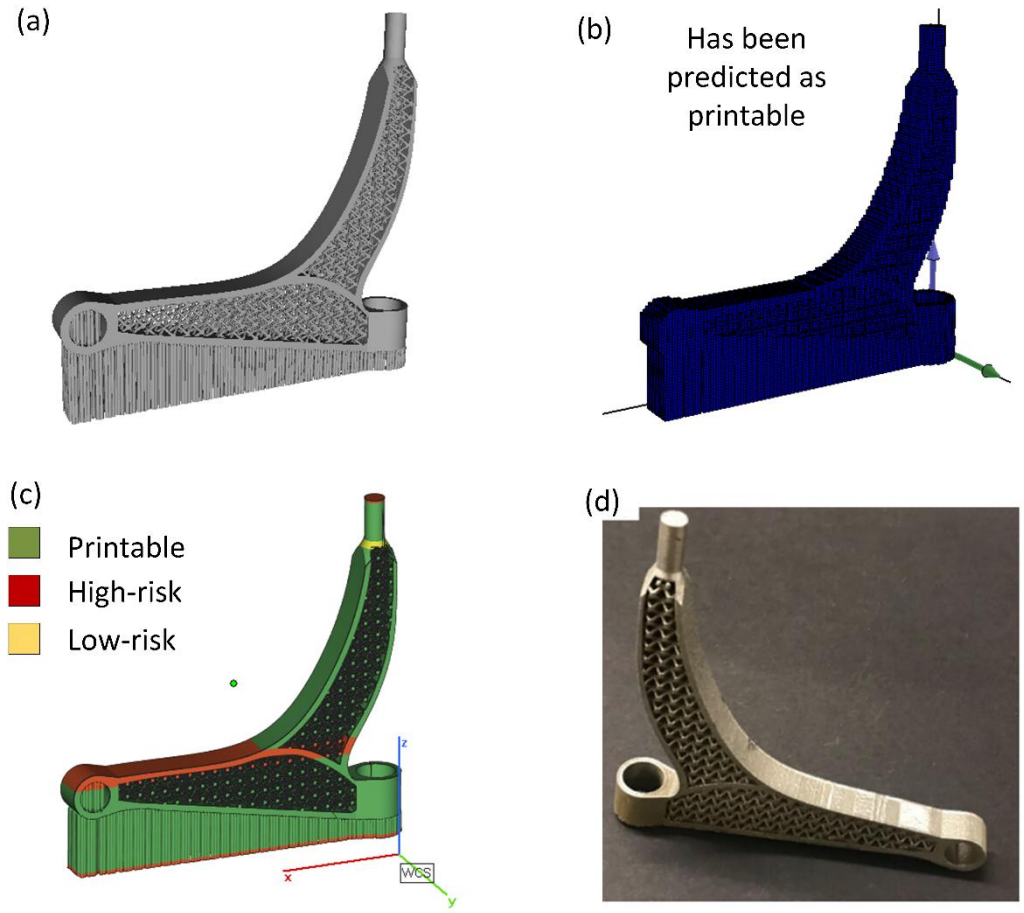

Fig. 9: Comparison of the (a) original design model, (b) prediction from the two-class manufacturability classification, (c) prediction from the commercial software Materialise Magics, and (d) the printed result [45].

Another example from Fig. 10 is a thin tensile bar printed through a Renishaw machine with the material selection of AlSi10Mg. Fig. 10b shows the prediction from the proposed ML model. The blue color indicated the printable area, and the red color indicates the non-printable area. The result from the commercial software is shown as the comparison in Fig.10c as well. The entire tensile bar is defined as non-printable since it suffers a warping issue, as shown in Fig.10d. It is clearly shown that the prediction from the proposed ML model is much better than the prediction from the commercial software. Note that for the figure of the printed result, the support structure has been removed.

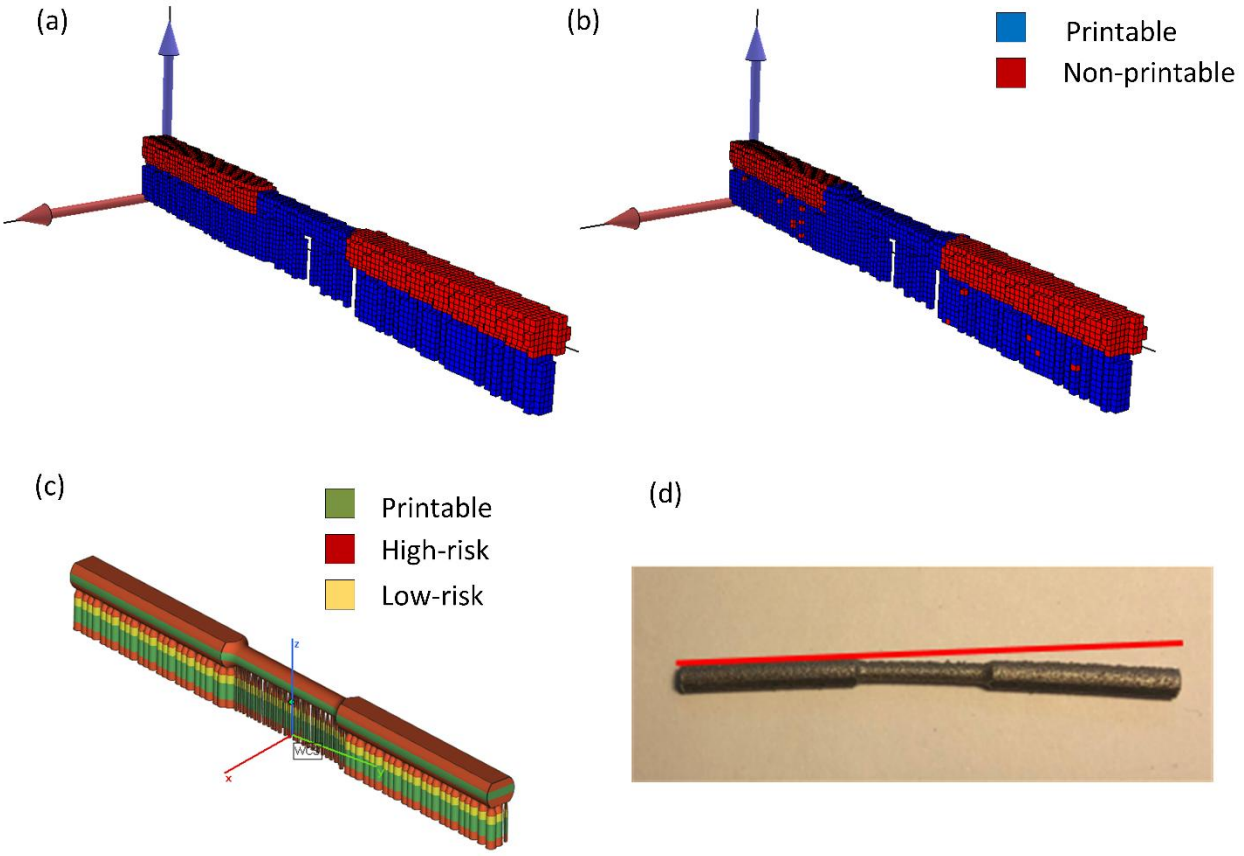

Fig. 10: Comparison of the (a) ground truth (labeling), (b) prediction from the semantic manufacturability segmentation, (c) prediction from the commercial software Materialise Magics, and (d) the printed result.

5. Conclusions and Future Perspective

This paper provides a novel approach on applying a hybrid sparse CNN model for manufacturability analysis. A generalized convolution operation is applied to the sparse matrix to reduce the computational cost and improve the model performance. The results demonstrate the effectiveness and efficiency of the proposed model in predicting the manufacturability of the given design. It also shows a better result compared to the literature. In this study, the methodology is evaluated in the LPBF process. The network structures and learning parameters are tuned to fit the best of the current dataset. However, the framework can also be adapted to other AM techniques with simple modifications to the parameters of ML models. In addition, the ML model proposed in this paper is targeted to predict manufacturability level 1 which is the visual defects. With the desired dataset and similar ML models, it is also expected to predict the other manufacturability levels including dimensional accuracy and microstructures that will be in future works.

There are also some drawbacks to this study. First, the dataset is still limited, which is a common issue in most ML applications in AM. For training an ML model with acceptable results, most studies suggest to have a large dataset. A small dataset could lead to an overfitting issue so that the proposed model could not represent the entire population. At the current stage, available AM

datasets are very limited. The AM process is even not fully standardized. There is a research gap to establish the data framework specific for AM. What needed to be stored in the dataset needs to be determined and standardized. Second, for the sparse matrix, the progress in generating all the coordinates and features is also time-consuming. To store the sparse matrix, there are more different formats such as dictionary of keys, list of lists, compressed sparse row or column, etc. Octree-based can also be a potential solution that leads to future investigation.

Acknowledgment
Financial support from Natural Sciences and Engineering Research Council of Canada (NSERC) Strategic Network for Holistic Innovation in Additive Manufacturing (HI-AM) with NSERC Project Number: NETGP 494158 - 16 and McGill Engineering Doctoral Award (MEDA) is acknowledged with gratitude.

Reference

1. Wohlers, T., *Wohlers report.* Wohlers Associates Inc, 2014.
2. Gisario, A., et al., *Metal additive manufacturing in the commercial aviation industry: A review*. Journal of Manufacturing Systems, 2019. **53**: p. 124-149.
3. Yi, L., C. Gläßner, and J.C. Aurich, *How to integrate additive manufacturing technologies into manufacturing systems successfully: A perspective from the commercial vehicle industry.* Journal of Manufacturing Systems, 2019. **53**: p. 195-211.
4. Aboulkhair, N.T., et al., *Reducing porosity in AlSi10Mg parts processed by selective laser melting.* Additive Manufacturing, 2014. **1**: p. 77-86.
5. Sufiiarov, V.S., et al., *The effect of layer thickness at selective laser melting.* Procedia engineering, 2017. **174**: p. 126-134.
6. Pegues, J., et al. *Effect of Specimen Surface Area Size on Fatigue Strength of Additively Manufactured Ti-Al-4V Parts*. in *28th International Solid Freeform Fabrication Symposium*. 2017.
7. Pegues, J., et al., *EFFECT OF PROCESS PARAMETER VARIATION ON MICROSTRUCTURE AND MECHANICAL PROPERTIES OF ADDITIVELY MANUFACTURED TI-6AL-4V*.
8. Krauss, H. and M. Zaeh, *Investigations on manufacturability and process reliability of selective laser melting.* Physics Procedia, 2013. **41**: p. 815-822.
9. Kruth, J., et al., *Benchmarking of different SLS/SLM processes as rapid manufacturing techniques.* Laser, 2005. **1**: p. 3D.
10. Adam, G.A. and D. Zimmer, *On design for additive manufacturing: evaluating geometrical limitations.* Rapid Prototyping Journal, 2015. **21**(6): p. 662-670.
11. Thomas, D., *The development of design rules for selective laser melting*. 2009, University of Wales.
12. Zhang, Y., S. Yang, and Y.F. Zhao, *Manufacturability analysis of metal laser-based powder bed fusion additive manufacturing—a survey.* The International Journal of Advanced Manufacturing Technology, 2020: p. 1-22.
13. Qi, X., et al., *Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives.* Engineering, 2019. **5**(4): p. 721-729.

14. Carbonell, J.G., R.S. Michalski, and T.M. Mitchell, *1 - AN OVERVIEW OF MACHINE LEARNING*, in *Machine Learning*, R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, Editors. 1983, Morgan Kaufmann: San Francisco (CA). p. 3-23.
15. Gobert, C., et al., *Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging.* Additive Manufacturing, 2018. **21**: p. 517-528.
16. Scime, L., et al., *Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation.* Additive Manufacturing, 2020. **36**: p. 101453.
17. Snow, Z., et al., *Toward in-situ flaw detection in laser powder bed fusion additive manufacturing through layerwise imagery and machine learning.* Journal of Manufacturing Systems, 2021. **59**: p. 12-26.
18. Liu, C., et al., *Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication.* Journal of Manufacturing Systems, 2019. **51**: p. 75-86.
19. Yazdi, R.M., F. Imani, and H.J.J.o.M.S. Yang, *A hybrid deep learning model of process-build interactions in additive manufacturing.* Journal of Manufacturing Systems, 2020. **57**: p. 460-468.
20. Li, Z., et al., *Prediction of surface roughness in extrusion-based additive manufacturing with machine learning.* Robotics and Computer-Integrated Manufacturing, 2019. **57**: p. 488-495.
21. Cerda-Avila, S.N., H.I. Medellín-Castillo, and T. Lim, *An experimental methodology to analyse the structural behaviour of FDM parts with variable process parameters.* Rapid Prototyping Journal, 2020.
22. Guo, Y., W.F. Lu, and J.Y.H. Fuh, *Semi-supervised deep learning based framework for assessing manufacturability of cellular structures in direct metal laser sintering process.* Journal of Intelligent Manufacturing, 2020: p. 1-13.
23. Mycroft, W., et al., *A data-driven approach for predicting printability in metal additive manufacturing processes.* Journal of Intelligent Manufacturing, 2020: p. 1-13.
24. Shalev-Shwartz, S. and S. Ben-David, *Understanding machine learning: From theory to algorithms*. 2014: Cambridge university press.
25. Rojas, R., *Neural networks: a systematic introduction*. 2013: Springer Science & Business Media.
26. Hassanin, H., et al., *Controlling the properties of additively manufactured cellular structures using machine learning approaches.* Advanced Engineering Materials, 2020. **22**(3): p. 1901338.
27. Zhang, Y., et al., *Predictive manufacturability assessment system for laser powder bed fusion based on a hybrid machine learning model.* 2021. **41**: p. 101946.
28. Yang, S., et al., *Towards an automated decision support system for the identification of additive manufacturing part candidates.* Journal of Intelligent Manufacturing, 2020: p. 1-17.
29. Khadilkar, A., J. Wang, and R. Rai, *Deep learning–based stress prediction for bottom-up SLA 3D printing process.* The International Journal of Advanced Manufacturing Technology, 2019. **102**(5): p. 2555-2569.
30. Ayeb, M., M. Frija, and R. Fathallah, *Prediction of residual stress profile and optimization of surface conditions induced by laser shock peening process using artificial*

neural networks. The International Journal of Advanced Manufacturing Technology, 2019. **100**(9): p. 2455-2471.
31. White, D.A., et al., *Multiscale topology optimization using neural network surrogate models.* Computer Methods in Applied Mechanics and Engineering, 2019. **346**: p. 1118-1135.
32. Zhang, J., P. Wang, and R.X. Gao, *Deep learning-based tensile strength prediction in fused deposition modeling.* Computers in industry, 2019. **107**: p. 11-21.
33. Scime, L. and J. Beuth, *Using machine learning to identify in-situ melt pool signatures indicative of flaw formation in a laser powder bed fusion additive manufacturing process.* Additive Manufacturing, 2019. **25**: p. 151-165.
34. Usama, M., et al., *Unsupervised machine learning for networking: Techniques, applications and research challenges.* IEEE Access, 2019. **7**: p. 65579-65615.
35. Arulkumaran, K., et al., *Deep reinforcement learning: A brief survey.* IEEE Signal Processing Magazine, 2017. **34**(6): p. 26-38.
36. Min, P., *Binvox 3D mesh voxelizer.* Available on: http://www. cs. princeton. edu/~ min/binvox, 2004.
37. Hiller, J. and H. Lipson, *Dynamic simulation of soft multimaterial 3d-printed objects.* Soft robotics, 2014. **1**(1): p. 88-101.
38. Graham, B., *Fractional max-pooling.* arXiv preprint arXiv:1412.6071, 2014.
39. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.
40. Dumoulin, V. and F. Visin, *A guide to convolution arithmetic for deep learning.* arXiv preprint arXiv:1603.07285, 2016.
41. Xu, L., et al. *Deep convolutional neural network for image deconvolution*. in *Advances in neural information processing systems*. 2014.
42. Choy, C., J. Gwak, and S. Savarese. *4d spatio-temporal convnets: Minkowski convolutional neural networks*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
43. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.
44. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python.* the Journal of machine Learning research, 2011. **12**: p. 2825-2830.
45. Dong, G., et al., *Design and optimization of solid lattice hybrid structures fabricated by additive manufacturing.* Additive Manufacturing, 2020. **33**: p. 101116.