

Deep Learning-based Semantic Segmentation of Machinable Volumes for Cyber Manufacturing Service

Xiaoliang Yan^a, Reed Williams^b, Elena Arvanitis^b, Shreyes Melkote^{a,*}

^a*George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, 30332*

^b*Siemens Corporate Technology, 755 College Rd E, Princeton, NJ, 08540*

* Corresponding author. Tel.: +1-404-894-8499; fax: +1-404-894-9342. E-mail address: shreyes.melkote@me.gatech.edu

Abstract

Enabling the vision of on-demand cyber manufacturing-as-a-service requires a new set of cloud-based computational tools for design manufacturability feedback and process selection to connect designers with manufacturers. In our prior work, we demonstrated a generative modeling approach in voxel space to model the shape transformation capabilities of machining operations using unsupervised deep learning. Combining this with a deep metric learning model enabled quantitative assessment of the manufacturability of a query part. In this paper, we extend our prior work by developing a semantic segmentation approach for machinable volume decomposition using pre-trained generative process capability models, which output per-voxel manufacturability feedback and labels of candidate machining operations for a query 3D part. Using three types of complex parts as case studies, we show that the proposed method accurately identifies machinable and non-machinable volumes with an average intersection-over-union (IoU) of 0.968 for axisymmetric machining operations, and a class-average F1 score of 0.834 for volume segmentation by machining operation.

Keywords: semantic segmentation, unsupervised learning, convolutional neural network, process planning, feature recognition

1. Introduction

With the rise of digital transformation and cloud computing in the manufacturing industry, on-demand cyber manufacturing services have begun to emerge with the goal of connecting geographically distributed designers with manufacturers through online interactive design tools and instant job quotes [1]. Because custom parts may prove technically or economically impractical to fabricate, there is a need for early-stage cloud-based manufacturability analysis and automated process selection. While commercially available computer aided design (CAD) tools often include design for manufacturability (DfM) modules [2], such a rule-based manufacturability checker is only developed for generic manufacturability rules, which do not factor in the supplier's actual manufacturing capabilities. In addition, DfM modules often require significant processing time to evaluate a query part, which makes them undesirable for deployment in a cloud-based environment.

Automated manufacturability analysis and process selection are long-standing challenges at the forefront of computer-aided process planning (CAPP), which seeks to bridge the gap between CAD and computer aided manufacturing (CAM), and aims to automatically generate a process plan from a CAD model of the part [3]. A major body of early literature targeting manufacturability analysis and process selection in machining is focused on extracting manufacturing features from 3D CAD models using feature recognition (FR) technologies [4]. The objective of FR is to first recognize a set of known features and then connect these features to manufacturing capabilities. Most recent efforts are focused on developing deep learning (DL) methods to recognize features from different representations of CAD models such as boundary representation [5, 6], surface representation [7, 8], and volumetric representation [9-11]. For instance, Colligan et al. [5] presented a hierarchical graph neural network (GNN) to identify machining features. Wang and Rosen [8, 12] developed a process classification method using Heat Kernel Signature and point cloud-based convolutional neural network (CNN). Zhang et al. [10] proposed FeatureNet that utilized a voxel-based 3D-CNN to recognize manufacturing features. Ning et al. [13] presented a hybrid machining feature recognition approach that combines graphs with 3D-CNN. Zhang et al. [14] developed an artificial bee colony algorithm with a neural network to recognize features in a STEP-NC part. Recent advances in object detection using methods such as single-shot multibox detector (SSD) [15, 16] and semantic segmentation [17, 18] have also permeated into FR. Shi et al. [19] presented an SSD-based machining feature localization and recognition network. Zhang et al. [20] proposed a

multi-task neural network using instance and semantic segmentation to identify machining features. While the feature-based approaches have demonstrated varying levels of success in FR, the definition of features is ambiguous across design and manufacturing systems, which adds to the challenge of determining a suitable manufacturing process for a given feature. Because a feature may be produced by more than one manufacturing process, feature-to-process matching is not trivial. In addition, it has been reported that complex/intersecting features can be difficult to identify [21].

Volumetric decomposition is a notable approach proposed to resolve the challenges in identifying complex/intersecting features in a query part [22]. Instead of identifying a clearly separate feature, cellular decomposition [23], convex hull decomposition [24], and maximal feature decomposition [25] have been proposed to convert the delta volume, which is the Boolean difference between a raw stock and a complex query part, into identifiable manufacturing features. Prior works have demonstrated the capability of fitting a sequence of manufacturing features to a delta volume, from which a “machinable volume” is identified [26]. Because machining is a subtractive process, “machinable” implies that volumes are subtracted from a raw stock and arranged in an order, such that the volumes are accessible for machining after the preceding volume has been removed [27]. However, critics argue that volumetric decomposition approaches may result in an excessive number of manufacturing features from which determination of the process plan is impractical [28]. Recent work by Nelaturi et al. [29] and Behandish et al. [28] on feature-free spatial planning of machining and hybrid manufacturing has focused on developing a systematic methodology to identify process primitives of a given part using filter-based sweeping operations of “manufacturing capability” that are pre-defined for a specific cutting tool/nozzle based on the allowable motion and tool/nozzle shape. Using the process primitives on a query part, a Boolean enumeration is employed to identify the machinable volumes and candidate processes [28, 29]. However, a disadvantage of this approach is the difficulty in pre-defining the manufacturing capability of a process. Manufacturing capabilities of a large number of processes can be difficult to determine and hard-code, as well as computationally expensive to permute for identifying a process primitive.

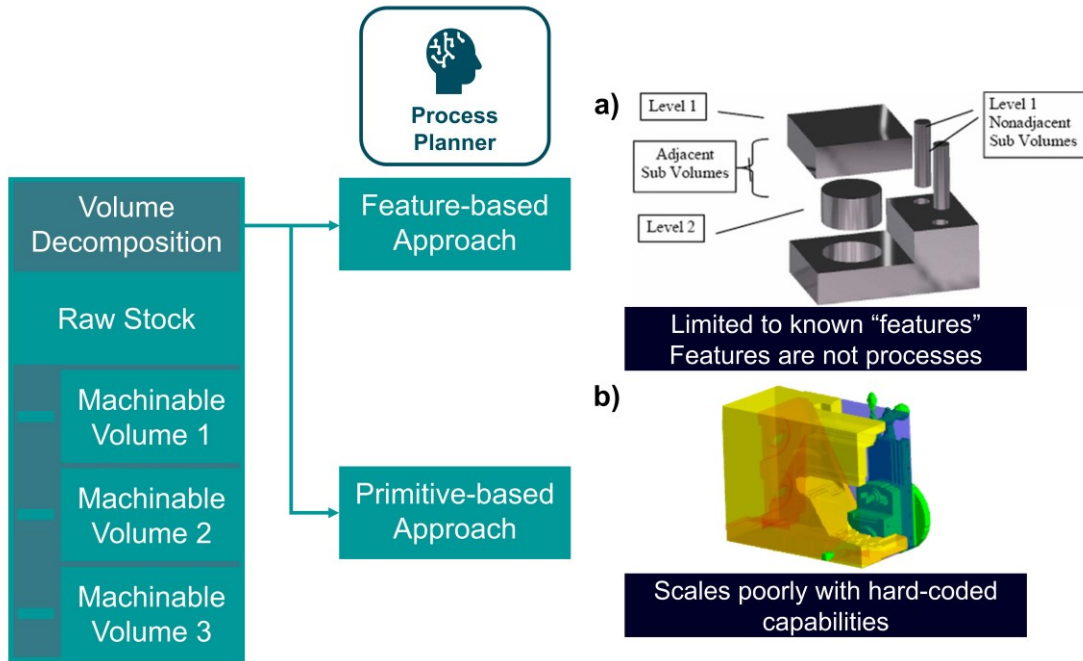


Figure 1. Representative literatures in volumetric decomposition for process planning automation: a) feature-based approach [26] and b) primitive-based approach [29].

As shown in Figure 1, while feature-based approaches have demonstrated success in recognizing geometric features in a CAD model, manufacturing processes must still be automatically assigned to the recognized features to realize the process selection step in automated process planning. For instance, the sub volumes shown in Figure 1(a) do not represent manufacturing processes directly, but rather are assumed to be manufacturable by some known processes. Therefore, a significant research gap still exists in bridging design and process planning to enable process planning automation. In the context of cyber manufacturing, a set of cloud-based computational tools are necessary for efficient manufacturability feedback and automated process selection. Here, we envision a platform-based process

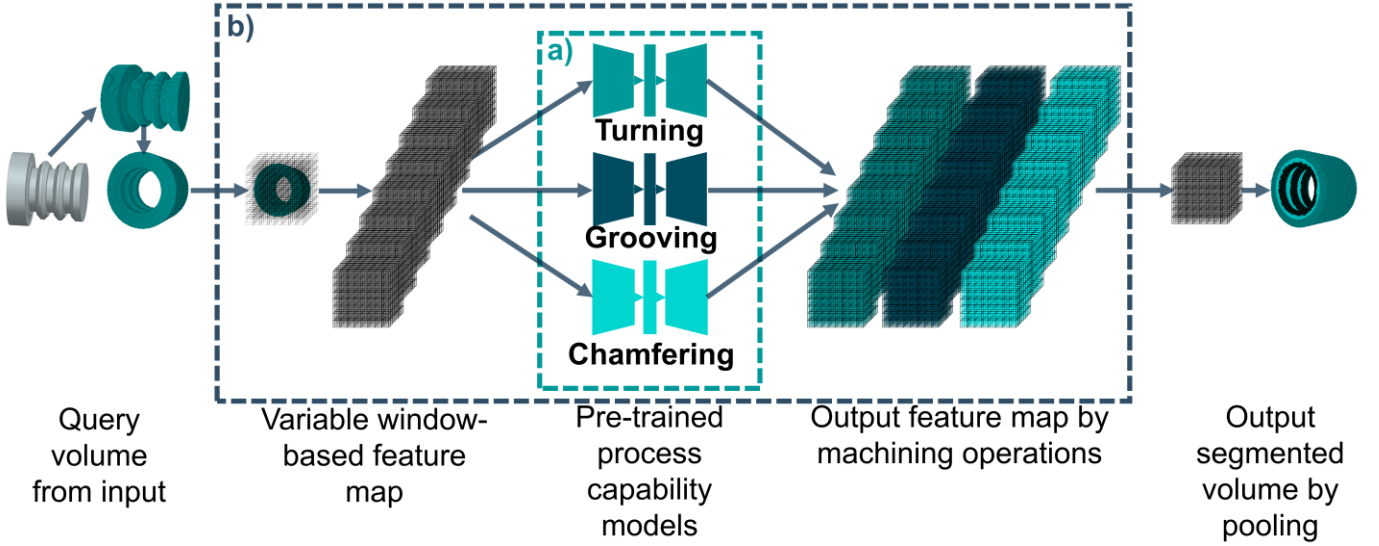


Figure 2. Overview of the proposed generative semantic segmentation approach with a) pretraining of process capability models, and b) a window-based feature scanning algorithm.

planning tool that visualizes manufacturability of a given design and suggest candidate machining processes in an interactive fashion [30]. To this end, we developed a generative modeling approach in voxel space to model the machining process/operation capabilities using unsupervised deep learning [31]. It was further demonstrated that combining a deep metric learning approach with data-driven process capability models enables quantitative assessment of manufacturability of a query part [32]. However, utilizing the data-driven machining process capability models for identification of candidate machining operations for producing complex query parts with multiple features remains a challenge. Therefore, in this paper we extend the methodology of process capability modeling to answer the following questions: (1) How can we efficiently identify machinable volumes using process capabilities learned from machining data? and (2) how can we assign candidate machining process labels to the identified machinable volumes? These questions are answered through a semantic segmentation approach for machinable volume decomposition using deep generative pre-trained process capability models. For purposes of illustration, we limit our scope to machining operations performed on a lathe, namely turning, grooving and chamfering. However, the methodology is general and can be extended to other machining operations as well. Figure 2 shows the workflow of the proposed generative semantic segmentation approach. The process capability models, along with a deep metric learning model, are trained on a synthetic dataset consisting of machinable volumes generated by turning, chamfering, and grooving operations. An iterative window-based feature scanning algorithm is developed to identify a machinable volume and to assign machining operation labels to each voxel contained in the volume. The performance of the generative semantic segmentation approach for machinable volume decomposition is evaluated through tests on eleven complex cases of three different types.

The rest of the paper is organized as follows. Section 2 introduces the generative semantic segmentation of machinable volumes by leveraging the pre-trained machining process capability models. Section 3 presents the results of three types of case studies with eleven complex parts and three realistic parts from a public repository. The paper concludes with a discussion of the effectiveness and limitations of the proposed generative semantic segmentation approach and provides recommendations for future work.

2. Generative semantic segmentation

2.1 Generative machining process capability modeling for manufacturability analysis and process selection

In our prior work, we presented three separately trained deep unsupervised learning models to learn the shape transformation capabilities of turning, grooving, and chamfering, which are lathe-based machining operations [31]. It was demonstrated that the shape transformation capability of a machining process can be learned as a latent probability distribution from voxelized training data. However, a limitation of this approach is that a human must be incorporated into the computational loop to *manually* assess manufacturability and perform process selection through visualization. To address this limitation, we extended the generative modeling approach by incorporating deep metric learning to

enable quantitative evaluation of the similarity of the part shapes output by the pre-trained machining processes capability models in response to the query part shapes [32]. Specifically, an autoencoder-Siamese neural network (AE-SNN) based generative model was developed and shown to outperform discriminative models trained on the same dataset in manufacturability analysis and process selection [32]. Because generative models capture the underlying manufacturing process capability and not a discriminative boundary between classes, the AE-SNNs do not require unmanufacturable parts in the training dataset for accurate manufacturability analysis, which is an advantage over traditional classification models since unmanufacturable parts are typically not available. Figure 3 shows the architecture of the AE-SNN adopted in this work.

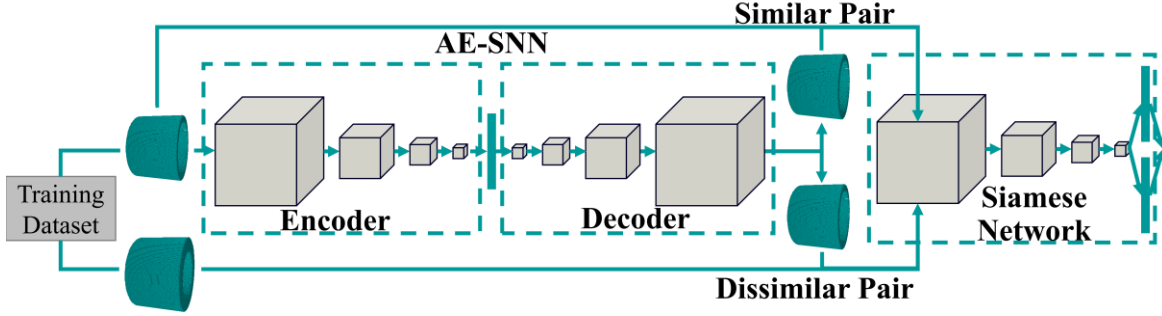


Figure 3. The architecture of AE-SNN, which consists of Encoder, Decoder and Siamese neural networks, all of which use 3D-CNN as their backbone.

In this work, we investigate the possibility of developing a semantic segmentation approach for machinable volume identification to evaluate manufacturability and to identify machining operations for complex part geometries that require more than one operation to produce them. As an extension to AE-SNN’s ability to determine manufacturability and select an operation for a query part feature, the outputs of the semantic segmentation approach are the manufacturability of and machining operations for the machinable volumes in the voxelized query part.

2.2 Synthetic data generation and processing

We created three synthetic datasets consisting of axisymmetric 3D CAD models of parts machinable by turning, grooving, and chamfering operations carried out on a lathe. These part shapes in the dataset generally follow the basic feature generation approach described by Peddireddy et al. [33]. The starting geometry of parts in all three datasets was a solid cylinder. We set the length of the cylinder to 100 mm, while the diameter was varied from 10 mm to 100 mm to produce the parts in the dataset. As shown in Figure 4, each part was parameterized using feature size and feature position parameters. It is evident that turning, grooving, and chamfering require different number of parameters. In addition, step turning, taper turning and profile turning are considered in this work as subsets of the turning operation. For profile turning, both convex and concave shapes were generated for training. In all, 1600 models for step turning, 1226 models for taper turning and 376 models for profile turning were generated, totaling 3202 parts for the turning operation. Similarly, we generated 100 parts for grooving and 64 parts for chamfering operations. A 70-30 data distribution was used for the training-validation split. The CAD models were generated automatically using a Siemens NX macro. We used binvox [34], a voxelizer library, to convert the generated CAD models into their voxelized representations with a resolution of $128 \times 128 \times 128$.

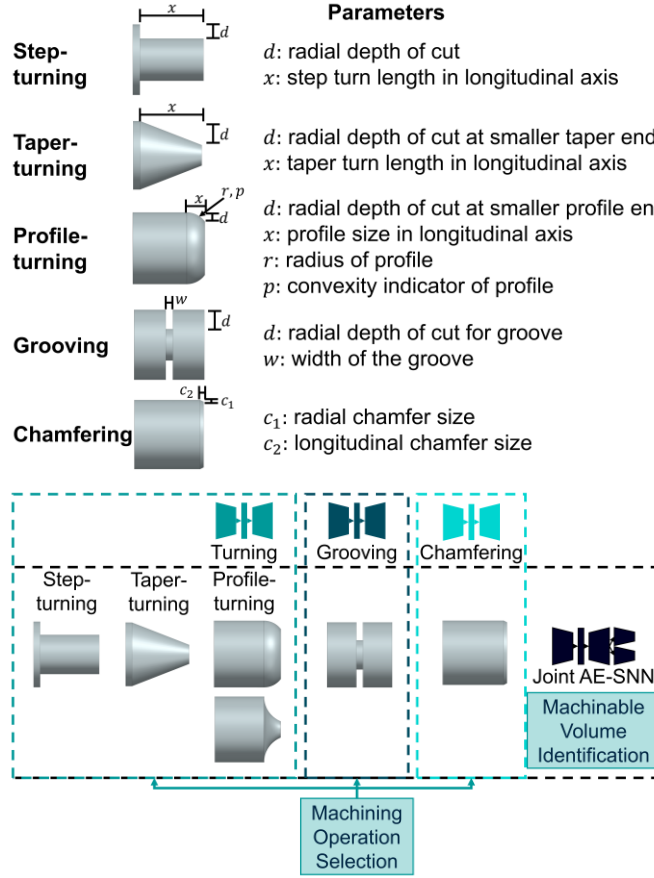


Figure 4. Synthetic data generation based on parameters for turning, grooving, and chamfering operations.

For each voxelized part in the dataset for a machining operation, the corresponding delta volume was determined by subtracting the voxelized part from a voxelized solid cylinder of the same diameter as the part. The delta (or machinable) volumes for each operation were used to train an autoencoder (AE) model for that operation, resulting in three AE models for the three machining operations. In addition, an AE-SNN model was also trained on the combined dataset of delta volumes for all three operations.

2.3 AE-SNN Architecture and Training

The specific model architecture and layer constructions of the AE-SNN follow that reported in [32]. As shown in Figure 4, three encoder and decoder pairs were trained (one each for turning, grooving, and chamfering), while an AE-SNN model was trained on the combined dataset. The encoder takes voxelized delta volumes from the training dataset as input to learn the corresponding latent probability distribution of the shape transformation capability of the machining operation, while the decoder generates realistic 3D shapes that can be produced by the operation, and the SNN outputs a similarity score between 0 and 1 for a pair of input delta volumes. Each voxelized input volume has dimensions of $128 \times 128 \times 128$. The encoder has four 3D-convolutional (Conv3d) layers followed by the Leaky rectified linear unit (LReLU) activation function, defined as $\text{LReLU}(x) = \max(x, \gamma x)$, where γ is set to 0.2. After the convolutional layers, a fully connected layer outputs a latent vector with dimension of 128×1 . The decoder closely mirrors the encoder. The first fully connected layer expands the dimension to 262144×1 , which is followed by four 3D-transpose-convolutional (ConvTranspose3d) layers. The Sigmoid activation function, defined as $\sigma(x) = [1 + e^{-x}]^{-1}$, is applied to enforce a value between 0 and 1 for each voxel in the $128 \times 128 \times 128$ output. The SNN has a network structure similar to the encoder with an additional cosine similarity layer and the Tanh activation function, which is defined as $T(x) = (e^x - e^{-x}) / (e^x + e^{-x})$. Because Tanh outputs a value between -1 and 1, we linearly transform the output by $S = (T(x) + 1) / 2$ to ensure that the output is between 0 and 1. Note that both similar and dissimilar pairs of samples are required to train the SNN. In the training dataset, a similar pair is composed of an anchor and a positive, and a dissimilar pair comprises the same anchor and a negative. For a given input, the output of the decoder is used as the anchor, the input is used as the positive, and a different sample randomly drawn from the

training dataset is used as the negative.

For a typical encoder and decoder pair, a simple reconstruction loss function is defined as follows:

$$||\hat{x} - x||_2 \quad (1)$$

where x is the encoder input, $\hat{x}$ is the decoder output. Note that the prediction of the machinable volume should not exceed the query delta volume, as excess material removal (overcutting) is undesirable; in this work, the reconstruction loss is modified as a weighted reconstruction loss to penalize overcutting as follows:

$$\alpha_1 ||\hat{x}_f - x_f||_2 + ||\hat{x}_t - x_t||_2 \quad (2)$$

where x_f are the voxels with ground truth as false (no delta volume), $\hat{x}_f$ are the corresponding predictions, x_t are the voxels with ground truth as true (contains delta volume), and their corresponding predictions $\hat{x}_t$. In this work, we set α_1 to 5 to increase penalty for overcutting during prediction.

For the SNN, a triplet loss function is used:

$$\alpha_2 \log(1 - S(x_a, x_n)) + \log S(x_a, x_p) \quad (3)$$

where S is the output of the Siamese neural network, and x_a, x_p , and x_n are the anchor, positive, and negative in a triplet, respectively. The value of α_2 in this work is set to 2.

The models were constructed using PyTorch [35] and trained on an HP ZBook workstation with a NVIDIA RTX A2000 GPU. Due to data imbalance among the three machining operation classes, the models were trained for the same number of batches, not epochs. The batch size in this work is one, and each model was trained for 40000 batches, which on average took approximately 8 hours. Note that the training time may improve with more GPUs used in parallel. Adaptive moment estimation (Adam) [36] was used as the optimizer with initial exponential decay rates of the first and second moments of gradient, β_1 and β_2 , set to 0.8 and 0.99, respectively. All learning rates were set to 6×10^{-4} . These training hyperparameters were manually tuned in this work.

The trained AE-SNN and AE models were evaluated by observing the training losses. Figure 5 shows the AE reconstruction loss given by Eq. (1). As training progresses, the reconstruction losses of all four models decrease and converge to a level close to zero after 40000 training batches. Note that the grooving and chamfering training losses had lower initial loss and decreased faster compared to turning model. This is the expected training behavior as the number of data points and data diversity in these two datasets are much lower than the turning dataset. Similarly, the triplet loss converged to a low level after 40000 batches.

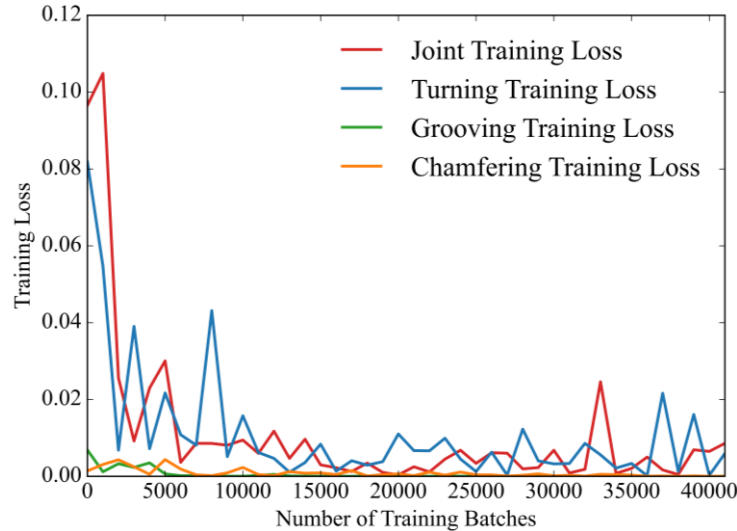


Figure 5. Training reconstruction losses for the combined model, turning model, grooving model, and chamfering model.

Additionally, the classification performance of the trained AE-SNN models were evaluated using the validation dataset, where the AE models were tasked to generate the closest matching output based on the features in the validation

dataset. Figure 6 illustrates the validation confusion matrix, which indicates that the AE-SNN models were able to distinguish among the basic operations in the dataset presented in section 2.2.

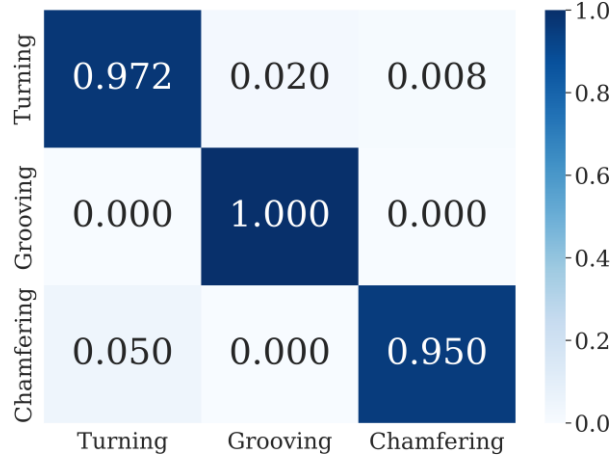


Figure 6. Validation confusion matrix for turning, grooving and chamfering operations.

2.4 Window-based algorithm for feature map generation

In the instance segmentation literature, Region Proposal is commonly adopted to locate objects in a 2D image by scanning with anchor boxes, from which the existence and locations of objects in an image can be determined [37]. Subsequently, an object class label is assigned to the isolated object using a multiclass classification model. Semantic segmentation goes one step further by assigning object class labels to each pixel in the image.

A key challenge in applying semantic segmentation to the manufacturing domain is the lack of pre-segmented training data. In this work, we use pre-trained generative models trained on machinable volumes to scan the query parts, which eliminates the need for a segmented training dataset. Similar to the instance and semantic segmentation literature, where the metric of intersection-over-union is used with a threshold, th , to isolate the locations of potential objects, here we assign a similarity score, S , to each isolated anchor window for outputs of the pre-trained process capability models. A voxel-space merging operation is used across the feature maps of a given operation to merge the similarity scores for each voxel in a query delta volume. An argmax operation is used to assign a process class label to each voxel across the three pre-trained models of turning, grooving and chamfering operations. For semantic segmentation, a noise reduction post-processing operation based on a 26-neighborhood connected component analysis [38] is implemented with a threshold of 100 voxels. The output is a fully segmented query delta volume for each pre-trained machining operation. Specifically, the pretrained generative process capability models for the three machining operations considered in this work, the pretrained Siamese network, the window sizes in voxel space, and the step sizes along the longitudinal axis of the part are inputs to the algorithm. We also ensure that the axes of the input object satisfy the three axes specified in the training dataset, where the first axis is the longitudinal axis. In the initialization step, a placeholder of segmentation logits of the query part size with an additional channel dimension equal to the number of process capability models is created. After that, for each process capability model, the part is scanned using each window size over all step sizes. If the window is not empty, the output of the process capability model is generated and the similarity, S , between the input and output of the generative process capability model is computed. If S is greater than the segmentation logits of the window from the previous step, update the placeholder logits with the new higher output. Here, the window sizes are set to 32 or 16 voxels, whereas the step size is set to 1 voxel.

Figure 7 shows a flowchart for the window-based operation scanning algorithm, which is used to assign similarity scores as segmentation logits. In this work, the dimension of segmentation logits is $3 \times 128 \times 128 \times 128$, where 3 is the channel size equal to the number pre-trained process capability generative models. Argmax is used across the first dimension to obtain integer labels (i.e. 0 = turning, 1 = grooving, 2 = chamfering) representing the corresponding machining operations.

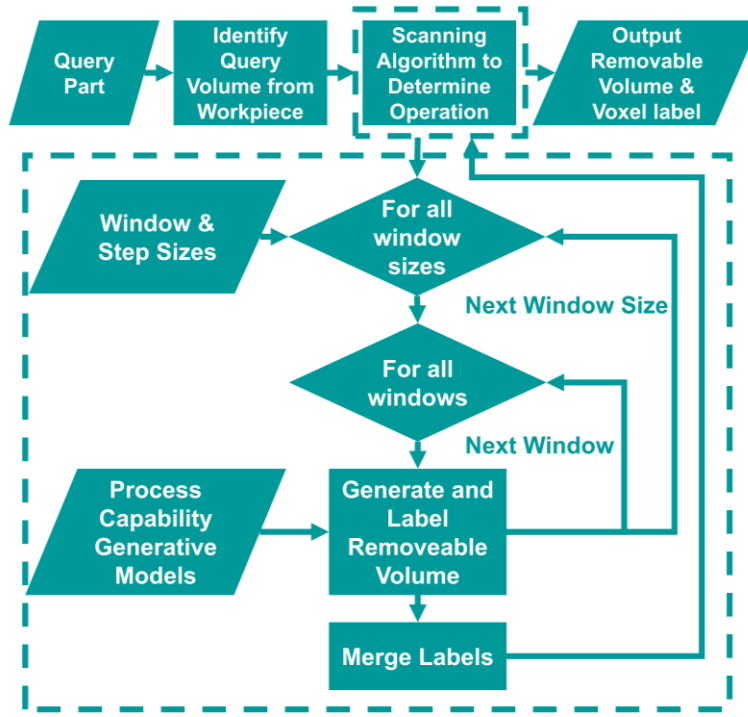


Figure 7. Flowchart of the window-based operation scanning algorithm.

3. Case studies

3.1 Case overview

The objective of the case studies presented here is two-fold: (1) evaluate the effectiveness of the proposed method for machinable volume identification, and (2) assess the performance of semantic segmentation by machining operations. Eleven complex 3D parts belonging to three case types shown in Figure 8 were created and evaluated as query part inputs to the proposed method. Specifically, Type 1 cases include five manufacturable query parts with ground truth machining operation labels; Type 2 cases are curvilinear query parts, which do not have ground truth operation labels based on the training dataset; Type 3 cases contain non-axisymmetric features that cannot be manufactured by the axisymmetric machining operations considered in this work.

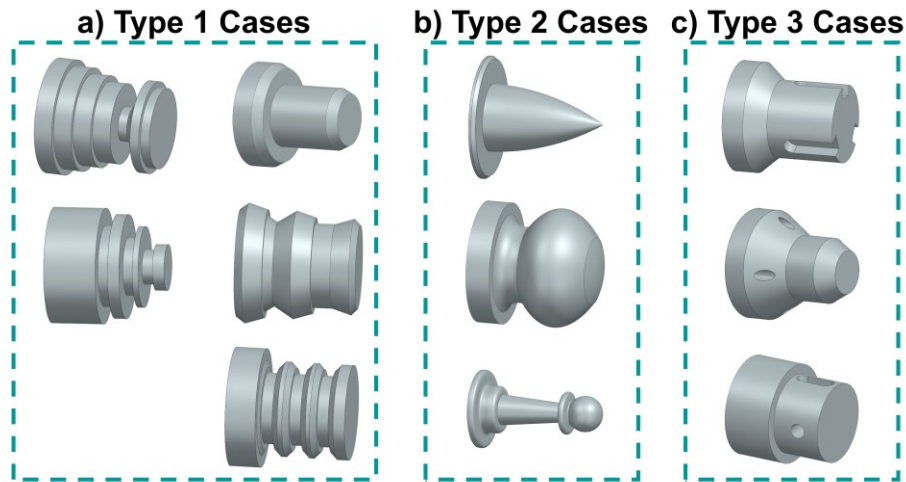


Figure 8. Case study overview: a) Type 1 cases are manufacturable query parts with defined operation ground truths, b) Type 2 cases are curvilinear query parts without ground truth, and c) Type 3 cases are query parts with features unmanufacturable using turning, grooving and chamfering operations.

3.2 Machinable volume identification

Two metrics are used to evaluate the performance of machinable volume identification: intersection over union (IoU) and unmanufacturable feature indicator (UI). IoU is a metric that has been commonly used to evaluate the match between two objects. The value of IoU ranges between 0 and 1, where 1 indicates 100% overlap between the two objects, and 0 indicates the two objects are mutually exclusive. In the voxel-space, IoU is computed by the number of intersecting voxels divided by the number of voxels that are collectively occupied by the two objects. In addition to IoU, UI is a metric based on connected components analysis in voxel space, which evaluates the residual voxels after identified machinable volumes are removed. Two occupied voxels are deemed connected if they coexist in a 26-voxel-neighborhood, which is a $3 \times 3 \times 3$ grid excluding the reference voxel. In this work, we set a threshold of 100 connected voxels to identify the presence of an unmanufacturable feature in the residual.

Metrics:					
Intersection over union (IoU)					
Unmanufacturable feature indicator (UI)					
	Query Part	Voxel Input	Voxel Output	Residual	
a)					0.9873 No
b)					0.9425 Yes

Figure 9. Machinable volume decomposition examples: a) Type 1 example where machinable volumes are fully identified without significant residual, b) Type 3 example with significant residual voxels, which indicates that additional manufacturing operations are needed to fully manufacture the query part.

Figure 9 shows two example case studies for machinable volume identification. Figure 9(a) is a Type 1 case that has ground truth labels for the operations, and the query part is manufacturable based on the ground truth. The IoU for the predicted output and input is 0.984, and the UI for this part is false, which indicates that this part is manufacturable by the axisymmetric operations considered here. Figure 9(b) is a Type 3 case that contains non-axisymmetric query volumes, which in this case are not manufacturable by the axisymmetric operations. The residual of the part shows the volumes that cannot be manufactured. Although the IoU is still quite high (0.943) indicating that the majority of the query part can be manufactured by the modeled machining operations, the UI returns true, which correctly points to the presence of unmanufacturable volumes after removal of the machinable volumes identified by the axisymmetric operations.

Figure 10 shows a chart of inputs, outputs, residuals, and evaluation metrics for machinable volume identification for all eleven complex geometries. With the exception of two Type 1 cases, which are misclassified as containing unmanufacturable volumes, the manufacturability of the remaining nine geometries from different case types are correctly predicted. The average IoU of 0.968 demonstrates that the majority of the delta volumes are manufacturable, which matches intuition as all cases contain mostly axisymmetric volumes. Note that, except for Type 3 cases, all residuals in other cases tend to be thin axisymmetric profiles. This could be due to the lack of similarly thin training data for any machining operation. While in practice such thin profiles maybe achievable, no distinctions can be made between the operations due to the limitations of voxel resolution, e.g., a chamfer and a groove with only one voxel thickness along the longitudinal axis are equivalent in the voxel domain. Such limitation is an inherent disadvantage of utilizing voxel representations for 3D objects in general. Because the total number of voxels scale cubically with increase in the number of voxels in the x, y, z directions, increasing the resolution of the 3D models requires much larger memory and computational capabilities. A possible solution to this limitation is to use a sparse-voxel representation such as an octree. In an Octree-based voxel representation, only voxels near the surface of the object are stored and used in computing and rendering, which significantly reduces the computational and memory cost. Recently, researchers have introduced methods to train deep learning models on sparse-voxel representations [39, 40]. However, how to develop deep learning methods for cyber manufacturing and process planning using hierarchical sparse voxel representation remains an open problem.

Query Part											
Voxel Input											
Voxel Output											
Residual											
IoU	0.9873	0.9848	0.9775	0.9716	0.9666	0.9672	0.9763	0.9515	0.9425	0.9622	0.9621
UI	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes

Figure 10. Machinable volume identification case study. From top to bottom: CAD model of query parts, input query parts in the voxel domain, output reconstructed parts from the generative models in the voxel domain, residual difference between voxel outputs and inputs, IoU and UI as metrics.

3.3 Semantic segmentation by machining operations

In addition to identifying the machinable volumes at the voxel level, another objective is to obtain per-voxel operation labels. Precision, recall, and F1-scores are used as evaluation metrics for semantic segmentation by machining operation. These metrics are defined as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (3)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (4)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Using the turning operation as an example, true positive refers to a correct prediction of turning for a given voxel that matches the ground truth; a false positive means a voxel is falsely predicted as manufacturable by turning, when in fact its ground truth is another operation; and a false negative occurs when a prediction falsely predicts a different operation for the given voxel, when its ground truth is turning. F1 is the harmonic mean of precision and recall, which accounts for both.

These performance metrics require a comparison to ground truth labels. Among the eleven cases, only Type 1 cases contain ground truth labels, which are used to evaluate the semantic segmentation performance. Figure 11 shows two examples of semantic segmentation predictions. Figure 11(a) shows a query part that requires turning, grooving, and chamfering operations based on the ground truth. Qualitative visualization indicates that the prediction correctly identifies these operations at their respective voxel locations. For turning and grooving operations, the F1 scores are 0.962 and 0.976, which suggest that most of the voxels belonging to turning and grooving have been correctly identified. However, the chamfering operation only reached F1 score and recall of 0.785 and 0.649, respectively, which are much lower than for the turning operation.

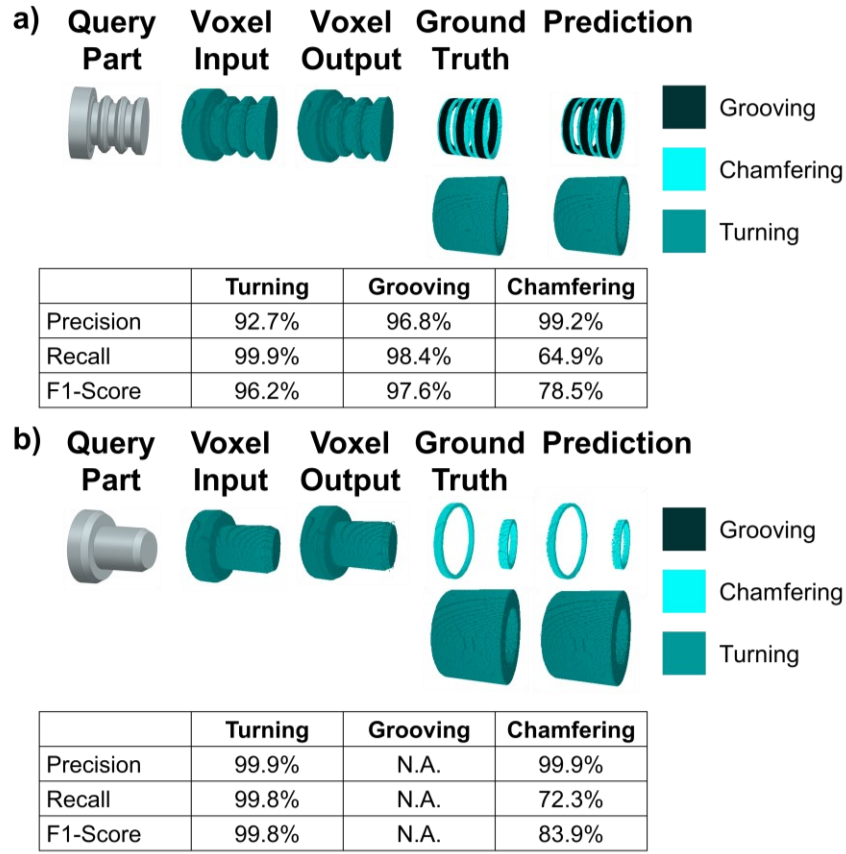


Figure 11. Semantic segmentation examples: a) Type 1 case with predictions for grooving, chamfering and turning operations, b) Type 1 case with only tuning and chamfering operations. In both cases, segmentation metrics for chamfering are the lowest among all operations.

Similar results are observed in Figure 11(b) where the F1 score for turning (0.998) is much higher than the F1 score for chamfering (0.839). Two possible reasons could have led to this result. First, because the volume removed by chamfering is small compared to the volume removed by other operations, any missed prediction has a higher impact on the evaluation metric. Similarly, because of the small volume, the chamfering operation is harder to capture, considering the fixed kernel size of the deep learning model.

Figure 12 shows the semantic segmentation results for all five Type 1 cases. Overall, the semantic segmentation by machining operation reached a class-average F1-score of 0.834. It is evident that turning operations reached high F1 scores for all five parts while the grooving and chamfering operations resulted in lower F1 scores. The same parts with misclassified machinable volumes yielded the lowest grooving and chamfering F1 scores. The result aligns with the intuitive observation that turning removes most of the volume in the cases presented here. Because chamfering and grooving operations remove less material than the turning operation, a higher similarity score is given to turning when considering large continuous volumes. Only when window sizes closely match the volumetric range of grooving and chamfering operations will they score a higher similarity than turning. This is a unique challenge in semantic segmentation for manufacturing. Unlike semantic segmentation for generic objects where discrete objects can be localized via transitions in pixel values, in manufacturing the machinable volumes are connected with few hints to assist in localizing the operation.

Another possible cause of this result is the geometric overlap of operations used in training. For instance, while

Query Part	Voxel Input	Voxel Output	Ground Truth	Prediction	Turning Precision	Turning Recall	Turning F1-Score	Grooving Precision	Grooving Recall	Grooving F1-Score	Chamfer Precision	Chamfer Recall	Chamfer F1-Score
					99.9%	99.8%	99.8%	N.A.	N.A.	N.A.	99.9%	72.3%	83.9%
					86.9%	97.8%	92.1%	98.8%	77.5%	86.9%	N.A.	N.A.	N.A.
					99.8%	88.0%	93.5%	52.4%	89.5%	66.1%	N.A.	N.A.	N.A.
					93.5%	93.9%	93.7%	N.A.	0%	N.A.	99.8%	35.7%	52.6%
					92.7%	99.9%	96.2%	96.8%	98.4%	97.6%	99.2%	64.9%	78.5%

Figure 12. Semantic segmentation case study. From left to right: CAD model of query parts, input query parts in the voxel domain, output reconstructed parts from the generative models in the voxel domain, ground truth labels for each voxel, prediction labels from the generative models, precision, recall and F1-Score for turning, grooving, and chamfering respectively.

the feature sizes of the chamfer and taper turning operations used in training are different, the volumes removed by these operations have the same shape, which may have contributed to misclassification of the operation label. Such geometric overlap is particularly troublesome when the resolution of the voxel representation is relatively low. As noted in Section 3.2, the current voxel resolution of $128 \times 128 \times 128$ cannot be significantly increased due to space and computational time complexity. Small features such as chamfers and grooves cannot be effectively distinguished when useful geometric information is lost in the process of converting the features to dense voxel representations. Future studies on semantic segmentation by manufacturing processes should focus on utilizing deep learning on a hierarchical sparse voxel representation, which has the potential to access geometric differences of features embedded in a much finer voxel resolution.

Furthermore, although we limited the parts in the training dataset to only one ground truth label for each operation, in reality, more than one possible operation can be used to remove a given machining volume. For example, one of the predicted grooves in the third part in Figure 12 is larger in diameter than the ground truth, and, by definition, many voxels are falsely predicted as grooves. However, the solution provided by our method can technically still produce the desired shape. While this solution might not be preferred, it is still valid if volume is the only factor considered. This suggests that there are other considerations such as process precedence constraints or user preference to be included in future research, as process selection is multi-objective in nature.

3.4 Visualizable manufacturability feedback for cyber manufacturing services

With the rise of platform-based manufacturing [30], a designer is presented with a large number of manufacturing resources. It can be tempting to assume that a design can always be manufactured if a manufacturing supplier with the required capability is available. However, not all manufacturing suppliers are equipped with the necessary manufacturing processes/operations for a given design. Our method enables a visualization-based evaluation of the manufacturability of a query design based on available manufacturing operations. Therefore, the designer can interactively evaluate whether a part design containing multiple features is manufacturable by a particular supplier through visualization. If additional operations are required to manufacture the query part, the designer can either choose to reduce the part complexity and conform to the available manufacturing operations or select a different manufacturer with additional process capabilities to create the required features in the part.

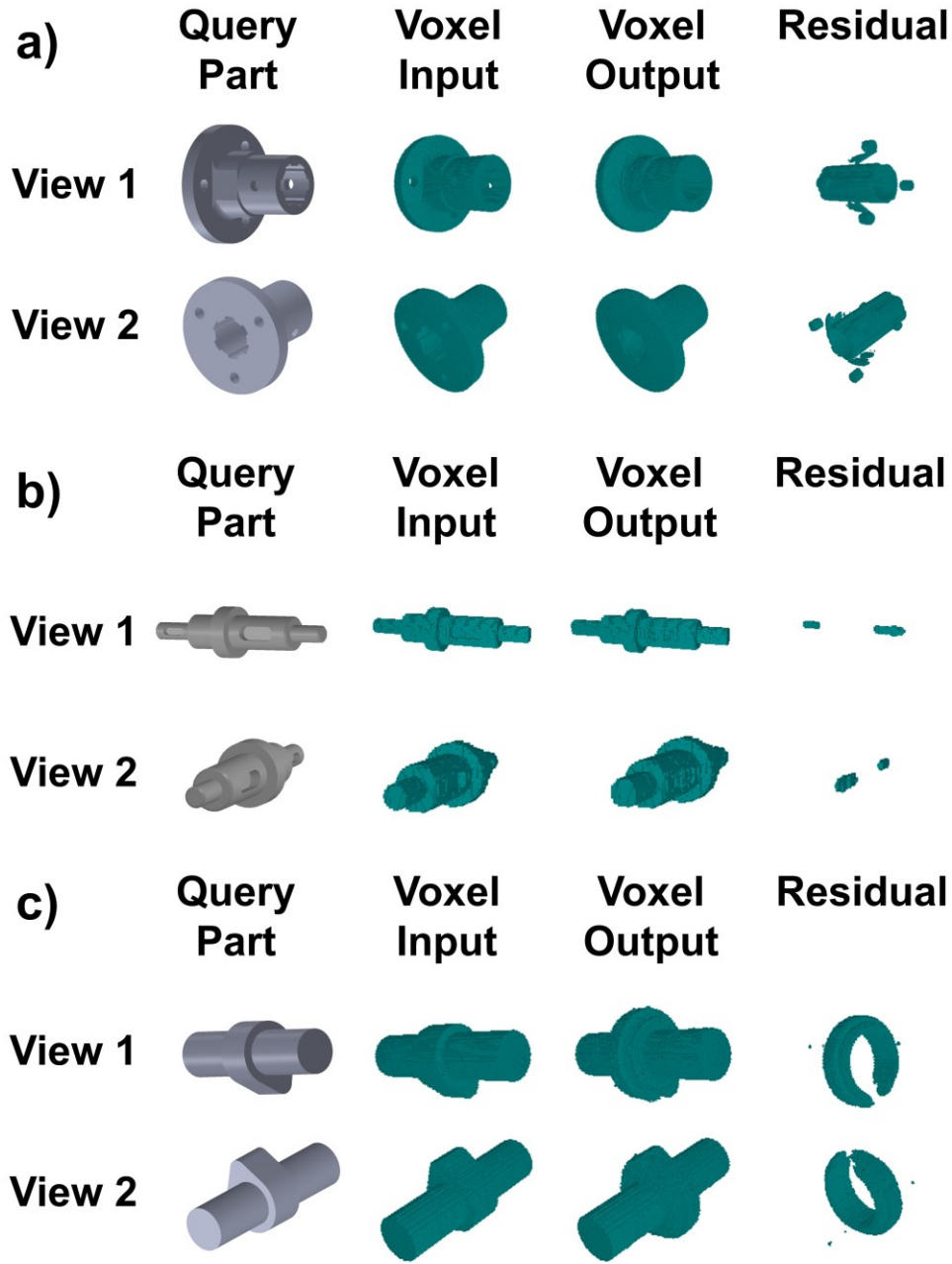


Figure 13. Manufacturability analyses of a) a wheel hub, b) a shaft with key seats, and c) a cam lobe. Unmanufacturable volumes are displayed as residuals.

Figure 13 shows three examples of realistic parts obtained from a publicly available dataset [41-43], which were input to the proposed semantic segmentation method for machinable volume decomposition. Based on the voxel inputs, voxel outputs, and the residuals, it is evident that all three parts contain machinable volumes and non-machinable volumes using only the three axisymmetric machining operations considered in this paper, namely turning, grooving and chamfering. Specifically, in Figure 13(a) our method identified the through holes on the hub body and backing plates as residual, which requires additional machining operations such as drilling. Furthermore, the internal geometric features of the hub were also identified as being part of the residual, which potentially requires specialized processes such as broaching. In Figure 13(b), our model successfully identified the key seats that cannot be manufactured using the axis-symmetric operations. In Figure 13(c), the cam lobe most likely requires an additional process (e.g., grinding) to achieve the specific lobe shape. Therefore, the residual indicates the corresponding volumes that cannot be produced by turning.

It is evident from the above examples that our method is able to identify machinable and non-machinable portions of the query parts, which can provide visualizable manufacturability feedback to the designers based on the generative models for available manufacturing operations. Contrary to discriminative machine learning approaches where a part is classified or segmented into known classes, our method does not assume all features can be manufactured by the available manufacturing resources, and instead, utilizes generative models of the manufacturing operations to determine the probability of recreating the desired machinable volumes. If more than one “equally good” operation exists, they can be provided as available options to choose from, which requires either human-in-the-loop decision making or additional optimization objectives. In the context of a cyber manufacturing service, such manufacturability feedback not only reduces the communication cost between designers and manufacturers, but also enables selection of the required manufacturing resources. For instance, if a machine shop is not equipped with specialized automotive grinding tools for precision machining of cam shafts, our model is able to demonstrate the limitation of the available manufacturing operations (e.g., axis-symmetric operations) to the designers at the early design stage.

4. Conclusion

In this work, we presented a semantic segmentation method for machinable volume decomposition of complex query parts using generative pre-trained neural networks. Through a case study of eleven query parts and three realistic parts from a public repository, we demonstrated that (1) the proposed approach achieves over 96.8% IoU between the voxel input and output pairs for machinable volume identification, (2) volumes in the query part that cannot be removed by axisymmetric machining operations can be identified and visualized in the residual, and (3) the segmentation results for the turning operation are better than for the grooving and chamfering operations, potentially due to the size differences in machinable volumes for these operations and the limited resolution of the voxel representations, which can be potentially addressed through deep learning on sparse voxel representations. These findings have profound implications in the context of cyber manufacturing services. For instance, the proposed approach can assist designers by determining which portion of their design is manufacturable using available manufacturing operations. Through intuitive visualizations, designers modify their designs to target elimination of the residual volumes identified by the proposed method. For process planners, the segmented volumes and visualizations serve as an intuitive assistant to enable initial process/operation selection. It is also conceivable that process planners can leverage such visualizations to plan for a sequence of operations, based on geometric and process precedence constraints.

For future work, a few notable limitations of the current approach should be addressed. First, in addition to volume and shape, the geometric and temporal context of the machinable volumes should be included in training. For example, the current approach does not specify the order of operations, whereas, in reality, some segmented volumes can only be accessed after removing the outer layer of materials (e.g., chamfering can only occur after turning in Figure 11b). In addition, an investigation of deep learning methods for octree-based sparse voxel representation should be conducted in the context of cyber manufacturing and process planning, which has the potential to access features in much finer resolution to improve classification and rendering of small manufacturing features. More complex part geometries and machining operations should be modeled using the proposed approach, e.g., a logical next step is to train models for all material removal operations available in a turn-mill machine tool. Other key manufacturing information such as surface roughness should be considered in the training dataset. Finally, adding supervised learning using pre-segmented training data has the potential to improve the segmentation results.

Acknowledgements

This work was funded by an INTERN Program Supplement to the National Science Foundation (NSF) award #2113672, and the NSF Future Manufacturing award #2229260.

Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of generative AI in scientific writing

The authors declare that no Generative AI and/or AI-assisted technologies were used in preparation of this paper.

References

- [1] Protolabs. "Hubs | On-demand Manufacturing: Quotes in Seconds, Parts in Days." <https://www.hubs.com/> (accessed).
- [2] "Best-in-Class, CAD Integrated DFM Software." <https://dfmpro.com/about-dfmpro/> (accessed 2023).
- [3] J. Y. Park and B. Khoshnevis, "A real-time computer-aided process planning system as a support tool for economic product design," *Journal of Manufacturing Systems*, vol. 12, no. 2, pp. 181-193, 1993.
- [4] S. Subrahmanyam and M. Wozny, "An overview of automatic feature recognition techniques for computer-aided process planning," *Computers in industry*, vol. 26, no. 1, pp. 1-21, 1995.
- [5] A. R. Colligan, T. T. Robinson, D. C. Nolan, Y. Hua, and W. Cao, "Hierarchical CADNet: Learning from B-Reps for Machining Feature Recognition," *Computer-Aided Design*, vol. 147, p. 103226, 2022.
- [6] P. Wang, W.-A. Yang, and Y. You, "A hybrid learning framework for manufacturing feature recognition using graph neural networks," *Journal of Manufacturing Processes*, vol. 85, pp. 387-404, 2023.
- [7] X. Yao, D. Wang, T. Yu, C. Luan, and J. Fu, "A machining feature recognition approach based on hierarchical neural network for multi-feature point cloud models," *Journal of Intelligent Manufacturing*, pp. 1-12, 2022.
- [8] Z. Wang and D. Rosen, "Manufacturing process classification based on heat kernel signature and convolutional neural networks," *Journal of Intelligent Manufacturing*, pp. 1-23, 2022.
- [9] C. Zhao and S. N. Melkote, "Learning the Part Shape and Part Quality Generation Capabilities of Machining and Finishing Processes Using a Neural Network Model," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2022, vol. 86212: American Society of Mechanical Engineers, p. V002T02A010.
- [10] Z. Zhang, P. Jaiswal, and R. Rai, "FeatureNet: Machining feature recognition based on 3d convolution neural network," *Computer-Aided Design*, vol. 101, pp. 12-22, 2018.
- [11] C. Zhao and S. N. Melkote, "Learning the manufacturing capabilities of machining and finishing processes using a deep neural network model," *Journal of Intelligent Manufacturing*, pp. 1-21, 2023.
- [12] Z. Wang and D. Rosen, "Manufacturing Process Classification Based on Distance Rotationally Invariant Convolutions," *Journal of Computing and Information Science in Engineering*, pp. 1-16, 2023.
- [13] F. Ning, Y. Shi, M. Cai, and W. Xu, "Part machining feature recognition based on a deep learning method," *Journal of Intelligent Manufacturing*, pp. 1-13, 2021.
- [14] Y. Zhang, Y. Zhang, K. He, D. Li, X. Xu, and Y. Gong, "Intelligent feature recognition for STEP-NC-compliant manufacturing based on artificial bee colony algorithm and back propagation neural network," *Journal of Manufacturing Systems*, vol. 62, pp. 792-799, 2022.
- [15] W. Liu et al., "Ssd: Single shot multibox detector," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016: Springer, pp. 21-37.
- [16] H. M. Ahmad and A. Rahimi, "Deep learning methods for object detection in smart manufacturing: A survey," *Journal of Manufacturing Systems*, vol. 64, pp. 181-196, 2022.
- [17] P. Wang et al., "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*, 2018: Ieee, pp. 1451-1460.
- [18] J. Cho, S. Kang, and K. Kim, "Real-time precise object segmentation using a pixel-wise coarse-fine method with deep learning for automated manufacturing," *Journal of Manufacturing Systems*, vol. 62, pp. 114-123, 2022.
- [19] P. Shi, Q. Qi, Y. Qin, P. J. Scott, and X. Jiang, "Intersecting machining feature localization and recognition via single shot multibox detector," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3292-3302, 2020.
- [20] H. Zhang, S. Zhang, Y. Zhang, J. Liang, and Z. Wang, "Machining feature recognition based on a novel multi-task deep learning network," *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102369, 2022.
- [21] Y.-J. Tseng and S. B. Joshi, "Recognizing multiple interpretations of interacting machining features," *Computer-Aided Design*, vol. 26, no. 9, pp. 667-688, 1994.
- [22] D. N. Sormaz and B. Khoshnevis, "Modeling of manufacturing feature interactions for automated process planning," *Journal of Manufacturing Systems*, vol. 19, no. 1, pp. 28-45, 2000.
- [23] R. Bidarra, K. J. De Kraker, and W. F. Bronsvort, "Representation and management of feature information in a cellular model," *Computer-Aided Design*, vol. 30, no. 4, pp. 301-313, 1998.
- [24] Y. S. Kim, "Recognition of form features using convex decomposition," *Computer-Aided Design*, vol. 24, no. 9, pp. 461-476, 1992.
- [25] Y. Woo and H. Sakurai, "Recognition of maximal features by volume decomposition," *Computer-Aided Design*, vol. 34, no. 3, pp. 195-207, 2002.
- [26] C. Mascle, H. Lu, and R. Maranzana, "Machining process planning using decomposition of delta volume," in *2007 IEEE International Symposium on Assembly and Manufacturing*, 2007: IEEE, pp. 106-111.
- [27] H. Nagaraj and B. Gurumoorthy, "Automatic extraction of machining primitives with respect to preformed stock for process planning," *Journal of manufacturing systems*, vol. 20, no. 3, pp. 210-222, 2001.
- [28] M. Behandish, S. Nelaturi, and J. de Kleer, "Automated process planning for hybrid manufacturing," *Computer-Aided Design*, vol. 102, pp. 115-127, 2018.

- [29] S. Nelaturi, G. Burton, C. Fritz, and T. Kurtoglu, "Automatic spatial planning for machining operations," in *2015 IEEE international conference on automation science and engineering (CASE)*, 2015: IEEE, pp. 677-682.
- [30] T. A. M. Tolio, L. Monostori, J. Váncza, and O. Sauer, "Platform-based manufacturing," *CIRP Annals*, vol. 72, no. 2, pp. 697-723, 2023.
- [31] X. Yan and S. Melkote, "Generative Modeling of the Shape Transformation Capability of Machining Processes," *Manufacturing Letters*, vol. 33, pp. 794-801, 2022.
- [32] X. Yan and S. Melkote, "Automated manufacturability analysis and machining process selection using deep generative model and Siamese neural networks," *Journal of Manufacturing Systems*, vol. 67, pp. 57-67, 2023.
- [33] D. Peddireddy *et al.*, "Identifying manufacturability and machining processes using deep 3D convolutional networks," *Journal of Manufacturing Processes*, vol. 64, pp. 1336-1348, 2021.
- [34] *binvox*. (2004-2021). Accessed: 2021-10-28. [Online]. Available: <http://www.patrickmin.com/binvox>
- [35] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026-8037, 2019.
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.
- [38] L. G. Shapiro, "Connected component labeling and adjacency graph construction," in *Machine intelligence and pattern recognition*, vol. 19: Elsevier, 1996, pp. 1-30.
- [39] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions On Graphics (TOG)*, vol. 36, no. 4, pp. 1-11, 2017.
- [40] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3577-3586.
- [41] Hub for Go Kart [Online] Available: <https://grabcad.com/library/hub-for-go-kart-1>
- [42] Model Shaft [Online] Available: <https://grabcad.com/library/model-shaft-1>
- [43] Cam Shaft [Online] Available: <https://grabcad.com/library/cam-shaft-87>