ARL-TR-10193 ● SEP 2025

DEVCOM ARMY RESEARCH LABORATORY

3D Carbon Fiber Tow Segmentation Using ML Methods

by Jennifer Sietins, Xiongjun Wu, and Andrew T. Gaynor

NOTICES

Disclaimers

ARL-TR-10193 ● Sep 2025

3D Carbon Fiber Tow Segmentation Using ML Methods

Jennifer Sietins, Xiongjun Wu, and Andrew T. Gaynor
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	3. DATES COVERED END DATE
September 2025	Technical Report	June 1, 2022	July 1, 2025

4. TITLE AND SUBTITLE
3D Carbon Fiber Tow Segmentation Using ML Methods

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Jennifer Sietins, Xiongjun Wu, and Andrew T. Gaynor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-MB Aberdeen Proving Ground, MD 21005	ARL-TR-10193

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCIDs: Jennifer Sietins, 0000-0003-2547-1653; Andrew T. Gaynor, 0000-0002-4384-9165

14. ABSTRACT
X-ray CT image segmentation of carbon fiber (CF) tow orientations is challenging due to the sample consisting of the same material phases having the same gray-scale contrast, so traditional thresholding methods are not successful. Advancements in ML and deep learning are overcoming this challenge by utilizing feature size, shape, and orientation information in segmentation algorithms. Two segmentation software programs (DragonFly and TomoSAM) were used to segment a 3D orthogonal CF preform. The specific procedures and methods to overcome segmentation challenges are described. The segmented datasets were then incorporated into Porous Microstructure Analysis (PuMA) software for quantification of the effective thermal conductivity in various directions. Sensitivity analysis was conducted by reassigning overlapping voxels from the DragonFly segmentation, which found only minor differences in the thermal conductivity results. Results from DragonFly and TomoSAM segmentations showed similar results and were in general agreement with one another. Both segmentation methods and subsequent PuMA simulation demonstrate the feasibility for quantifying thermal properties on real microstructures with ML overcoming prior fiber orientation segmentation challenges.

15. SUBJECT TERMS
machine learning, ML, image segmentation, computed tomography, CT, finite element analysis, homogenization, thermal properties, carbon/carbon composites, Sciences and Extreme Materials

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	34

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jennifer Sietins	(520) 691-6405

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

3D-woven carbon fiber (CF) and carbon matrix composites are used in high-performing applications such as the car-racing industry, aerospace structural components, and defense. Composite weaves typically undergo a densification process with resin infusion or chemical vapor infiltration, and these densification steps are often repeated numerous times to minimize porosity within the part. Shear deformation can cause significant distortions to the weave design during preforming and curing, especially dependent on the viscoelasticity of the resin and starting geometry of the weave pattern.[1] It is well known that the behavior of composites is highly dependent on the weave architecture, so an accurate geometric model is essential for correct thermomechanical predictions.

Several methods have been explored to generate models; however, the methods typically fall into the categories of meshes generated from microstructure statistics, extracted geometries from numerical simulations, and X-ray computed tomography (XCT) data.[2] Determining statistically significant microstructure input parameters, such as tow major and minor axes, undulation, and morphology can be time-consuming and still not accurately capture the geometry. Deriving the geometry from numerical simulations is not a predictive technique and requires a lot of assumptions regarding the model parameters. XCT accurately captures the tow geometries in 3D, but the segmentation and model generation can be difficult due to the tows having the same gray-scale contrast since they are made from the same material.

Different analysis techniques can overcome the challenge with 3D tow segmentation from XCT data. One method involves manual segmentation to reconstruct geometric models, which is tedious and labor-intensive process.[3] Another method uses results from fiber orientation analysis and the structure tensors to segment tows based on their respective orientation.[4] The disadvantage of this method is that the segmentation results are dependent on the accuracy of the orientation analysis and a higher scan resolution is often needed for accurate orientation quantification.[5] ML and deep learning (DL) segmentation techniques are more recent methods that have shown great promise. ML generally uses very sparse input data for training basic segmentation models and DL requires more accurate input labeling with improved segmentation results. While it takes more time to manually label input data for DL, the results of the neural network model provide faster, superior results compared with conventional ML and DL models.[6]

DL training algorithms incorporate edge and shape information within the segmentation algorithms rather than traditional image-processing methods that strictly use gray-scale thresholding.[7,8] This helps to automatically identify features based also on their size or orientation. It has also been found that DL can reduce scan times by as much as 80% and still preserve the segmentation accuracy since images with increased levels of noise are not as problematic compared with gray-scale thresholding segmentation methods.[9] With the success of DL segmentation to label tows of different orientations as different regions of interest, finite element (FE) meshes can readily be generated and exported for thermomechanical simulations.

Recently, Meta AI introduced the Segment Anything Model (SAM),[10,11] a foundation model for image segmentation designed to identify and segment any object in any image with minimal user input. SAM exhibits zero-shot capabilities, enabling it to segment objects it has never encountered during training without requiring additional fine tuning. These capabilities have rapidly found applications in domains such as XCT scan segmentation, where 3D segmentation is essential but often labor intensive.

An example of this is TomoSAM,[12] an open-source extension of 3D Slicer that integrates the SAM to assist in the segmentation of 3D volumetric data, particularly from tomographic imaging. It enables semiautomated segmentation by using SAM's 2D slice-based predictions. However, the process still requires manual user input to verify and annotate segmentation masks at selected slice intervals. To use TomoSAM, the user must first generate a multipage TIFF image stack from the CT scans. Next, image embeddings are created along the X, Y, and Z directions using SAM's image encoder. The user then selects representative slices at regular intervals and performs interactive segmentation by placing manual prompts (including and excluding points) on each slice. These verified masks are subsequently used by 3D Slicer to interpolate segmentations across the remaining slices, significantly improving efficiency while still requiring human oversight.

To predict the thermal material response (MR) of the C/C weave, we must use state-of-the-art 3D material response codes such as CHarring Ablator Response (CHAR),[13] Icarus,[14] Kentucky Aerothermodynamics and Thermal-response System (KATS),[15] and Aria.[16] The MR models require the definition of the bulk, or effective, thermal conductivity tensor for any material simulated. The calculation of this effective, and often anisotropic, thermal conductivity tensor is complicated, as there are sources of anisotropy at various scales of interest. For example, at the "microscale," or individual fiber scale, CFs are known to have much higher thermal conductivity on-axis versus off-axis. This report considers the so-called "mesoscale" where material architecture is resolved at a fiber tow scale. Thus, fiber tows, or bundles of fibers, are resolved by the simulation mesh while individual fibers are not. The mesoscale also resolves such features as matrix-rich regions and void regions.

To calculate effective macroscale, or bulk, properties, one may employ a traditional rule of mixtures to obtain the upper and lower bounds for a property of interest. However, the properties may vary wildly, depending on the complex architecture of the material. Thus, we leverage a code capable of accepting material architectures characterized through CT. NASA's Porous Microstructure Analysis (PuMA) is an open-source software package capable of calculating effective material properties from these defined architectures.[17,18] In 2021, Semeraro et al.[19] published details on the calculation of effective thermal conductivity values from the defined architecture within PuMA.

This paper details two segmentation methods for mesh generation of warp, weft, and Z-tows of an orthogonal CF weave from XCT data. A DL method and a zero-shot approach based on SAM were used to perform FE modeling (FEM) simulations in the procedures summarized in Figure 1. A plain orthogonal weave was selected as the most simplistic geometry, and future work aims to segment more complex fiber architectures and fully densified C/C.

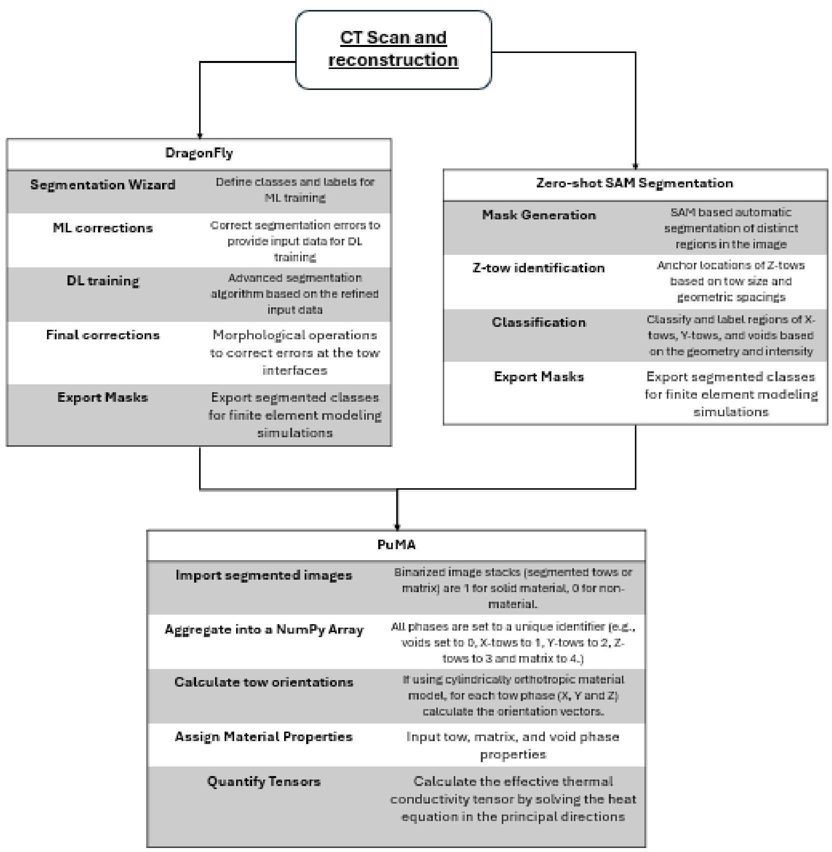

Figure 1. **Flowchart for segmentation methods and modeling analysis.**

2. Segmentation with DragonFly

2.1 Data Acquisition

A Zeiss Xradia 520 XCT scanner was used to scan a 3D orthogonal C/C weave. The sample was a 0.5-inch-diameter cylinder core cut via water-jet from a 12- × 12- × 0.5-inch panel. Scan settings included a voltage of 60 keV, 1601 projections, and a 15.74-µm voxel size. The 2D projections were reconstructed into a 3D volume using Zeiss's XMReconstructor version 16.2.18058 software and all subsequent analysis was conducted in DragonFly 2024.1.

2.2 Defining a Volume of Interest

The 3D dataset was loaded into DragonFly and a volume of interest defined. First, the translate/rotate tool was used to align the CF tows both vertically and horizontally. The dataset was cropped to a smaller volume of interest to eliminate edge effects due to the cutting of the sample, resulting in frayed CF bundles or, in some cases, missing CF bundles. The through-thickness direction was also cropped to eliminate difficulties in segmenting Z-tows that curve and form U-shapes at the top and bottom of the panel. The cropping was achieved by drawing an 8- × 8- × 10-mm box from the shapes menu, right-clicking on the box, and selecting "extract structured grid." Figure 2 illustrates the extracted volume (A) and the region of interest (ROI) box overlayed on the full sample in the various cutting plane orientations (B–D).

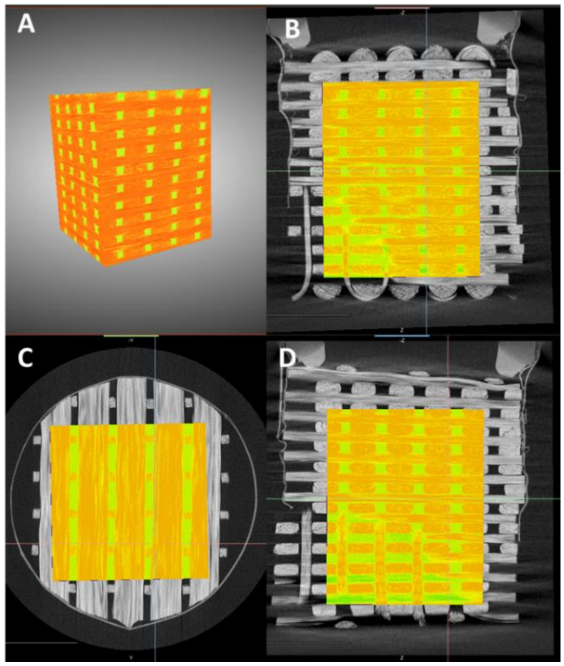

Figure 2. **8- × 8- × 10-mm volume of interest cropped to eliminate edge effects. A) 3D volume rendering. B–D) Overlay of the cropping box on the 2D views at three different viewing planes.**

2.3 Segmentation Wizard

The Segmentation Wizard is a tool in DragonFly that only requires sparse data inputs for training a model. The first step is to select specific frames of interest. In the case of a 3D orthogonal weave, it was of interest to select tows oriented both horizontally and vertically. Any number of classes can be defined and labeled with different colors. The ROI paint tools were used to very quickly color the images with the corresponding classes. The brush size was easily increased or decreased with CTRL + scrolling the mouse wheel up or down to obtain a brush size slightly smaller than the feature of interest. Figure 3 demonstrates the raw gray-scale images, defined classes and labels, and input training data for the Segmentation Wizard.

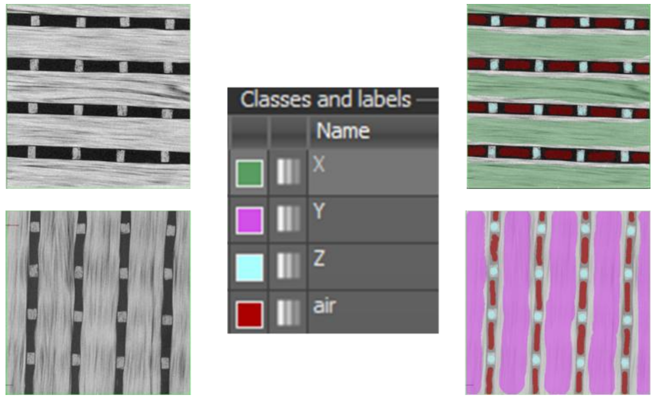

Figure 3. Input for the ML segmentation Wizard. (Left) Raw gray-scale images with horizontal and vertical tows. (Right) Sparsely painted classes and labels used as the input data for training the first model.

Predictions on the full image are displayed once the initial training is completed, as shown in Figure 4. In this case, a random forest model had the best accuracy and was saved for subsequent use. There are clearly some errors, but overall the model did a nice job assigning the correct class to the gray-scale regions that were not part of the input data.

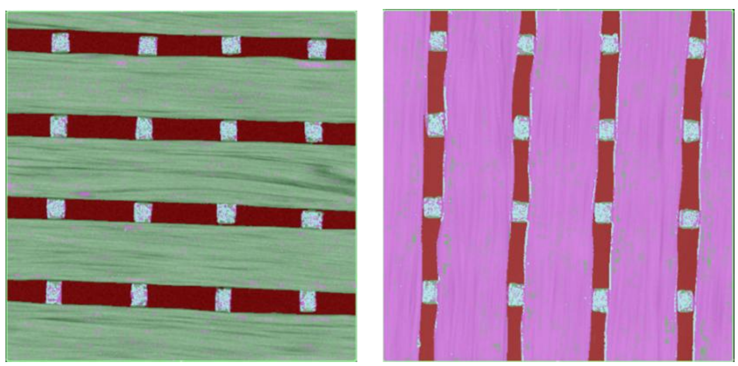

Figure 4. Segmentation Wizard model prediction on the full image.

It is recommended to have at least three well-segmented images as input data for DL. Four frames were selected as input data to have at least two images with tows oriented horizontally and two vertically. Slices were marked by checking the box in the lower left-hand corner and a new dataset was created by right-clicking on the full dataset and selecting the "new image from marked slices" option. This generated a new dataset entitled "extracted marked slices." The Segmentation Wizard model was applied to the "extracted marked slices" within the "segment with AI" menu. The results of this segmentation are shown in Figure 5, which illustrates the four raw gray-scale images on the top row and the same four images with the Segmentation Wizard model applied on the bottom row.

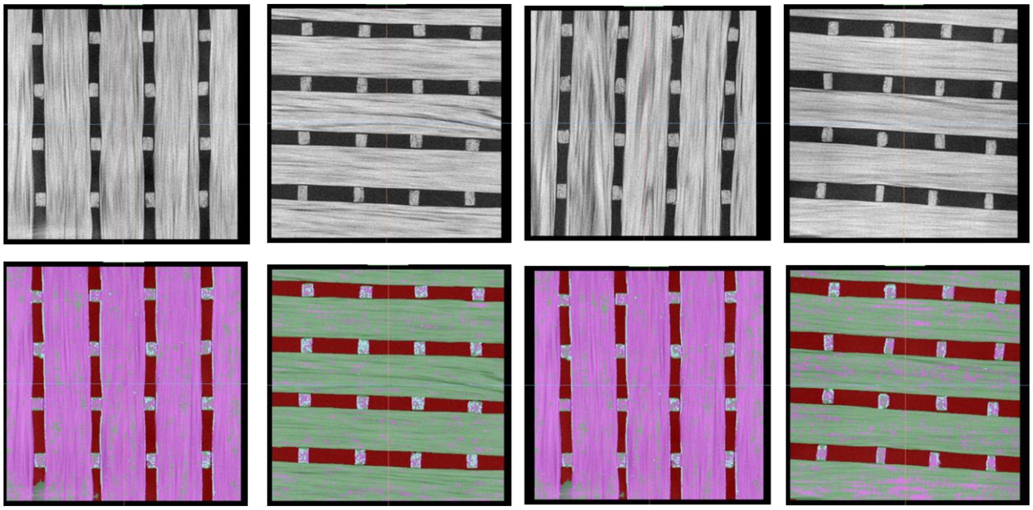

Figure 5. Raw gray-scale images of marked slices (top) and with the Segmentation Wizard model applied (bottom).

To have four well-segmented images as the input data for DL, it is important to manually correct ML-predicted segmentation errors. While this is a much more time-consuming task, it is important to have accurate input data for the DL training to improve the model accuracy that will later be applied to the full 3D volume. There are painting tools within the DragonFly software to make the manual painting process easier and less tedious. For example, a "local otsu" painting will only paint the brighter- or darker-colored pixels. This tool is particularly helpful with interfaces of significantly different gray-scale contrasts. Another beneficial feature is a method to lock particular classes and only modify selected classes of interest. This can be accomplished by holding down CTRL and clicking only on the class that should remain the same and the class that should be the corrected color. Figure 6 shows an example of this technique for a manual correction of the erroneous Z-tow class (cyan) that should be labeled as the Y-tow class (magenta). By only selecting the air (dark red) and Y (magenta) classes, the air phase (dark red) will be locked for modification when the brush tool is on the magenta color. The erroneous boundary can quickly be painted and corrected to change the Z (cyan) to Y (magenta) while leaving the air class (dark red) the same.

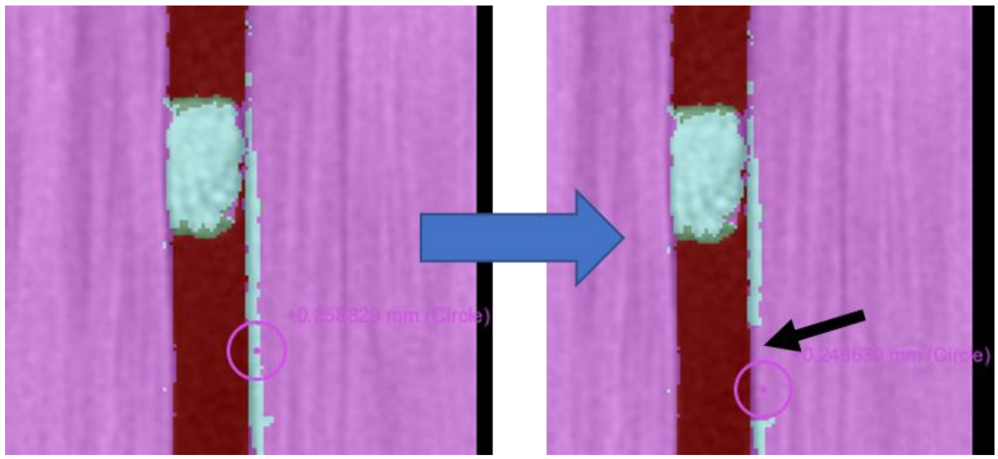

Figure 6. Example of manual correction of erroneous Z (cyan) class to Y (magenta) class while locking the air class (dark red) from modification with the brush tool. The black arrow shows a region that has been manually corrected with the brush tool, it is clear the air phase remained unchanged.

Figure 7 illustrates the four slices after manual correction. These images serve as input data for training the DL model.

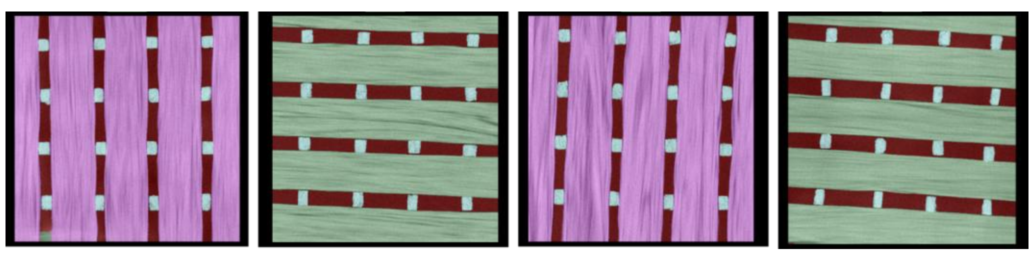

Figure 7. Manually corrected, well-segmented slices as input for the DL training.

2.4 DL

The input data for the DL training is the "extract marked slices" that are the raw gray-scale image, and the output is the dataset of the manually-corrected segmentation from the Segmentation Wizard model prediction. The training settings were the default values except the flip horizontally, flip vertically, rotate, and shear boxes were unchecked since the classes are geometry-dependent rather than strictly contrast- or size-dependent.

The model was saved and DL window closed upon the completion of the training. Figure 8 shows previews of the DL model applied to two different single-image slices, neither of which is being served as DL input data. The results showed a significant improvement of the correct class identification compared with the Segmentation Wizard model.

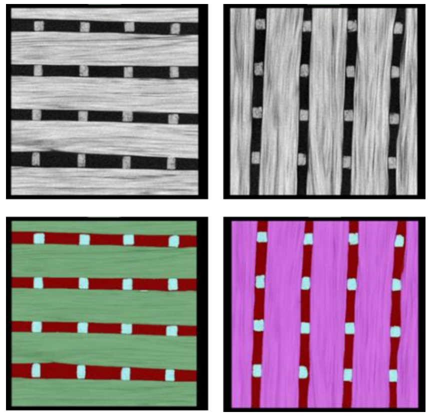

Figure 8. Raw gray-scale images of unmarked slices with horizontal and vertical oriented tows (top) and DL model applied (bottom) demonstrate minimal errors in class identification.

Applying the model to the full 3D volume is accomplished within the "Segment with AI" menu. Select the DL model and then "segment all slices" to generate a new multi-ROI. Some errors became apparent when examining the results on the 3D volume, particularly with the Z (cyan) class located near the interfaces of the different tow orientations between X and Y directions. This is evident in Figure 9, which shows erroneous cyan colors above and below the air (dark red) class. There are two main techniques that could be used to correct these errors: manually correcting the errors and retraining an improved model with additional data or correction through image-processing techniques such as erosion/dilation functions or despeckling.

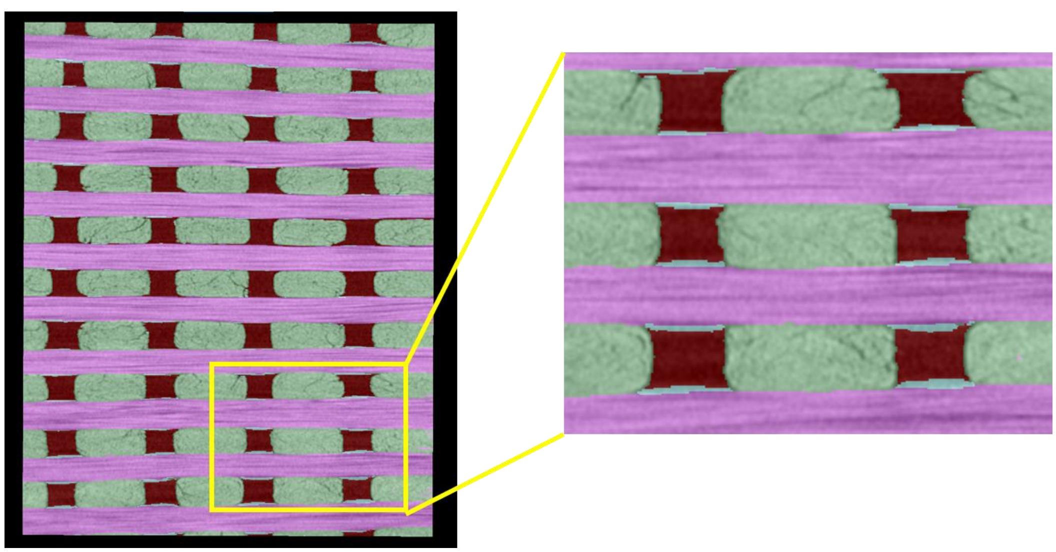

Figure 9. DL model results included errors with the Z (cyan) class segmentation at the interfaces of the X- and Y-tow orientations.

Because manual correction and retraining would take more time and still not guarantee to fix these errors with a new model, image-processing techniques were employed. First, the classes were separated into four distinct regions of interests by right-clicking on the segmented multi-ROI and selecting “extract ROIs.” This generated one ROI for each class.

The opening function was used, which is a successive erosion and dilation procedure of the same magnitude. The erosion function serves to eliminate all the small, erroneous Z class (cyan) at the interfaces, and then the dilation function serves to return the remainder of the Z class back to the original size. Figure 10 clearly shows the effectiveness of this image-processing morphological operation through the elimination of the model errors.

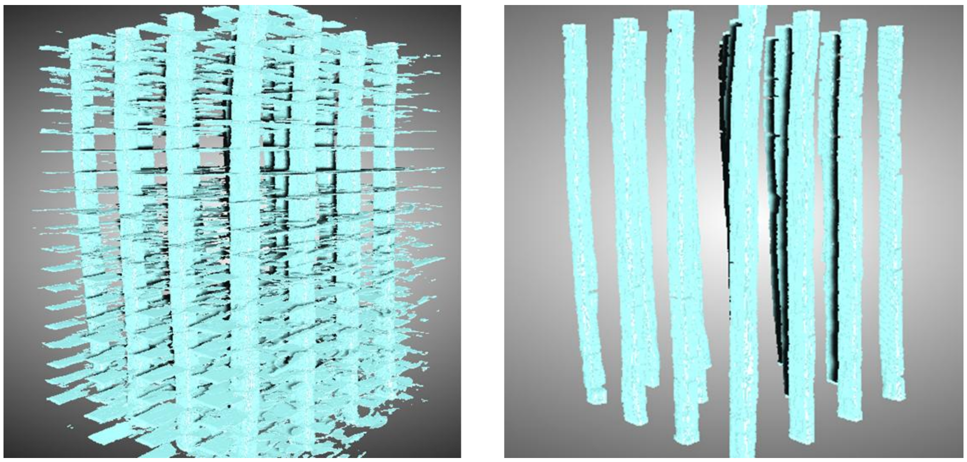

Figure 10. Z class before (left) and after (right) application of the opening morphological operation function to eliminate errors at the X–Y tow interfaces.

The final segmented volume is shown in Figure 11 after the manual correction of the ROIs through traditional image-processing techniques. Surface meshes were generated for each class simply by right-clicking on each class ROI and creating an .stl file format mesh. These models were exported for subsequent FE analysis.

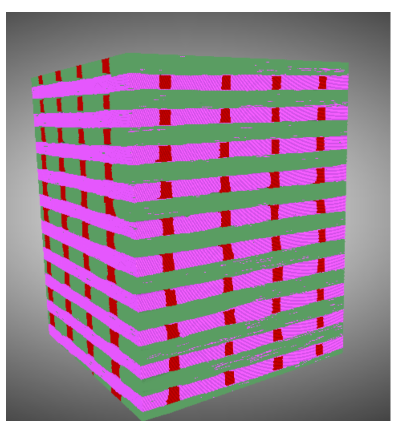

Figure 11. Segmented 3D volume rendering of X, Y, Z, and air classes of a 3D orthogonal CF weave.

3. Zero-shot Segmentation of CT Data with SAM

Another zero-shot approach developed for segmenting the orthogonal CF weave is based on the SAM. It uses a vision transformer (ViT) to encode the image into a set of visual features, followed by a transformer decoder that aggregates the features to generate segmentation masks. It is trained to predict the segmentation mask from input images and user-provided prompts using a combination of supervised and self-supervised learning techniques, leveraging a massive dataset of approximately 11 million images and 1.1 billion segmentation masks to achieve good performance.[10,11]

SAM can operate in two modes:

1) Automatic mask generation mode: performs zero-shot segmentation by automatically segmenting objects or regions in an image.
2) Mask prediction mode: generates masks based on user-provided prompts such as points and bounding boxes.

To minimize the user intervention and maximize the automation in the zero-shot approach, an automated tow segmentation algorithm was developed based on SAM's automatic mask generation. The objective was to ensure that all the tows were adequately covered by the segmentation masks generated by SAM and the automatically segmentation masks were accurately classified into the correct categories. The segmentation procedure consisted of the following steps:

1) Perform SAM automatic mask generation using SAM to segment distinct regions in the image and generate the segmentation masks.
2) Identify anchoring locations of Z-tows based on tow size and geometric spacing and partition the CT scan into characteristic regions using Z-tow map generation based on Z-tow locations.
3) Classify regions into X-tows, Y-tows, and voids based on their geometric relationship with respect to the Z-tows and average pixel intensity. The corresponding masks are then identified and labeled.
4) Export the segmented masks for Z-tow, Y-tow, and X-tow as binary images for downstream applications (e.g., FEM simulations).

The following sections detail each step of the algorithm.

3.1 Automatic Mask Generation Using SAM

The segmentation process began with automatic mask generation using the SAM. This step aimed to ensure all relevant structural features—X-tows, Y-tows, Z-tows, and voids—were captured in the SAM-generated segmentation masks. To achieve comprehensive coverage, the input hyperparameters to SAM's automatic mask generator were carefully tuned.

The key hyperparameters included the following:

- points_per_side: controls the sampling density of points across the image. A higher value increases the number of candidate regions for segmentation.
    - Example: points_per_side = 16 provides a dense grid of 256 points (16 × 16), improving the likelihood of capturing fine tow structures.
- pred_iou_thresh: sets the threshold for predicted intersection-over-union (IoU) confidence. Only masks with a predicted IoU above this value are retained.
    - Example: pred_iou_thresh = 0.8 ensures high-confidence masks, reducing false positives but potentially missing low-contrast features.
- min_mask_region_area: filters out small regions below a specified pixel area. This helps eliminate noise or irrelevant fragments.
    - Example: min_mask_region_area = 60 removes very small masks that are unlikely to correspond to actual tows.

These parameters were adjusted iteratively until the generated masks adequately covered all tows of interest. Figure 12 illustrates how the input hyperparameters provided to SAM's automatic mask generator affected the segmentation results of the XCT scan.

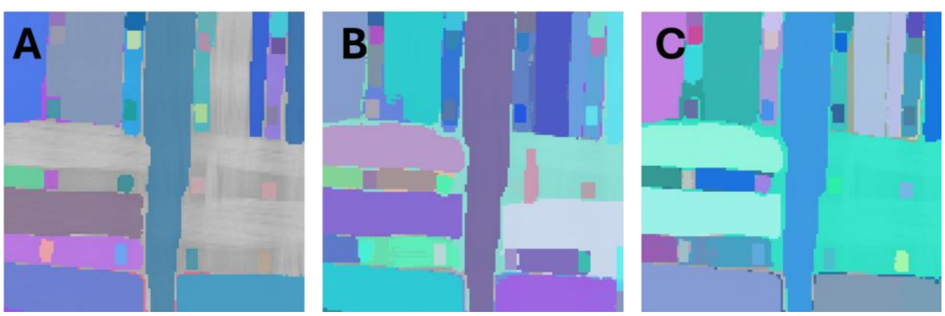

Figure 12. SAM-generated segmentation masks with different input parameters: (A) points_per_side = 32, pred_iou_thresh = 0.88, min_mask_region_area = 0; (B) points_per_side = 32, pred_iou_thresh = 0.8, min_mask_region_area = 60; (C) points_per_side = 16, pred_iou_thresh = 0.8, min_mask_region_area = 60.

3.2 Z-tow Identifications

Z-tows are typically well-spaced and geometrically distinct, making them more straightforward to identify compared with X- and Y-tows. The identification process is performed iteratively using a set of geometric and intensity-based criteria to ensure accurate classification of SAM-generated masks. The procedure involves the steps explained in the four subsections that follow.

3.2.1 IoU-based Filtering

Each segmentation mask generated by SAM is compared with a user-defined reference window representing the average Z-tow size (e.g., 7- × 8-pixel rectangle in the example case). The IoU between each mask and the reference window is computed and sorted in descending order. This helps prioritize masks that closely match the expected Z-tow shape.

3.2.2 Intensity Thresholding

To distinguish tows from the surrounding matrix or voids, the mean pixel intensity of each mask is evaluated. Only masks with intensities above a threshold, defined as the global mean image intensity adjusted by a user-specified factor, are retained. This helps eliminate low-density regions that are unlikely to correspond to CF tows.

3.2.3 Area Constraint

The area of each mask is checked to ensure it falls within a reasonable range around the expected Z-tow size (e.g., approximately 56 pixels^2 for a 7- × 8-pixel region). This filters out masks that are too small (noise) or too large (merged regions).

3.2.4 Tow Count Consistency

The total number of identified Z-tows is compared against the expected number of Z-tows in the scan (e.g., 16 in the sample case). This acts as a final validation step to ensure completeness and consistency.

To satisfy the above criteria, the algorithm automatically adjusts the pred_iou_thresh parameter in SAM's mask generator described previously. It begins with a high threshold (favoring high-confidence masks) and gradually lowers it until the desired number and quality of Z-tow masks are obtained. Figure 13 shows an example of successfully identified Z-tow masks, demonstrating the effectiveness of this iterative filtering and refinement process.

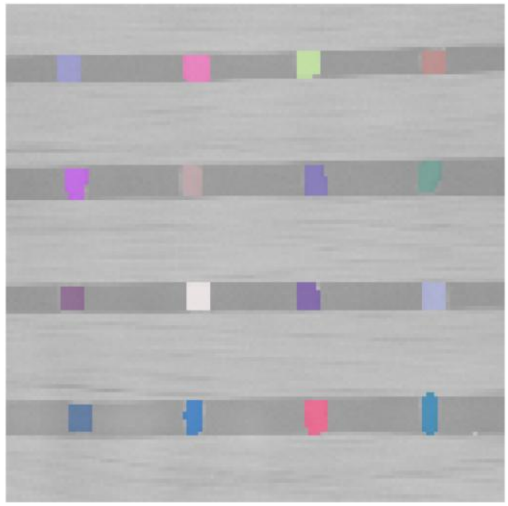

Figure 13. Sample Z-tows identified from the SAM masks based on the automated iterative Z-tow identification scheme.

3.3 Z-Tow–based XCT Scan Partition and Tow Characterization

With the Z-tow masks successfully identified, the next step is to partition the XCT scan and characterize the remaining regions as X-tows, Y-tows, or non-tow areas. This process leverages the spatial arrangement of Z-tows as anchors to guide the classification of surrounding regions. The procedure is described in the following four subsections.

3.3.1 XCT Scan Partitioning

The scan is divided into smaller rectangular regions using the positions of the Z-tows as anchors. These partitions are defined by the horizontal and vertical spacing between neighboring Z-tows (Figure 14). This spatial grid forms the basis for localized tow classification.

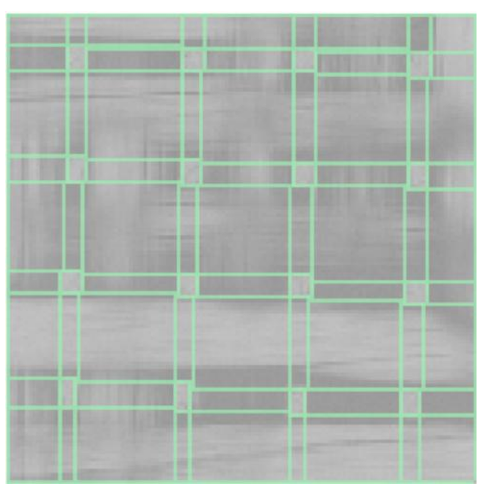

Figure 14. Sample scan partition based on identified Z-tows.

3.3.2 Non-Tow Region Identification

Each partition is analyzed based on its average pixel intensity. Regions with significantly lower intensity—typically corresponding to voids or resin-rich areas—are classified as non-tow regions and labeled as "N" in Figure 15.

T	X	T	X	T	X	T	X	T
Y	Z	Y	Z	Y	Z	Y	Z	Y
T	X	T	X	T	X	T	X	T
Y	Z	Y	Z	Y	Z	Y	Z	Y
T	X	T	X	T	X	T	X	T
Y	Z	Y	Z	N	Z	N	Z	Y
T	X	T	X	T	X	T	X	T
Y	Z	Y	Z	N	Z	N	Z	N
T	X	T	X	T	X	T	X	T

Figure 15. Sample identified tow characteristics of direct neighbors of Z-tows. Regions marked "T" are corner neighbors to be identified.

3.3.3 Tow Type Assignment (X/Y)

Regions located between two horizontally adjacent Z-tows are classified as Y-tows, while those between vertically adjacent Z-tows are classified as X-tows. These are labeled as "Y" and "X," respectively, in Figure 15. Regions that lie at the corners between diagonally or non-neighboring Z-tows are initially marked as "T" (transitional or ambiguous regions).

3.3.4 Tow Type Resolution in Transitional Regions

For each "T" region, the algorithm examines the adjacent X- and Y-tow masks to determine which tow direction crosses through the region. Based on this analysis, the "T" region is reclassified as either an X- or Y-tow, depending on the dominant orientation. This step ensures complete and consistent tow labeling across the scan. The final result is a fully characterized tow map with each region labeled as X-, Y-, Z-, or non-tow (Figure 16).

X	X	X	X	X	X	Y	X	Y
Y	Z	Y	Z	Y	Z	Y	Z	Y
Y	X	X	X	X	X	X	X	Y
Y	Z	Y	Z	Y	Z	Y	Z	Y
Y	X	Y	X	X	X	X	X	Y
Y	Z	Y	Z	N	Z	N	Z	Y
X	X	X	X	X	X	X	X	X
Y	Z	Y	Z	N	Z	N	Z	N
X	X	X	X	X	X	X	X	X

Figure 16. Sample fully identified tow characteristics based on Z-tows.

3.4 Generation of Final Tow Mask Images

With all regions in the XCT scan now characterized as X-, Y-, Z-, or non-tows, the final step involved organizing and consolidating the segmentation masks for downstream modeling applications. Each segmentation mask generated by SAM was assigned to one of the three tow categories—X-, Y-, or Z-tow—based on the region in which it resides. Masks belonging to the same tow category were merged to form unified binary masks for each tow type. This resulted in three comprehensive masks: one each for X-, Y-, and Z-tows. These final masks were then exported as labeled images or volumes, which serve as the basis for constructing a 3D grid suitable for FEM or other structural simulations.

Figure 17 shows the original scan with the final merged masks for each tow type in the selected scan, while Figure 18 presents results from a different scan. These segmentation masks of the entire scan volume—representing X-, Y-, and Z-tows—are then used to construct the 3D grid used for the FEM.

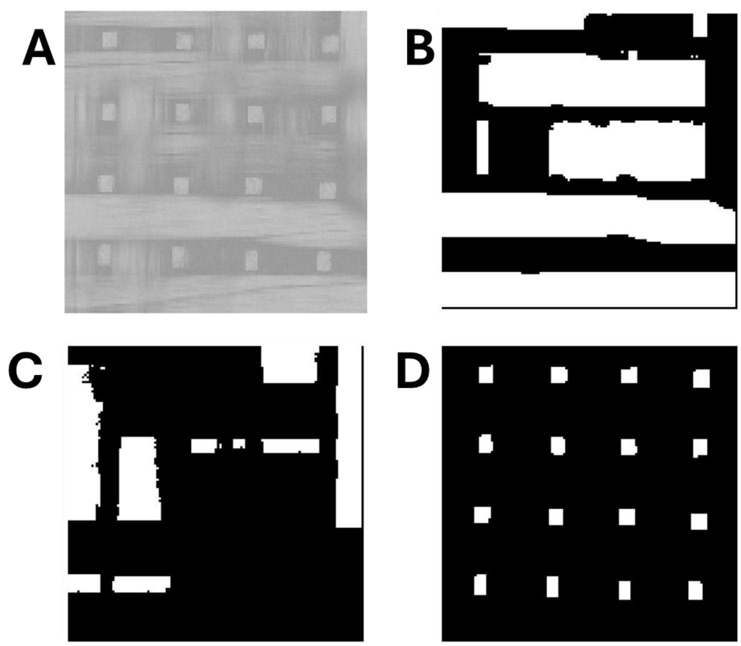

Figure 17. Segmentations of a selected scan: A) sample image, B) X-tow mask, C) Y-tow mask, and D) Z-tow mask.

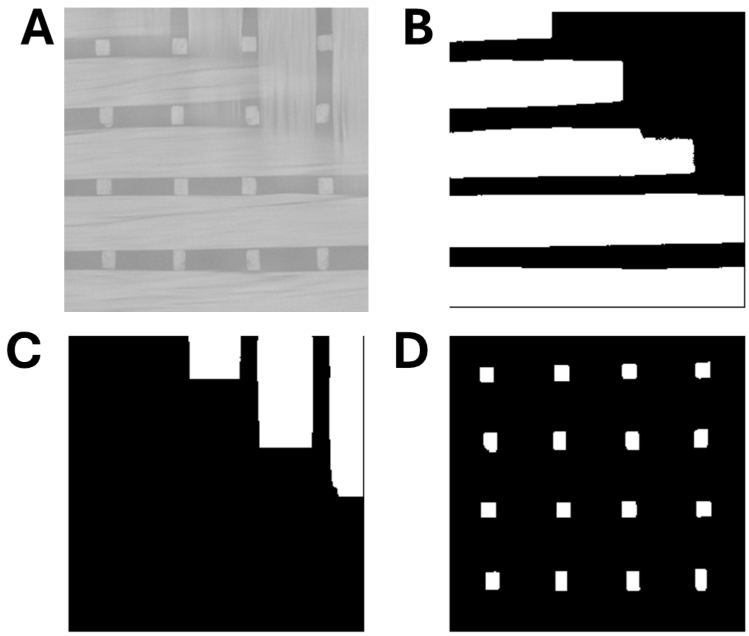

Figure 18. Segmentations of another selected scan: A) sample image, B) X-tow mask, C) Y-tow mask, and D) Z-tow mask.

4. Effective Property Prediction from Segmented Material Architectures

This section outlines an approach to predict the effective thermal conductivity tensor given the XCT scanned and segmented material architecture. NASA's PuMA code is used for these calculations.[19] We compare predicted properties for both Dragonfly and SAM segmentations, and for different assumed material constitutive models (direction-dependent material properties on the voxel-by-voxel level).

PuMA can define a material as isotropic, transversely isotropic, orthotropic, cylindrically isotropic, and fully anisotropic. For comparison purposes, we will define the materials here as both orthotropic and cylindrically orthotropic. The materials are defined for any defined set of voxels in the domain. The orthotropic case assumed the following properties for a hypothetical AS4 CF.[21] The matrix material (here assumed to exist anywhere there was void in the noninfiltrated 3D orthogonal weave) is assumed to be made of the same CF material and the thermal conductivity is assumed to be isotropic and set to an average of the on-axis versus off-axis thermal conductivities:

$$\begin{gathered}\text{X-tows: } [k_x, k_y, k_z] = [6.9, 2.8, 2.8] \text{ W/(mK)} \\ \text{Y-tows: } [k_x, k_y, k_z] = [2.8, 6.9, 2.8] \text{ W/(mK)} \\ \text{Z-tows: } [k_x, k_y, k_z] = [2.8, 2.8, 6.9] \text{ W/(mK)} \\ k_{matrix} = 4.85 \text{ W/(mK) (isotropic)}\end{gathered} \tag{1}$$

The cylindrically orthotropic model specifies both on-axis and off-axis fiber tow thermal conductivity values. The orientation of individual constituents (e.g., fiber tows) will have a significant impact on the effective thermal properties. Thus, the developers of PuMA leverage the structure tensor method to calculate local orientation based on the imported XCT scan.[20] This material model requires the calculation of material orientation on the imported XCT scan. Here, we run the orientation calculation on the imported X-, Y-, and Z-tow phases separately and merge before assigning the orientation-dependent material properties to the entire material architecture. The material properties of AS4 for the cylindrically orthotropic model are defined as follows:

$$\begin{gathered} k_{axial} = 6.9 \text{ W/(mK)} \\ K_{off\_axis} = 2.8 \text{ W/(mK)} \\ k_{matrix} = 4.85 \text{ W/(mK) (isotropic)} \end{gathered} \tag{2}$$

The workflow to calculate effective thermal properties of the CF weave is defined as follows:

1) Import segmented (binarized) fiber tow TIFF image stacks as numpy array into PuMA. Binarized fiber tows will be 1 for solid material and 0 for nonmaterial.
2) If importing separate segmented fiber tows (X-, Y-, and Z-tow image stacks), aggregate into a single 3D numpy array. A visualization of an example combined array is seen in Figure 19, where void is set to 0, X-tows to 1.0, Y-tows to 2.0, and Z-tows to 3.0.

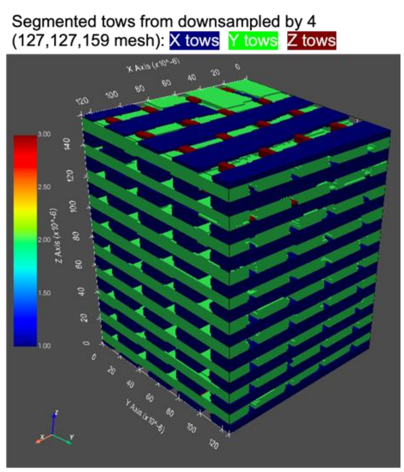

Figure 19. Visualization of the imported fiber tow orientations segmented with ML methods.

3) For the cylindrically orthotropic material model, calculate orientation of each fiber tow directions (e.g., all tows in X direction) individually.

4) Assign material properties to each tow, matrix material, and void phase. If using a cylindrically orthotropic material model for a phase, assign it after calculating the fiber orientation.
5) Obtain the effective, global thermal conductivity tensor by solving the heat equation in all three principal directions. For this, a temperature gradient is applied across the domain and the heat equation is solved iteratively to stead state. PuMA uses the finite volume method to solve the heat equation. This will ultimately yield the flux vector, shown here for the X direction:

$$q^X = [q^{Xx}, q^{Xy}, q^{Xz}]^T. \tag{3}$$

Using these flux values from the X direction simulation, we can calculate the first row of the effective thermal conductivity tensor by multiplying the flux by the domain lengths. This is represented by Equation 4:

$$\begin{aligned} k^{xx} &= q^{Xx} L_x \\ k^{xy} &= q^{Xy} L_x \\ k^{xz} &= q^{Xz} L_x \,. \end{aligned} \tag{4}$$

Calculating the steady state flux in the Y and Z direction and following the same procedure, we can fill out the effective thermal conductivity tensor, represented here:

$$\mathrm{K}^{\mathrm{eff}} = \begin{bmatrix} k^{xx} & k^{xy} & k^{xz} \\ k^{xy} & k^{yy} & k^{yz} \\ k^{xz} & k^{yz} & k^{zz} \end{bmatrix}. \tag{5}$$

The raw exported XCT scan had a resolution of (508, 508, 635) pixels. This produces a representation with nearly 164 million voxels, or mesh with 164 million finite volumes, which proved too computationally expensive, even for our DoD High-Performance Computing cluster with 128 GB of RAM (memory constrained). To reduce the computational requirements, the voxelized representation was downsampled by two- and fourfold in all directions to produce voxelized representations of (254, 254, 318) and (127, 127, 159), respectively. The computational requirements precluded the calculation of effective thermal conductivity for the full-resolution mesh, but calculations are compared for both downsampled meshes. The approach used for downsampling is averaging and combing every 2 × 2 voxels into one voxel for downsampling by two and averaging and combining every 4 × 4 voxels into one voxel for downsampling by four.

First, we perform the PuMA calculations on the Dragonfly-segmented data for both downsampled "resolutions." The homogenized thermal conductivity tensor, K^{eff}, was first calculated assuming the orthotropic model. Thus the X-, Y-, and Z-tows were set to have the orthotropic properties defined in Equation 1. For comparison,

K^{eff} was calculated assuming a cylindrically orthotropic material model as well, with tow-specific material properties defined in Equation 2 and results shown in Table 1.

Table 1. Effective thermal conductivity tensors for the Dragonfly-segmented architecture, assuming orthotropic and cylindrically orthotropic material models.

	Downsampled by 2	Downsampled by 4
Orthotropic	$\begin{bmatrix} 4.73 & 0.00 & 0.00 \\ 0.00 & 4.69 & 0.00 \\ 0.00 & 0.00 & 3.31 \end{bmatrix}$	$\begin{bmatrix} 4.72 & 0.00 & 0.00 \\ 0.00 & 4.68 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$
Cylindrically orthotropic	$\begin{bmatrix} 4.74 & 0.04 & 0.00 \\ 0.04 & 4.25 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$	$\begin{bmatrix} 4.75 & 0.02 & 0.01 \\ 0.02 & 4.34 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$

Table 2 shows the results of the PuMA calculations comparing the Dragonfly (downsampled by four) and SAM-segmented data. The orthotropic and cylindrically orthotropic material properties were defined the same as in the Dragonfly-segmented case above.

Table 2. Effective thermal conductivity tensor comparison for Dragonfly- and SAM-segmented architecture, assuming orthotropic and cylindrically orthotropic material models.

	Dragonfly	SAM
Orthotropic	$\begin{bmatrix} 4.72 & 0.00 & 0.00 \\ 0.00 & 4.68 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$	$\begin{bmatrix} 4.72 & 0.00 & 0.00 \\ 0.00 & 4.69 & 0.00 \\ 0.00 & 0.00 & 3.38 \end{bmatrix}$
Cylindrically Orthotropic	$\begin{bmatrix} 4.75 & 0.02 & 0.01 \\ 0.02 & 4.34 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$	$\begin{bmatrix} 4.71 & 0.01 & 0.00 \\ 0.01 & 4.47 & 0.00 \\ 0.00 & 0.00 & 3.38 \end{bmatrix}$

There are a few comparisons to make in the following categories:

- Downsampling amount: There is little change in predicted K^{eff} from the two resolution levels. This would indicate the downsampled by four resolution is sufficient for this material architecture and would save significant computational time.
- Orthotropic versus cylindrically orthotropic: There is minimal change in predicted K^{eff} between orthotropic and cylindrically orthotropic thermal conductivity. The largest discrepancy occurs in the k^{yy} term for the SAM-segmented case (4.69 vs. 4.47). This is not surprising at all, given that the X-, Y- and Z-tows are almost perfectly aligned along the principal directions. Misalignment from principal axes and tow curvature would produce significant discrepancies in predicted K^{eff} from the two material constitutive models.

- Dragonfly versus SAM: There are some differences in predicted K^{eff} for the orthotropic case, but especially for the cylindrically orthotropic case. In the orthotropic case, the k^{zz} term is roughly 2.4% greater. In the cylindrically orthotropic case, the k^{zz} term is identically different, but the k^{yy} term has a larger discrepancy of 3.0%. In any case, these discrepancies are still minimal.

The Dragonfly tow-level segmentation approach requires additional dilation and erosion operations to produce a more accurate final structure. In performing those operations, there becomes zones of overlap with the tows. Thus, after importing the three tow-segmented cases (X-, Y-, and Z-tows) and intersecting the meshes, there are regions where voxels are assigned to more than one tow. Looking at the downsampled by four case, this overlap is seen to be

Overlap X-tow/Y-tow = 1.41%

Overlap X/Z = 0.16%

Overlap Y/Z = 0.15%

In the calculations above, the voxelized X-, Y-, and Z-tows are imported into PuMA and any overlapping zones were assigned as air/void for the thermal conductivity calculations. To test the sensitivity of the predicted thermal conductivity tensor to these overlap zones, we perform a small study in which the significant X/Y tow overlap is instead assigned to the X- and Y-tows. The results of this study are seen in Table 3.

Table 3. Influence of overlapping voxels from the DragonFly segmented data.

	X/Y overlap assigned to air	**X/Y overlap assigned to X**	**X/Y overlap assigned to Y**
Percent of each phase	X-tows = 38.64% Y-tows = 38.06% Z-tows = 4.88% Air = 18.42%	X-tows = 40.05% Y-tows = 38.06% Z-tows = 4.88% Air = 17.01%	X-tows = 38.64% Y-tows = 39.47% Z-tows = 4.88% Air = 17.01%
Orthotropic	$\begin{bmatrix} 4.72 & 0.00 & 0.00 \\ 0.00 & 4.68 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$	$\begin{bmatrix} 4.75 & 0.00 & 0.00 \\ 0.00 & 4.65 & 0.00 \\ 0.00 & 0.00 & 3.28 \end{bmatrix}$	$\begin{bmatrix} 4.69 & 0.00 & 0.00 \\ 0.00 & 4.71 & 0.00 \\ 0.00 & 0.00 & 3.29 \end{bmatrix}$
Cylindrically orthotropic	$\begin{bmatrix} 4.75 & 0.02 & 0.01 \\ 0.02 & 4.34 & 0.00 \\ 0.00 & 0.00 & 3.30 \end{bmatrix}$	$\begin{bmatrix} 4.88 & 0.02 & 0.00 \\ 0.02 & 4.24 & 0.00 \\ 0.00 & 0.00 & 3.28 \end{bmatrix}$	$\begin{bmatrix} 4.63 & 0.02 & 0.00 \\ 0.02 & 4.49 & 0.00 \\ 0.00 & 0.00 & 3.28 \end{bmatrix}$

The X/Y tow overlap is only 1.41%, so the expected impact on K^{eff} is minimal. Contributing the overlap region to X and Y resulted in a minor bump in thermal conductivity in that principal direction. For example, in the orthotropic case contributing the overlap region to Y-tows, the k^{yy} thermal conductivity bumped up to 4.71 from 4.68, a 0.6% change. The change was more pronounced in the cylindrically orthotropic case—for the same k^{yy} prediction, the value went from 4.34 to 4.49, a 3.5% change. In any case, the differences were still small and can be attributed primarily to the small overlapping region.

5. Conclusions

ML and DL methods for image segmentation were successfully demonstrated on a CF orthogonal preform to generate surface meshes of X-, Y-, and Z-tows as well as an air phase. Automated image segmentation has potential to save significant analysis time and reduce human error or bias, and tow-level segmentation is critical for composite materials with strong directional dependencies based on the fiber architecture. This analysis framework also improves modeling and simulation accuracy through modeling real material microstructures versus assumed or idealized models.

The tow-level image segmentation was demonstrated using methods based in both DragonFly and SAM. Meshes from both programs were analyzed with PuMA and the effective thermal conductivity was quantified. The differences between the image segmentation methods were found to be minimal. Additionally, it was found that downsampling the data to have a more manageable file size and faster computational time had minimal effect. For the DragonFly segmentation, the influence of overlapping phase assignments was explored and assigned to different segmentation classes with minimal influence on the effective thermal conductivity values. All these results are promising to show the image segmentation is not so highly sensitive as to impact the outputs of the thermal MR model. Therefore, minor segmentation errors with either image-processing method should have minimal impact on the accuracy of the computational results for the thermal properties.

6. References

1. Feng Y et al. Numerical prediction for viscoelasticity of woven carbon fiber reinforced polymers (CFRPs) during curing accounting for variation of yarn angle caused by preforming. Compos A: Appl Sci Manuf. 2023;173:107631.

2. Sinchuk Y, Kibleur P, Aelterman J, Boone M, Van Paepegem W. Variational and deep learning segmentation of very-low-contrast X-ray computed tomography images of carbon/epoxy woven composites. Materials. 2020;13(4):936.

3. Huang W, Causse P, Brailovski V, Hu H. Trochu F. Reconstruction of mesostructural material twin models of engineering textiles based on micro-CT aided geometric modeling. Compos A: Appl Sci Manuf. 2019;124:105481.

4. Wijaya W, Kelly PA, Bickerton S. A novel methodology to construct periodic multi-layer 2D woven unit cells with random nesting configurations directly from μCT-scans. Compos Sci Technol. 2020;193:108125.

5. Straumit I, Lomov SV, Wevers M. Quantification of the internal structure and automatic generation of voxel models of textile composites from X-ray computed tomography data. Compos A: Appl Sci Manuf. 2015;69:150–158.

6. Ali MA, Guan Q, Umer R, Cantwell WJ, Zhang T. Deep learning based semantic segmentation of μCT images for creating digital material twins of fibrous reinforcements. Compos A: Appl Sci Manuf. 2020;139:106131. https://doi.org/10.1016/j.compositesa.2020.106131

7. Badran A et al. Automated segmentation of computed tomography images of fiber-reinforced composites by deep learning. J Mater Sci. 2020;55(34):16273–16289.

8. Lorenzoni R et al. Semantic segmentation of the micro-structure of strain-hardening cement-based composites (SHCC) by applying deep learning on micro-computed tomography scans. Cem Concr Compos. 2020;108:103551.

9. Yang H et al. Segmentation of computed tomography images and high-precision reconstruction of rubber composite structure based on deep learning. Compos Sci Technol. 2021;213:108875.

10. Segment anything. Meta; c2023 [accessed 2025 July 22]. https://segment-anything.com/

11. Kirillov A et al. Segment anything. arXiv; 2023 Apr 5. arXiv: 2304.02643. https://doi.org/10.48550/ARXIV.2304.02643

12. Semeraro F, Quintart A, Izquierdo SF, Ferguson JC. TomoSAM: a 3D Slicer extension using SAM for tomography segmentation (Version 1). arXiv; 2023 June 14. arXiv: 2306.08609. https://doi.org/10.48550/arXiv.2306.08609

13. Amar AJ, Oliver B, Kirk B, Salazar G, Droba J. Overview of the CHarring Ablator Response (CHAR) Proceedings of the 46th AIAA Thermophysics Conference; 2016 June 13–17; Washington, DC. AIAA; 2016.

14. Schulz, J, Bellas-Chatzigeorgis, G, Stern, E, Palmer, G. Overview of the material response code Icarus. Proceedings of AIAA SciTech; 2023 Jan 23–27; Orlando, FL. AIAA; 2023.

15. Weng H, Martin A. Multidimensional modeling of pyrolysis gas transport inside charring ablative materials. J Thermophys Heat Transf. 2014;28(4):583–597.

16. SIERRA Thermal/Fluid Development Team. SIERRA multimechanics module: Aria thermal theory manual – version 5.10. Sandia National Laboratories (US). Report No. SAND2022-12443. Sandia National Laboratories; 2022 [accessed 2025 June 27]. https://www.osti.gov/servlets/purl/1888149

17. Ferguson JC, Panerai F, Borner A, Mansour NN. PuMA: the Porous Microstructure Analysis software. Software. 2018;7:81–87.

18. Ferguson JC et al. Update 3.0 to "PuMA: The porous microstructure analysis software," (PII: S2352711018300281). Software. 2021;15:100775.

19. Semeraro F, Ferguson JC, Acin M, Panerai F, Mansour NN. Anisotropic analysis of fibrous and woven materials part 2: computation of effective conductivity. Comput Mater Sci. 2021;186:109956.

20. Semeraro F, Ferguson JC, Panerai F, King RJ, Mansour NN. Anisotropic analysis of fibrous and woven materials part 1: estimation of local orientation. Comput Mater Sci. 2020;178:109631. https://doi.org/10.1016/j.commatsci.2020.109631

21. Ji X, Matsuo S, Sottos NR, Cahill DG. Anisotropic thermal and electrical conductivities of individual polyacrylonitrile-based carbon fibers. Carbon. 2022;197:1–9. https://doi.org/10.1016/j.carbon.2022.06.005

List of Symbols, Abbreviations, and Acronyms

2/3D	two-/three-dimensional
C/C	carbon/carbon
CF	carbon fiber
CHAR	CHarring Ablator Response
CT	computed tomography
DL	deep learning
DoD	Department of Defense
FE	finite element
FEM	finite element modeling
IoU	intersection over union
KATS	Kentucky Aerothermodynamics and Thermal-response System
ML	machine learning
MR	material response
NASA	National Aeronautics and Space Administration
PuMA	Porous Microstructure Analysis
RAM	random access memory
ROI	region of interest
SAM	Segment Anything Model
ViT	vision transformed
XCT	X-ray computed tomography

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB