



# Optimization of Local Processing Conditions in Complex Part Geometries Through Novel Scan Strategy in Laser Powder Bed Fusion Process

SANDEEP SRINIVASAN,<sup>1</sup> BRENNAN SWICK,<sup>1</sup>  
and MICHAEL A. GROEBER <sup>1,2</sup>

1.—Department of Integrated Systems Engineering, The Ohio State University, Columbus, OH 43210, USA. 2.—e-mail: groeber.9@osu.edu

Additive manufacturing (AM) involves construction of 3D parts by sequentially adding material to a component and has undergone advancements in the range of materials used and the complexity of parts being printed. Laser powder bed fusion (LPBF) AM, which is the focal point of this work, has sparked interest in the materials and manufacturing community. LPBF has the ability to print complex designs, which may reduce production costs depending on materials, machine, time to print the part, and desired part quality. These complex designs introduce complex processing spaces, resulting in local processing heterogeneities, which may limit the application of LPBF. Hence, there is a need to develop methods to efficiently search for process parameter sets that reduce local processing heterogeneities. We present an optimization methodology implemented to demonstrate the advantages of locally tailored process parameters to produce a more homogeneous component. The optimization is applied to two geometries, using an optimized single parameter set for the entire geometry and locally optimized scan parameters developed based on vector level analysis. Lastly, we show how different optimized scan parameter sets can be related to the different sub-regions in the part in a generalized way to be applied to numerous geometries without retraining.

## INTRODUCTION

Additive manufacturing (AM), unlike conventional subtractive manufacturing, constructs the desired part by adding material to reduce waste. AM has been finding extensive applications in many domains, such as aerospace, automotive, medical, dental, and others.<sup>1</sup> It is valuable to these domains because it can print and fabricate complex parts with intricate geometries rapidly with a good surface finish and accuracy.

Laser powder bed fusion (LPBF) is a hot topic in the metal AM world due to its design flexibility and provision for location-specific process control. However, location-specific process control may be difficult in complex geometries. Complex geometries

have many unique constituent parts that require their own set of location-specific process parameters. Consequently, exploring the AM processing space for acceptable location-specific process parameters can be slow. Our previous work focused on creating a pipeline for integrating physics-based process modeling with machine learning and optimization techniques.<sup>2</sup> The pipeline was illustrated on small-scale, basic geometries that varied in size. This research work is committed to developing location-specific process parameters using component processing history to achieve homogeneity.

How the laser source moves across the powder bed drives LPBF's locally resolved thermal history. Thermal histories (or temperature versus time curves) are an important aspect of a material's processing because they can have strong influences on the thermophysical properties of the material. These properties can affect distortions, mechanical

(Received June 11, 2023; accepted October 23, 2023;  
published online November 28, 2023)

properties, and melt pool characteristics. The laser's movement is dictated by the path planning strategy, which describes the laser positions, powers, velocities, focus, and on times for each layer of the part.<sup>3</sup> Differences in the path planning strategy affect the local processing conditions in the part. Thus, path planning strategy selection is critical because it drives the quality and the properties of the final fabricated part, especially in a complex part geometry with intricate features.<sup>4</sup>

In this paper, we describe a method for controlling the local thermal history of a part by adjusting the laser positions, power, and speed. Section “[Laser Scan Strategies in LPBF and Impact on Local Processing](#)” introduces laser scan strategies in LPBF. Section “[Simulating Thermal Fields in LPBF](#)” outlines our thermal model and how we assess local thermal history in a part. Section “[Optimization of Local Thermal Histories](#)” defines our optimization framework, focusing on quality measurements, local part region definition, proportional control of power and speed, convergence of results, and post-processing. Section “[Results](#)” shows the results of our process to generate a more homogeneous local thermal history for different path planning strategies and part geometries.

### LASER SCAN STRATEGIES IN LPBF AND IMPACT ON LOCAL PROCESSING

During the LPBF process, the laser is translated on each layer with the goal of melting the powder at all points within the polygon(s) defined by the intersection of the layer with the 3D component being printed. There are endless laser paths that could achieve this goal, but the most common scan strategy in LPBF follows a serpentine pattern. It is a simple strategy where the laser rasters over the powder bed in a ‘back-and-forth’ pattern. Each

segment of the path where the laser is moving in a constant direction is generally referred to as a vector. When the laser reaches the end of a vector, it translates perpendicular to that vector a small distance, often referred to as the hatch spacing, and then moves along another vector that is anti-parallel to the previous vector. In this strategy, the length of the vectors has a large impact on the thermal histories within the part, since they control how long it takes for the laser to move away from and then back towards a given position in the component.

Typically, LPBF machine vendors will establish general scan strategies that are then intersected with the components the users want to print. To keep the vectors short and bind thermal histories, the general laser scan strategy will predivide the build domain into subregions. There are multiple methods for doing this (e.g., checkerboards or lanes/stripes), but the common motivation is to standardize the length of the vectors the laser moves along. The vectors will run from one side of the subregion to the other but not extend past the subregion (Fig. 1). Typically, the subregions will be rotated from layer to layer to ‘randomize’ the directions of the vectors. After all the vectors are predefined, they are then intersected with the part boundaries on each layer, with any portion of any vector falling outside of the component being discarded. As shown in Fig. 1a, the vector in the bottom right consists of three segments inside the part: (1) a short segment bounded by the part geometry and a predefined boundary; (2) a max-length segment bounded by two predefined boundaries; (3) a near max-length segment bounded by a predefined boundary and the part geometry. Overall, this strategy produces many vectors that are of similar length that run from one side of a subregion to another; however,

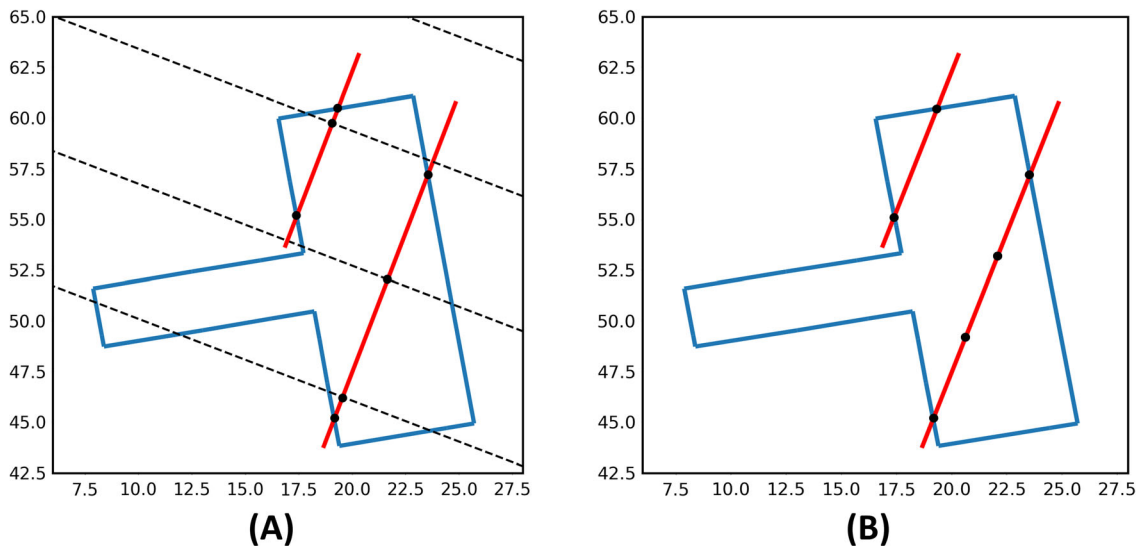


Fig. 1. Illustration of vector generation using the common industry scan strategy (A) and custom scan strategy (B). Predefined stripe boundaries for the common industry strategy are represented with dotted lines.

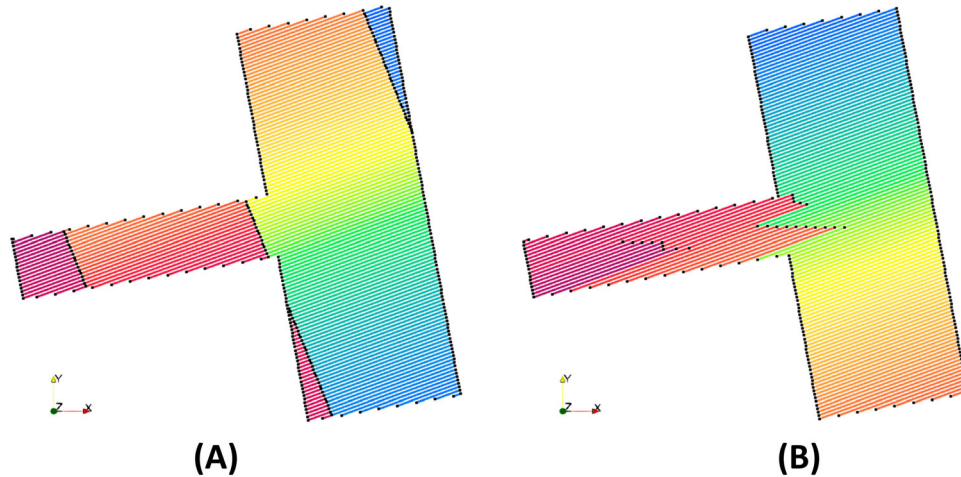


Fig. 2. Vectors in the common industry strategy (A) versus the custom scan strategy (B). The black lines in the common industry scan strategy represent the predefined stripe boundaries that drive vector segmentation along with the part boundaries. Coloring corresponds to the order in which the vectors are traversed by the laser, with blue being first and red being last (Color figure online).

there is no control of very small vectors that result from the intersection of vectors with the part boundaries. Notably, some LPBF machine vendors try to implement adjustments for this, but they are generally ad hoc.

One could design a subtly different approach that does not predefine vectors, but rather generates vectors specific to the component geometry being printed, to ensure more uniform distribution of vector lengths. One such custom scan strategy is shown in Fig. 1, which uses the same parallel vectors on each layer as the common industry strategy but implements a different segmentation strategy. Instead of using predefined boundaries and then the part geometry borders, the custom strategy starts segmentation with the part geometry borders. The intersections of the part borders and an infinite vector are determined, and the lengths of the vector segments are calculated. Similar to the common industry strategy, it is desirable to segment these vectors if they are too long. A maximum length can be defined, and if a vector is longer than the maximum, it is segmented into  $n$  equal lengths, where  $n$  is the number of maximum length vectors that can fit on the vector plus one. Figure 1 illustrates the results of the segmentation steps on two different vectors. The common industry strategy creates short segments that are not in the custom scan strategy. The custom strategy does not have predefined stripe boundaries, which allows the segmentation to be flexible, preventing scan vectors with lengths at the extremes. Figure 2 shows the results of the two scan strategies for a single layer of a simple component. The color in the image represents the time the vector was traversed by the laser, with blue being first and red being last.

## SIMULATING THERMAL FIELDS IN LPBF

### Thermal Model

The scan path planning strategy directly affects the local thermal histories that are generated because the laser positions, powers, velocities, focus, and on time all affect the local energy input. Modeling how the scan path of the laser affects the local processing within the component is advantageous because it eliminates the need to collect empirical data on laser scan path parameters that produce the desired material properties in the component. Acquiring such empirical data in LPBF requires a thorough investigation of a large parameter space, which is cost and time intensive. This cost can be avoided by modeling the results of laser scan paths. Due to the large potential laser scan path parameter space, it is important that the thermal model used is computationally efficient. To achieve this efficiency, the thermal model used in this paper does not consider power losses brought on by laser plumes, powder dispersion, or melt pool dynamics such as waves, sloshing, or convection.<sup>5</sup> These parameters could and may be modeled by other groups, but we excluded them because our model offers a sufficient illustration of the LPBF thermal effects for this parameter space search method at an acceptable computational speed.

The thermal model used in this work is presented in significantly higher detail in Ref. 6 and in similar work.<sup>7</sup> The model represents the laser's continuously moving Gaussian energy source as a discrete stationary distribution of Gaussian points in space. The discrete Gaussian energy source representation is used to determine the local thermal histories via an analytical thermal model, derived using a Green function approach. The analytical thermal model gives the relationship for the temperature at the

location  $\vec{r}_j$  at a time 't' due to all sources  $\vec{r}_i$ , as represented in Eq. 1,

Analytical thermal model for discrete energy source

$$T(\vec{r}, t) = \sum_i^N \frac{\eta P_i \Delta t \Theta(t - \tau_i)}{\rho c_p \sqrt{2} [\pi(\sigma^2 + 2\sigma(t - \tau_i))]^{3/2}} \exp \left[ -\frac{(\vec{r} - \vec{r}_i)^2}{2\sigma^2 + 4\sigma(t - \tau_i)} \right] \quad (1)$$

where  $\eta P_i$  is the effective power at the source,  $P$  is the nominal power,  $\eta$  is efficiency,  $t - \tau_i$  is the time since source 'i' was activated,  $\Delta t$  is the source discretization,  $\Theta$  is the Heaviside function,  $\sigma$  is the beam radius,  $\rho$  is mass density,  $c_p$  is the heat capacity, and  $\alpha$  is the thermal diffusivity.

In this work, the scan paths are fixed, which fixes the locations of all Gaussian point sources. During the optimization process, the power and speed of the individual vectors will be modified, which in turn set the power and times associated with each Gaussian point source.

### Thermal History Dimensionality Reduction

The thermal history output from the thermal model has high dimensionality because it describes the temperature at a point for each time step in the simulation. Consequently, computing the thermal history changes at a point would be computationally prohibitive. Therefore, a reduced space needs to be identified that can track the most relevant details of the thermal history, which may be material and application dependent.

The thermal model output is discretized using a SAX approximation,<sup>8</sup> which produces a categorical labeling at each time step rather than an exact temperature; see Fig. 3. In this work, the temperature values were assigned to one of five categories, which correspond to phase transition temperatures

for the titanium alloy being simulated. The categorization itself does not reduce the dimensionality of the thermal history. For this paper, the thermal histories were reduced to the total time a point spent in the category associated with being in the liquid phase (i.e., the total time molten). The total time molten was selected as the parameter to optimize because it correlates to potential material defects.<sup>9</sup> Moreover, the total time molten is controlled by the input energy density, which is a function of the laser power and the time spent near the point. These values are directly controlled by the scan path planning strategy discussed previously. Total time molten is not the only relevant metric for controlling part quality, but it was selected as a starting test case. Additional metrics that are relevant to microstructure formation, residual stresses, or other aspects of part quality could also be included.

## OPTIMIZATION OF LOCAL THERMAL HISTORIES

### Quantification of Vector Contribution to Quality

The thermal model provides point histories but does not directly describe the thermal history for a vector in the scan path planning strategy. To address this, points were sampled in a boundary around the vector, and their thermal histories were grouped to develop a parameter that could define the melt quality of a vector. In addition, thermal influences on the vector from nearby vectors were considered to develop a metric that considered the local heating of LPBF.

Points were sampled around the vectors using the melt pool as a boundary (Fig. 3). The vector's melt pool has the strongest influence on points within one layer thickness and one hatch spacing centered

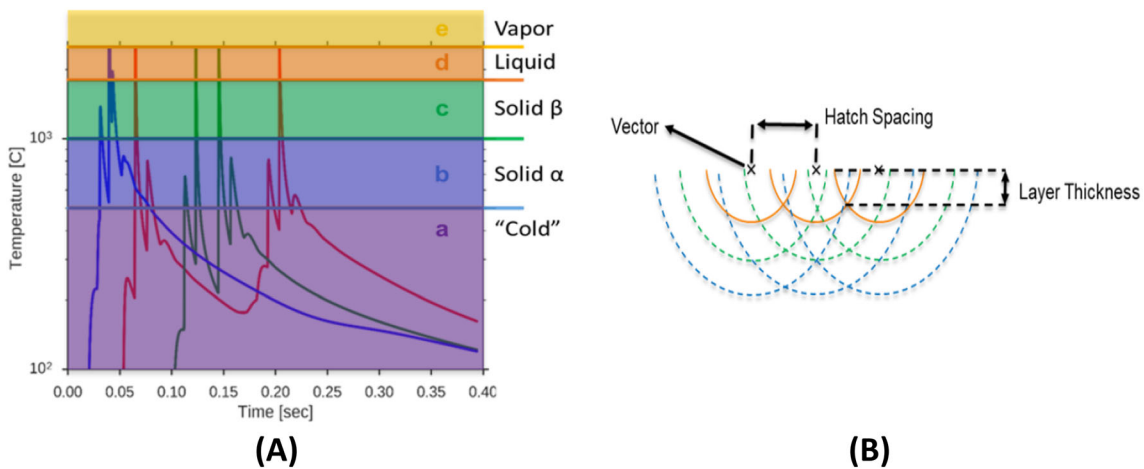


Fig. 3. Cross section of a scan vector melt pool in the perpendicular direction including influences from the neighbors (A). Thermal histories for a scan vector with the relevant isotherms for an exemplar metal (B). The colored lines in the cross section match the isotherm ranges in the thermal history graph. The orange line in the cross section is the only area where points will reach the liquid phase. As shown, there is a small overlap in the liquid phase between the middle vector and either neighbor (Color figure online).

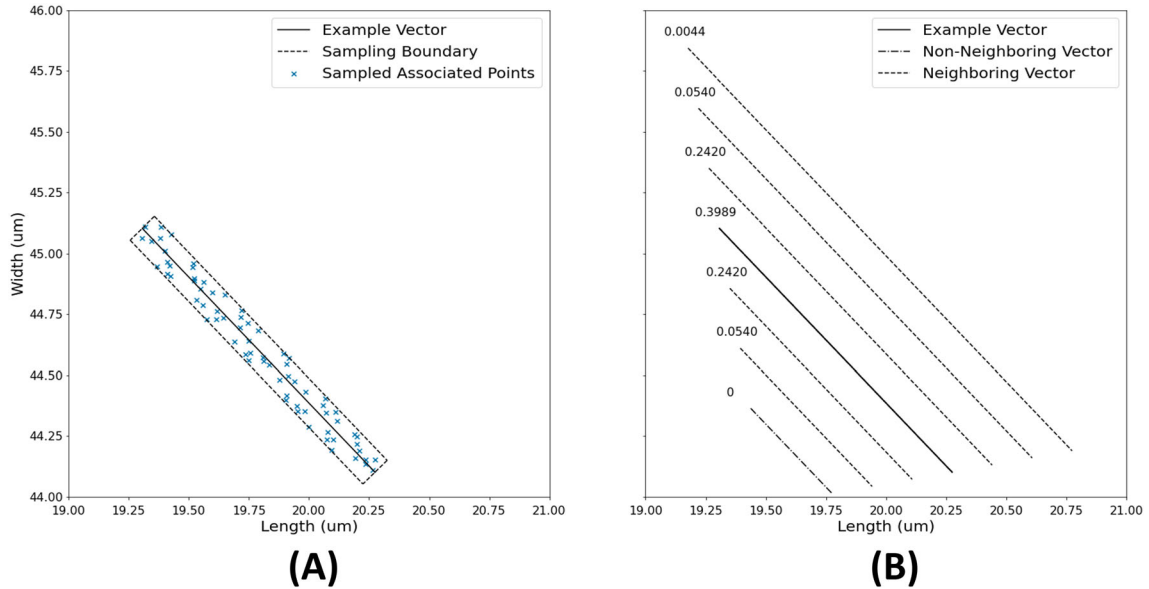


Fig. 4. Latin hypercube sampled points shown in a 2D space within a boundary defined by the hatch spacing width and length of the scan vector (A). Identification of scan vector neighbors and their associated Gaussian weights based on the overlap with a smoothing window of  $\sigma = \frac{5}{3}$  (B).

on the vector. For every layer, points were sampled using a Latin hypercube sampling<sup>10</sup> method in a three-dimensional sampling space around the vectors (Fig. 4). This sampling space was bounded by the layer thickness, hatch spacing, and length of the vector to simulate the points that would be most influenced by the vector. A uniform sampling density of 50 points/mm was used to create a consistent representation despite the varied length of the vectors. The sampling space is defined by a rectangular domain that surrounds the vector with principal axes along: (1) the length of the vector; (2) the direction perpendicular to the vector in the surface of the layer; (3) the direction perpendicular to the layer (i.e., the build direction). The domain was bounded by the length of the vector, a hatch spacing centered on the vector, and the layer thickness, in the three respective directions. After these points were sampled, the coordinates were converted to the x, y, and z space of the vectors.

The total time molten for the sampled points was compared to a set desired total time molten of 1.2 ms. This desired time was selected to prove the concept of the paper that the total time molten can be controlled but it does not have a known relation to desirable material properties. The difference between a point's total time molten and the desired total time molten was defined as the 'thermal error' for a point. A scalar value for the vector's melt quality was generated by taking a weighted average of the thermal errors based on the point's perpendicular distance from the vector. The scalar value for the vector's melt quality was labeled the vector thermal error. A positive vector thermal error corresponded to over-melting of the vector and a

negative vector thermal error corresponded to under-melting of the vector.

### Defining a Descriptor of Local Scan Path

During the optimization process, each vector in the laser scan path could be treated individually and its parameters updated independent of any other vectors. A drawback to this approach is the large number of vectors to optimize over and the inability to transfer the knowledge learned from one component to another. Instead, a feature descriptor was developed that could be used to categorize similar vectors into groups. Categorizing the vectors allows for the optimization to select parameter updates for a set of vectors, which has benefits of: (1) reducing the total number of parameter updates; (2) smoothing the update actions by averaging over multiple similar vectors; (3) learning a parameter set for each category. The last benefit provides generalizable parameters tied to a vector descriptor that could be applied to vectors in a different geometry without rerunning the optimization process.<sup>11,12</sup>

The categorization of vectors into groups of similar vectors is dependent on the definition of similar. In this work, the aim was to group vectors where the laser's local raster path creates a similar sequence of energy deposition. A three-component descriptor was developed that captures information about the vector and its nearest neighbors. A vector's nearest neighbors correspond to vectors that are near it in both space and time. The LPBF process is local, meaning energy entered into the system at a point spatially removed from another has little effect on the temperature at that point.

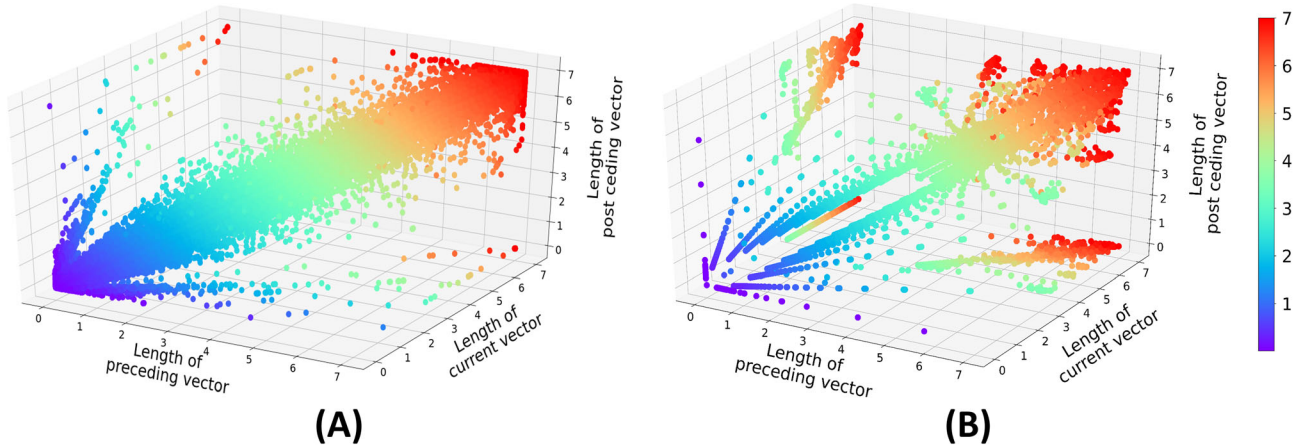


Fig. 5. Reference and neighboring vectors in the tensile bar with common industry (A) and custom scan strategy (B).

Additionally, energy entered at a moment temporarily removed from another has little effect on the temperature at that time. As such, the nearest neighbors for a particular vector were determined using the temporal distance from and the projected spatial overlap with the reference vector (Fig. 4). Projected overlap was selected to determine the spatial nearest neighbors because heat most rapidly dissipates perpendicularly to the direction of travel of the laser. To be a nearest neighbor of a reference vector, a vector must meet these criteria: (1) the vector covers  $> 50\%$  of the length of the reference vector, (2) the vector is lazied within a time period that corresponds to the time necessary to print 5 maximum length vectors. Only one pre- and one post-ceding vector are selected for a given reference vector. If there are multiple pre- or post-ceding vectors that meet both criteria, the closest to the reference vector is selected. The three components of the descriptor are the lengths of the preceding, reference, and post-ceding vectors. The length of each vector was chosen for the length's impact on thermal history noted in the previous section. The descriptor maps each vector to a point in a 3D space. Furthermore, the 3D descriptor space can be discretized into 'bins' that group vectors that have similar nearest neighbors and thus similar energy input schedules.

The descriptor space of a rectangular cuboid component, referred to from here as a tensile bar, is visualized in Fig. 5 to compare vector distributions for the common industry and custom scan strategy. The custom scan strategy has a sparser distribution and, therefore, fewer unique vectors compared to the common industry scan strategy. This implies that the custom scan strategy generates a more homogeneous vector distribution than the common industry scan strategy. Consequently, scan path plans generated with the custom scan strategy are expected to have fewer unique local scan regions and require fewer parameters to control the thermal history.

The number of unique parameter sets used within the component and the amount of averaging across vectors can be adjusted by changing the bin size of the descriptor space. We selected two different parameter sets and bin sizes for the optimization step. A single parameter set was selected that used a single bin to control the power and velocity for every vector in the entire part. This was selected to replicate the current method for controlling power and velocity on typical LPBF machines. In addition to the single parameter set, a large parameter set was selected that used  $20 \times 20 \times 20$  binning in the descriptor space. This binning yielded 8000 unique powers and velocities that could be used to adjust different vector groupings. The large parameter set was compared against the single parameter set to determine the impact of using more parameters to homogenize the thermal histories.

### Scan Vector Parameter Update Strategy

The scan vector parameters were updated using proportional control with clipping. The idea for the update is based in the classic control example of building temperature regulation using a thermostat. The scan vector neighborhoods can be thought of as rooms that should be kept at a consistent temperature. Thus, in a process similar to a thermostat, we compare the difference between the set point and the actual temperature of a scan vector, which has been captured in the vector thermal error, to make our adjustments. The vector thermal error is used to update the input energy density via heuristic knowledge. Specifically, the input energy density is adjusted using the power and velocity in the scan path planning strategy parameters. If the scan vector neighborhood is too cold, or undermelted, the input energy density is increased by increasing the power and decreasing the speed. If the scan vector neighborhood is too hot, or overmelted, the input energy density is decreased by decreasing the power and increasing the speed. The

power and velocity were bounded by the operating conditions of a real-world LPBF machine.

Calculating the vector thermal errors with a weighted average led to large vector thermal errors when there were large point thermal errors. These large vector thermal errors generated large parameter updates, which caused fluctuations around the set point. Again, this can be compared to thermostat fluctuations, which are undesirable because they increase the settling time. A clipping function was introduced to reduce the magnitude of the update and decrease fluctuations. Multiple clipping functions were compared and each function bounded the update by 5% of the magnitude (Fig. 6). The  $\tanh^2$  function was selected as the best clipping function and was used to update the power and velocity for each scan vector on every layer.

Updates to the power and velocity were smoothed using a Gaussian kernel (Fig. 4). This was done to account for the local influence the vectors will have on each other. Specifically, if the input energy density increases for each vector, then the local input energy density will increase, and each vector will require less energy to reach the desired total time molten. This is similar to the heating or cooling effect that connected rooms in a house will have on each other. If the heat transfer between the rooms is not considered, the system will over-compensate and miss the set point. Hence, the smoothing is a method to account for the heat transfer in the local processing region of a vector and prevent over-compensation by the controller.

This proportional control method was used to optimize the scan vector parameters on each layer in the scan path planning strategy. The layer's sampled points had converged if: (1)  $> 99\%$  were

molten  $> 0$  ms; (2)  $\geq 95\%$  were molten  $> 0.5$  ms; (3)  $< 5\%$  were molten  $> 0.9$  ms. These thresholds were chosen to show that the distribution of the vector thermal errors could be controlled. In addition, an early stopping criterion was used for layer optimization. The early stopping criterion measured the absolute difference between sampled points with a total time molten  $> 0$  ms for: (1) the current iteration and (2) the moving average over the last three iterations. If the absolute difference dropped below 0.005, the optimization had converged.

### Clipping Functions to Reduce Fluctuations

The following clipping functions were compared: (1) 'scaled'; (2)  $\tanh$ ; (3)  $\tanh^2$ ; (4)  $\tanh^3$ . The 'scaled' function updates each vector based on the fraction of that vector's thermal error divided by the maximum vector thermal error on that layer. Consequently, varying max vector thermal errors on the layers will generate different outputs for the same input vector thermal error when using the 'scaled' function. The hyperbolic tangent clipping functions were selected because they are a commonly used to limit deep neural network weight updates.<sup>13</sup> In addition, each hyperbolic tangent function has different behavior around 0, which will affect the ability of the system to settle around the set point based on the vector thermal error. The  $\tanh^2$  function will only give positive outputs based on the vector thermal error, so the output was multiplied by the sign of the vector thermal error.

The clipping functions were evaluated based on the difference between the results and the desired total time molten and the number of iterations until convergence. The optimization was run for each clipping function on one layer of a simple

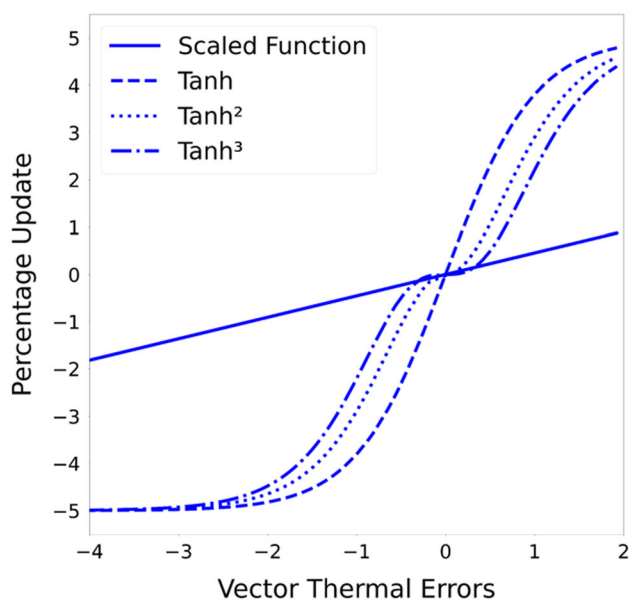


Fig. 6. Example outputs for different clipping functions based on vector thermal error. As the power of the hyperbolic tangent function increases, the plateau of the output around 0 increases.

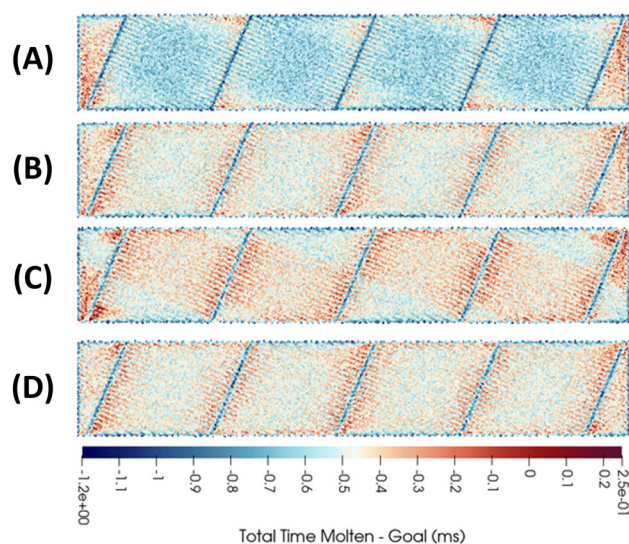


Fig. 7. Difference between the optimized results and the desired total time molten as obtained using different clipping functions. The rows show (A) the tensile bar before optimization, (B) 'scaled' results, (C)  $\tanh$  results, and (D)  $\tanh^2$  results (Color figure online).

rectangular cuboid component (Fig. 7). The ‘scaled’ and  $\tanh^2$  functions create a visually more homogeneous part than the  $\tanh$  function. This occurs because the  $\tanh^2$  and the ‘scaled’ functions have a similar slope near 0. The  $\tanh$  function, however, has a steep slope near 0 that may perturb low vector thermal errors and cause the system to over-correct. Consequently, more energy is added to bulk and corner regions that have reached a steady state until the optimization process terminates. The darker blue ‘lines’ that are present in the bulk, as well as near the perimeter of the component, are the locations where the laser reaches the end of a vector and turns off while traversing to the next vector. There are sophisticated laser acceleration and deceleration parameters that can be implemented on LPBF machines to attempt to mitigate these features. It is out of the scope of this work to optimize those parameters as well though that could be done with this same approach.

The point thermal error distributions were examined for a more detailed comparison (Fig. 8). These distributions confirm that the ‘scaled’ and  $\tanh^2$  functions have tighter distributions closer to the desired total time molten than the  $\tanh$  function. Also, the  $\tanh$  function has a longer tail, which indicates that the part is overheating in certain regions. The  $\tanh^2$  function has a higher peak, which means that it has more points near the desired total time molten. Additionally, the left tail is slightly shorter, and the right tails are the same for the  $\tanh^2$  and ‘scaled’ functions. Therefore, the  $\tanh^2$  function created fewer under-melted points while maintaining the same number of over-melted points. The  $\tanh^2$  function converged the fastest,

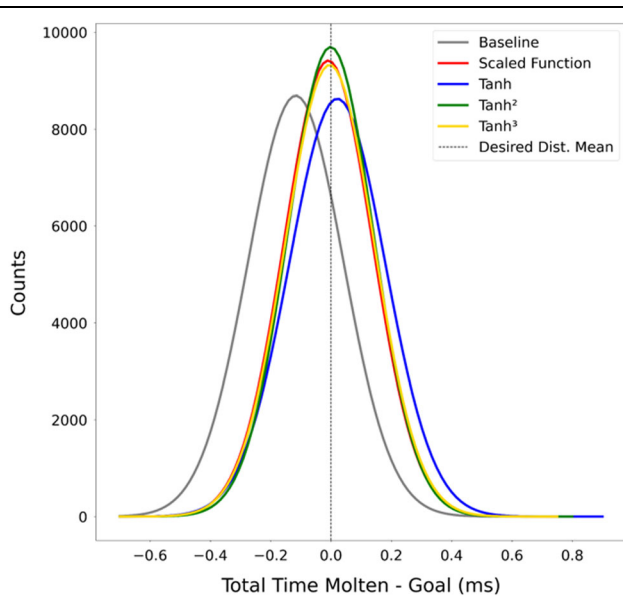


Fig. 8. Distributions of the point thermal error for the optimized tensile bar results obtained via the different clipping functions. The point thermal error is the difference between the optimized results and the desired total time molten.

taking just 5 iterations compared to the 12 iterations required for the ‘scaled’ function and 20 iterations required for  $\tanh$ . In short, the  $\tanh^2$  function produced less error between the results and the desired total time molten while also converging faster. Thus, it was selected as the best clipping function and was used for all optimization runs going forward.

#### Accounting for Heat Transfer with Smoothing

Once the best clipping function was chosen, the influence of the smoothing parameter,  $\sigma$  on the optimization results was examined. This value assigned weights to the nearest neighbors of a reference vector by using a Gaussian weight kernel (Fig. 4). Nearest neighbors were determined based on distance and projected overlap. To be a nearest neighbor of a reference vector, a vector must meet these criteria: (1) the vector covers  $> 50\%$  of the length of the reference vector and (2) the vector is one of the pre- or post-ceding vectors in the smoothing window of the reference vector in a time sorted list. Both criteria ensure the neighboring vector has a non-negligible influence on the thermal history of the reference vector. Again, if we compare this to building heat transfer, small rooms or far away rooms are not likely to significantly contribute to the temperature in a given room.

These values, combined with the  $\tanh^2$  function, were used to obtain new optimization results. The results showed no visible difference, and the distributions were similar for all values of  $\sigma$  (Fig. 9). However, these values did affect the speed of convergence, and  $\sigma = 5/3$  reached the optimized state in fewer iterations. The similarity in the results may be explained by the melt pool overlap

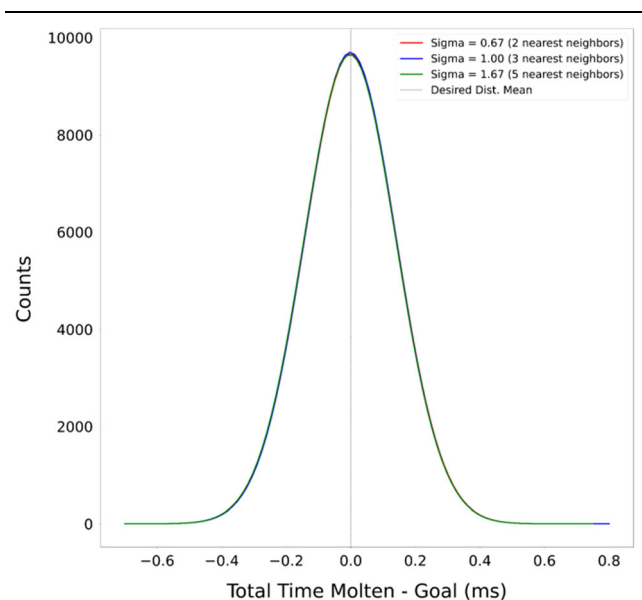


Fig. 9. Distributions of the point thermal error for the optimized tensile bar results obtained via the different  $\{\sigma\}$  values.



of neighboring vectors (Fig. 1). The melt pool of a scan vector is mostly isolated from its neighbors, except for some small overlap from the closest vectors. Therefore, it should be expected that the input energy required for melting will only transfer between the vectors on either side of the current vector. Consequently, we expect that any smoothing window considering more than one neighbor, or  $\sigma = \frac{1}{3}$ , will not affect the optimization because not enough heat transfers for melting. However, it may be valuable to consider the smoothing window and heat transfer effects for lower temperature phase transitions such as the  $\alpha$  or  $\beta$  phase. These transitions require less input energy density, and it would be important to consider the heat transfer in a larger neighborhood.

### Convergence of Optimization Results

After determining the best clipping function to update the power and speed for the vectors on a layer and selecting a  $\sigma$  value that smoothed the update to account for local thermal influences from nearby vectors, all the layers were optimized for two different geometries. The geometries optimized were a tensile bar and a T-object. The tensile bar had 115 layers, and the T-object had 500 layers to be optimized, where each layer was rotated by  $67^\circ$  according to LPBF conventions. For the first layer, the scan vectors are initialized with a baseline power of 300 W and a baseline speed of 1300 m/s. After each layer is optimized, the scan vectors for the next layer are initialized by checking them against bins of similar scan vectors in similar part regions on previous layers. If a similar scan vector has not been seen on a previous layer, it is

initialized with the baseline conditions of power and speed. Once all layers were optimized, the convergence of the results for each geometry was examined to confirm the method was reducing the vector thermal error and driving the geometry towards the desired total time molten. In addition, a support vector regression (SVR) function<sup>14</sup> was fit on the bins that described similar scan vectors in similar part regions to determine whether general part regions could be optimized in a single geometry and transferred across layers or across geometries.

For the layers in the geometry, the scan vectors were binned in a  $20 \times 20 \times 20$  dimensional space based on their lengths and the lengths of their pre- and post-ceding neighbors. Each bin consisted of the number of vectors in each bin, their powers, and their speeds. These values created an average power and speed associated with each bin that was used to update the power and speed for any additional vectors in that bin on subsequent layers. After optimizing each layer, the output was an  $N \times N \times N$  of  $N \times N \times N$  matrix of powers and speeds. These powers and speeds were associated with a total time molten for the layer. For subsequent layers, the difference between the total time molten was found using the L2 norm (Fig. 10). The optimization is reducing the difference in the total time molten for each geometry and either reaching the desired total time molten criteria or the early stopping criteria.

### Smoothing Optimization Results

Once the optimization was carried out for a geometry, the output  $n \times n \times n$  matrix of powers and speeds was stored. Two different cases were examined in this work,  $n = 1$ , or the single parameter condition, and,  $n = 20$ , or the large parameter condition. In the case of the large parameter condition, support vector regression (SVR) was used to fit a functional representation of the data in the 3D descriptor space.<sup>14</sup> To fit this function, first, the matrix was preprocessed to remove powers and speeds equal to zero. Then, the resulting data were split into 80% training and 20% testing data. Next, the hyperparameters were narrowed down using a grid search technique, and the best hyperparameters were selected using the k-fold cross-validation technique.<sup>15</sup> After determining the hyperparameters, the SVR surface was fit on the training data and predictions made on the testing data yielded  $R^2$  values of 0.8985 for the power and 0.9153 for the speed. These values show a strong positive correlation between the vector groups and the optimized powers and speeds. Thus, we suspect a general trend exists between the scan vector parameters and the vector descriptor defined in Section “[Defining a Descriptor of Local Scan Path](#)” that can be estimated with a surface. This surface can be used to predict the vector parameters given a vector

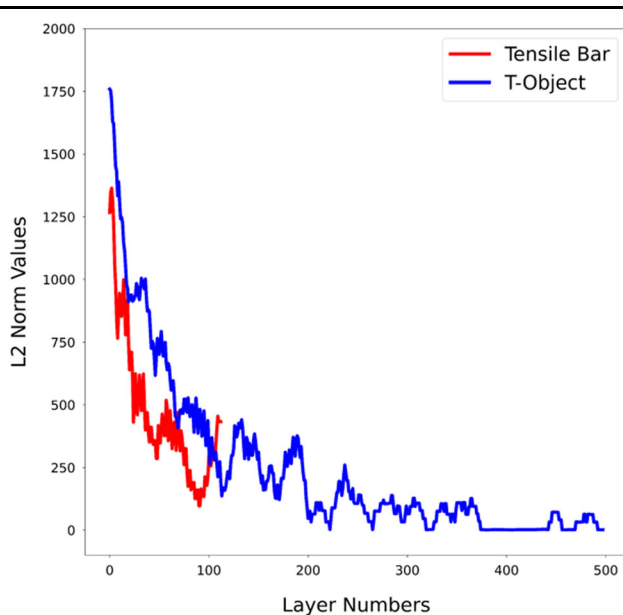


Fig. 10. L2 norm difference between the previous and current layer's optimized results for the tensile bar and T-object.

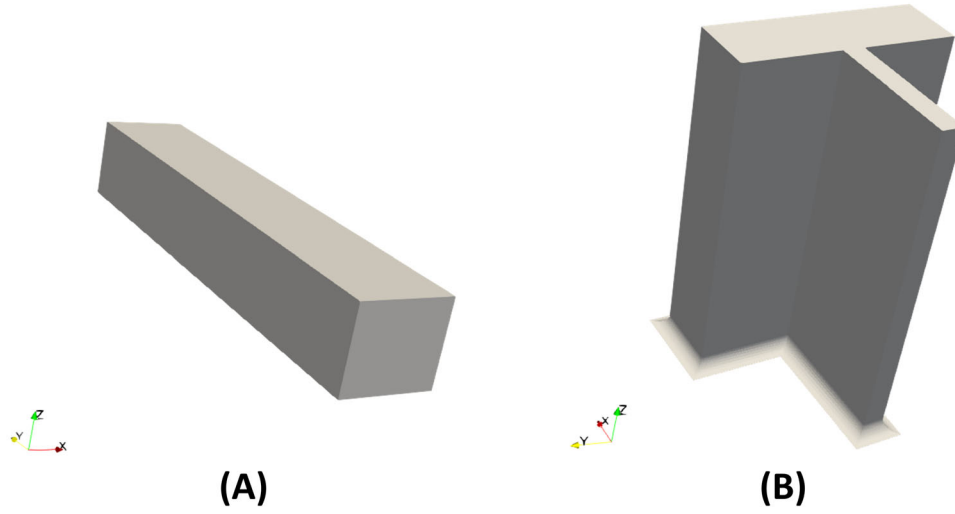


Fig. 11. Tensile bar (A) and T-shaped object (B).

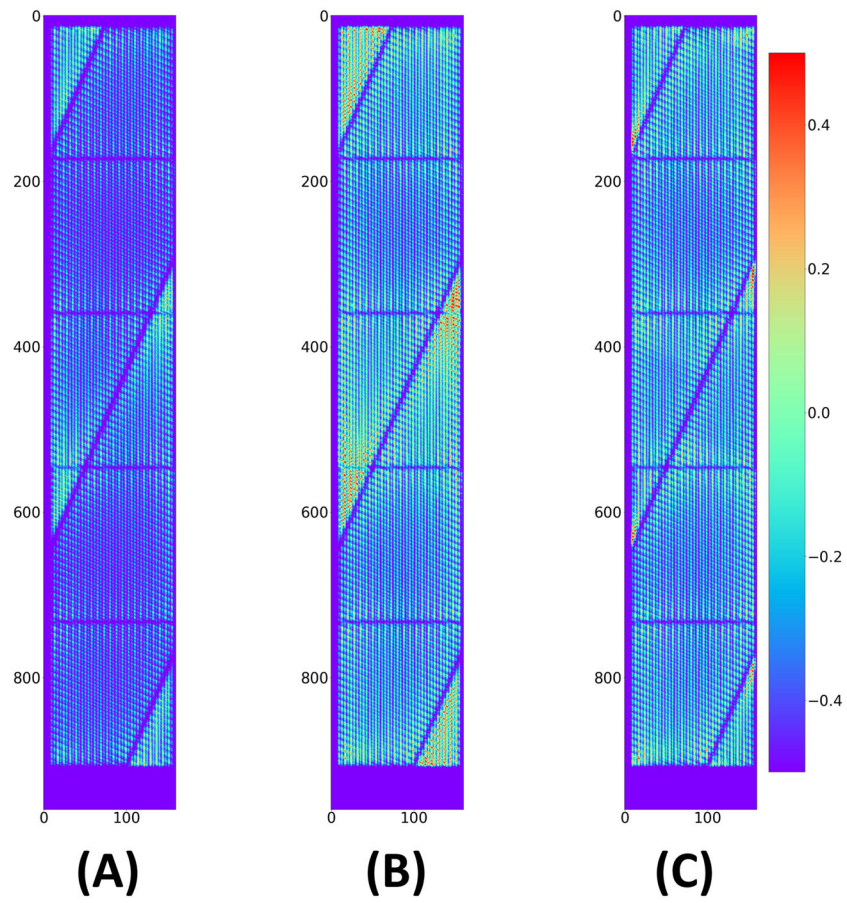


Fig. 12. Baseline and optimized tensile bar results obtained using the common industry scan strategy. Results shown are baseline (A), optimized single parameter (B), and optimized large parameter (C). The color bar represents the difference between the total time molten and the desired total time molten (Color figure online).

descriptor for a vector in any component and reduce the need for simulation and optimization runs. Having this functional form of the relationship allows a user to choose any level of control they would like regarding the discretization of the vector descriptor space.

### RESULTS

The results presented in this work are from two component geometries to show variations between components of different geometric complexity. The first geometry considered was a simple rectangular cuboid (referred to as tensile bar in this paper). The second geometry considered was a T-shaped object, with slightly more complex features like thin and thick cross sections. The geometries are shown in Fig. 11.

In the analysis of the results, the goal time molten was estimated to be  $\sim 1.2$  ms. Notably, each layer was optimized individually and did not directly optimize the time molten in 3D, where subsequent layer melting would increase the total time molten at a given location in the component. The optimization criteria on each layer drove the time molten towards 0.5–0.7 ms within the layer but would also remelt the previous layer for a shorter time. Typically, a layer remelts two layers below but for increasingly shorter times. It was estimated that a given point in the component, with ideally optimized vectors, would be melted for 0.6 ms, 0.4 ms, and 0.2 ms by the three layers that would melt it (totaling 1.2 ms). This estimate is noted on all plots as the goal time molten but should be treated as a rough estimate and is not what was actually the optimization criterion.

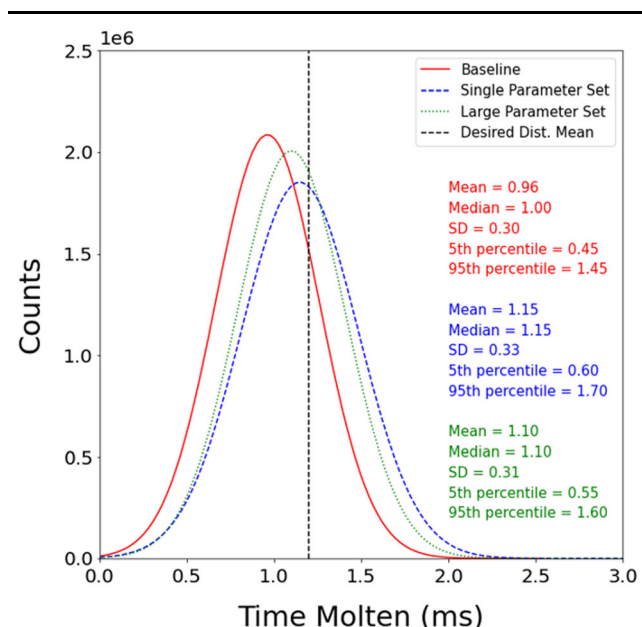


Fig. 13. Baseline and optimized tensile bar distributions using the common industry scan strategy.

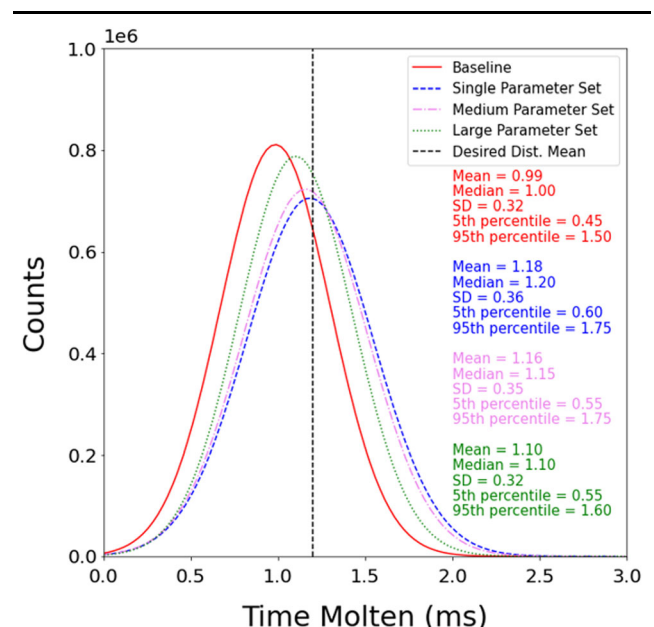


Fig. 15. Baseline and optimized T-object distributions using the common industry strategy.

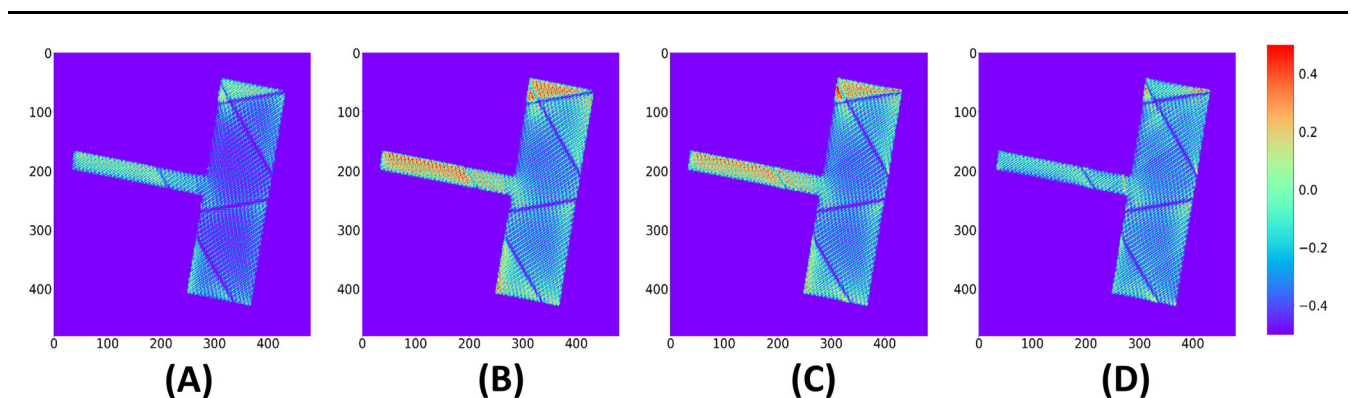


Fig. 14. Baseline and optimized T-object results obtained using the common industry scan strategy. Result shown are baseline (A), optimized single parameter (B), optimized middle parameter (C), and optimized large parameter (D). The color bar represents the difference between the total time molten and the desired total time molten (Color figure online).

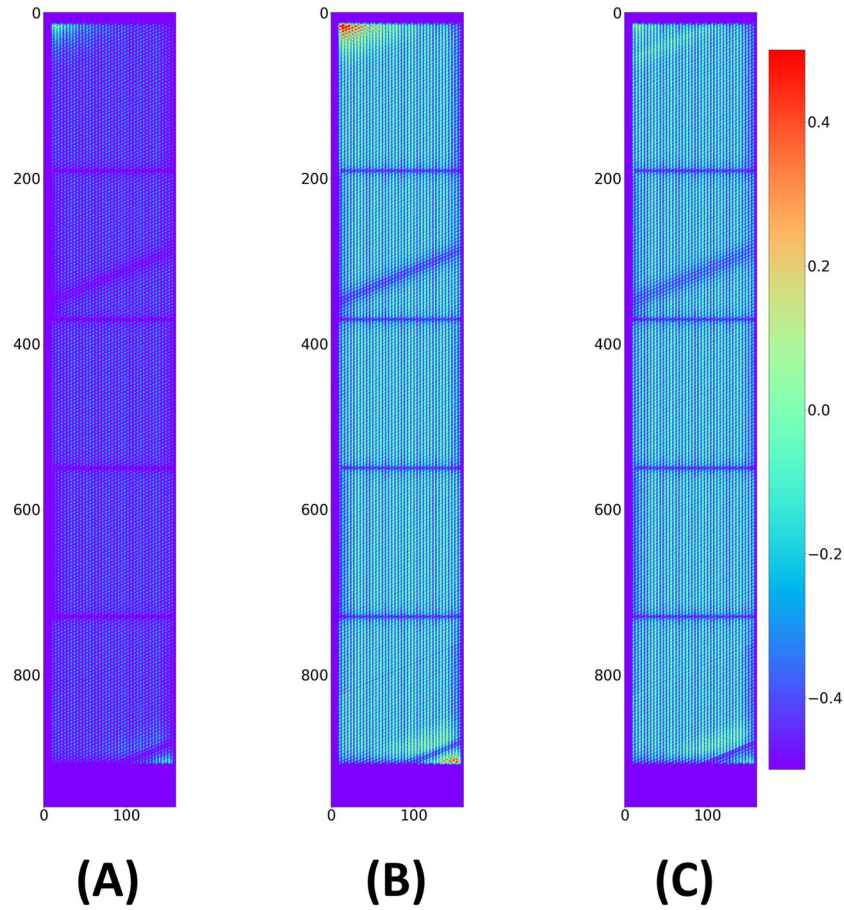


Fig. 16. Baseline and optimized tensile bar results obtained using the custom scan strategy. Results shown are baseline (A), optimized single parameter (B), and optimized large parameter (C). The color bar represents the difference between the total time molten and the desired total time molten (Color figure online).

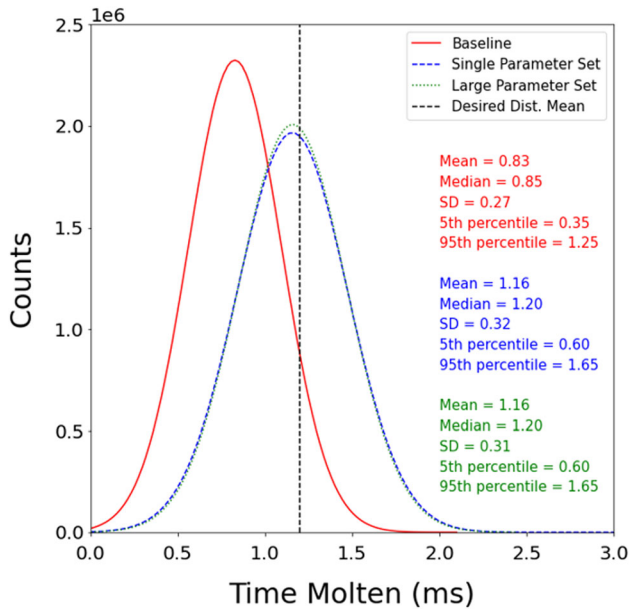


Fig. 17. Baseline and optimized tensile bar result distributions using the custom scan strategy.

### Common Industry Strategy

Figure 12 shows the results of the tensile bar optimized using the common industry scan strategy with the selected clipping function and value. The common scan strategy creates triangular corner regions within the part where the vectors are all very short. The baseline parameter set (300 W and 1200 mm/s) shows that large portions of the part are below the desired time molten with increased total time molten in the corners. The optimized single parameter set returned an optimal power of 311 W and an optimal speed of 1257 mm/s. This parameter set improved the melt quality of the bulk regions at the expense of the corner regions. The single parameter cannot address the differences in local processing conditions in the bulk and corner regions. The large parameter set is able to reduce the differences by using more parameters sets to balance the needs of the corner and bulk regions.

The visual results are supported by the distributions shown in Fig. 13. The baseline has the lowest mean total time molten followed by the large parameter set and then the single parameter set.

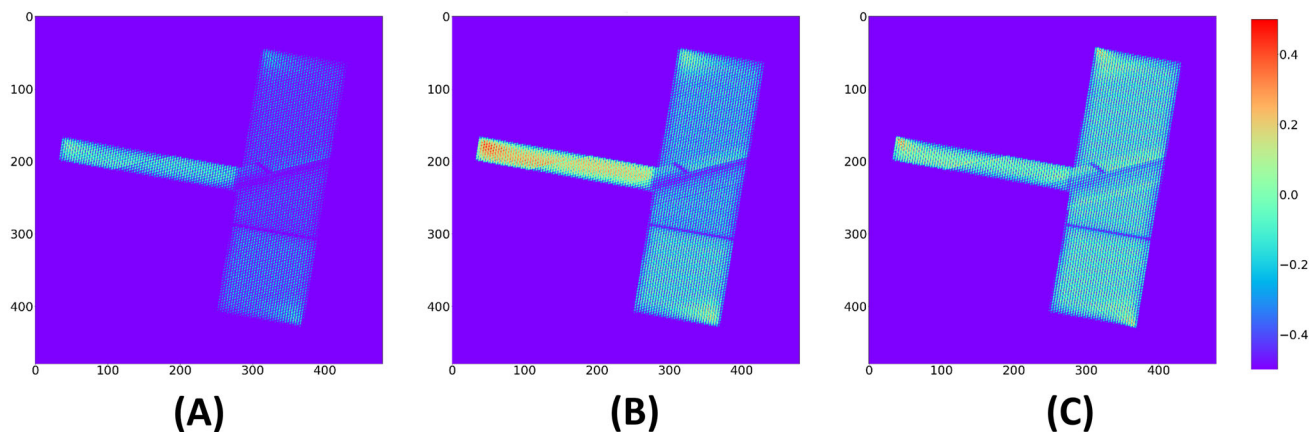


Fig. 18. Baseline and optimized T-object results obtained using the custom scan strategy. Results shown are baseline (A), optimized single parameter (B), and optimized large parameter (C). The color bar represents the difference between the total time molten and the desired total time molten (Color figure online).

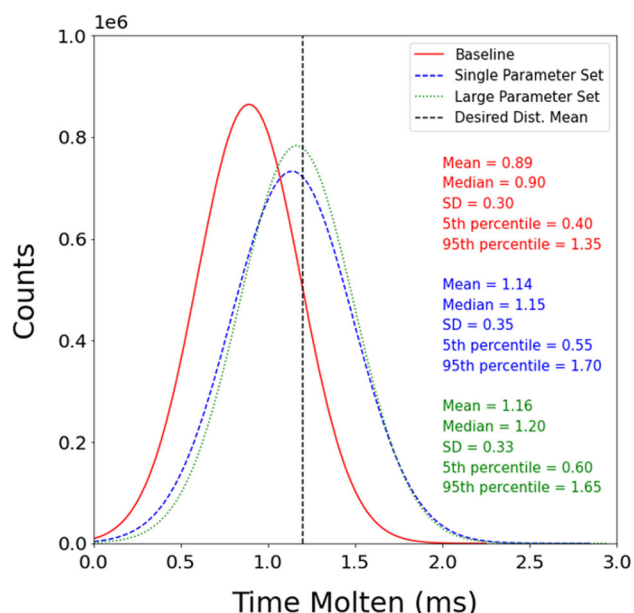


Fig. 19. Baseline and optimized T-object result distributions using the custom scan strategy.

This confirms that the optimization strategy can control the scan vectors to reach a desired total time molten for the part. The large parameter set generates a more homogeneous part, as seen by a lower standard deviation. However, the large parameter set is farther from the desired set point than the single parameter set. This may be an effect of the control strategy falling into local minima. Importantly though, the large parameter set reduces the severely overmelted points that are present in the single parameter set case.

Similarly, the T-object was optimized using the common industry scan strategy, and the results are shown in Figs. 14 and 15. The results followed similar trends seen in the tensile bar. An additional

parameter set size, termed medium parameter set, which uses a  $5 \times 5 \times 5$  discretization of the vector descriptor space was added to determine the minimum level of control needed for the common industry scan strategy. The medium parameter set had a mean total time molten between the single and large parameter sets, which followed an expected trend. As the level of control increases, so does the homogeneity of the part, though not linearly. It is clear again that the large parameter set case is able to significantly reduce the severely overmelted regions. Because the large parameter set case can provide more control of the vectors, it is able to keep the vectors closer to the individual layer goal of 0.5 ms. This is likely the reason the full 3D time molten (with remelting from subsequent layers) is slightly lower than the estimated target of 1.2 ms.

### Custom Strategy

Compared to the common industry scan strategy, the custom scan strategy generates a visually more homogeneous part for all three optimized conditions of the tensile bar (Fig. 16). However, the baseline condition is significantly undermelted. Visual results are supported by the distributions for each condition (Fig. 17). The standard deviations for the baseline and single parameter set were reduced. The standard deviation for the large parameter set remained the same, but the mean is closer to the desired total time molten. Therefore, the custom scan strategy generates a more homogeneous part than the common industry scan strategy. This strategy generates scan vectors that are more homogeneous by removing the outlier scan vectors that the common industry scan strategy generates. In addition to the lower standard deviations, the distributions show that the single and large parameter sets have the same mean and almost the same standard deviation. This supports the homogeneity of the custom scan strategy vectors because a single

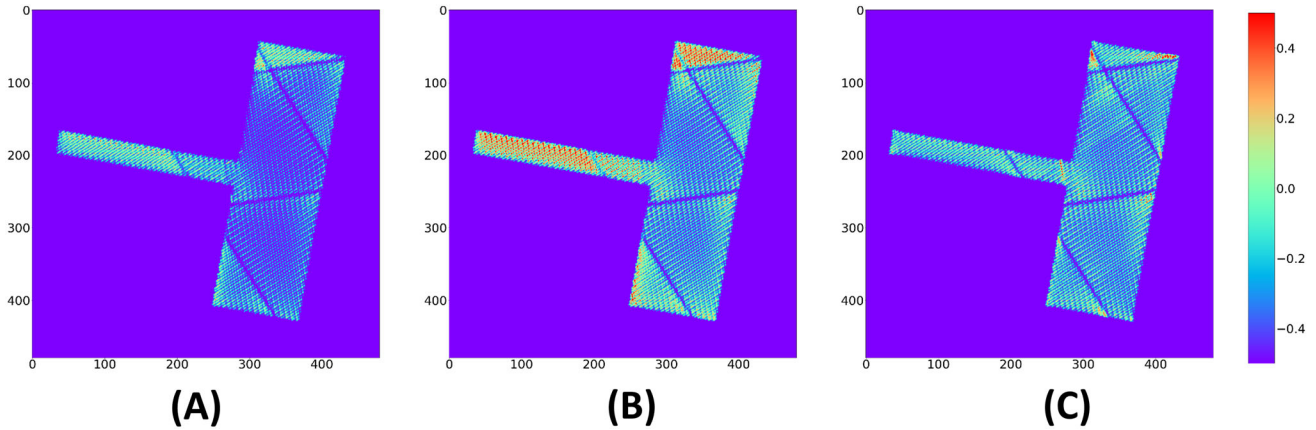


Fig. 20. Baseline and optimized T-object results obtained using the common industry scan strategy. The optimization was performed on the tensile bar and then applied to the T-object to test the generalizability of the approach. Results shown are T-object baseline (A), optimized single parameter tensile bar results applied to the T-object (B), and optimized large parameter tensile bar results applied to the T-object (C). The color bar represents the difference between the total time molten and the desired total time molten (Color figure online).

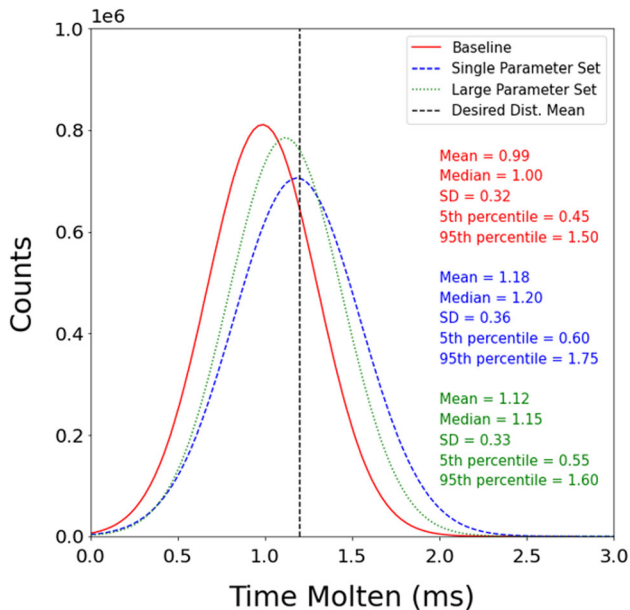


Fig. 21. Baseline and optimized T-object result distributions using the common industry scan strategy with generalization from tensile bar to the T-object.

parameter set can optimize the scan vectors nearly as well as 8000 parameter sets. The single parameter does well because the scan vectors in the custom scan strategy are more uniform in length relative to the common industry strategy.

The results of the optimized T-object using the custom scan strategy followed a similar trend seen in the tensile bar (Figs. 18 and 19). These results demonstrate that the custom scan strategy can generate homogeneous vectors on more complicated geometries.

### Generalizability of Optimization Approach

Generalizability, somewhat different from transfer learning,<sup>16,17</sup> in this application refers to using

the optimized results from one geometry (i.e., the tensile bar) and applying them to another geometry (i.e., the T-object). The optimized results should not depend on the geometry of the part but the distribution of the part's scan vectors. If scan vectors and scan vector neighborhoods of two parts are similar, the optimized parameter sets should be generalizable.

The ability to generalize is accomplished through the vector descriptor space, which can be either a single or a large parameter set. The vector descriptor space attempts to generalize the local scan paths contained in a geometry. If enough vectors with unique vector descriptors have been seen by the algorithm, any new geometry could be optimized by using previously learned results. This is not typical transfer learning referred to in machine learning, but generalization because optimized conditions are mapped into a generalizable space that can be applied to another geometry. To demonstrate the generalizability, the optimized parameters for the common industry strategy applied to the tensile bar were 'transferred' to the T-object (Figs. 20 and 21). Visually, the T-object looks similar whether it is optimized with the tensile bar results or its own results. This is true for both the single and large parameter sets, which have hot spots in similar locations. The distribution for the single parameter set in both cases is the same. The mean, median, standard deviation, 5th percentile, and 95th percentile values are all equal in the original custom industry scan strategy and in the 'transferred' case. The distribution for the large parameter set is slightly better in the 'transferred' case compared to the original case. The mean of the distribution is closer to the desired total time molten, and the standard deviation increases minimally. Consequently, this suggests that the optimized results from one geometry can be used to optimize a different geometry. Similar results were seen when testing generalizability on the custom scan strategy.

## CONCLUSIONS AND FUTURE WORK

This paper presents a methodology for homogenizing the total time molten for a LPBF part. An optimization process was outlined that changes the scan parameters of power and speed to move the total time molten for each scan vector to the desired total time molten. The best update strategy utilized a  $\tanh^2$  function to clip the step size during the scan parameter update. A smoothing method was examined to account for the influence of neighboring vectors on a particular vector. The smoothing did not have a strong influence on the distribution of total time molten for the scan vectors. However, the optimization converged faster if the smoothing parameter,  $\sigma$ , was  $> 5/3$ .

The results showed that a more homogeneous total time molten in an LPBF part geometry was achieved when optimizing the power and speed at the vector level. The optimized vector parameters were selected based on the scan vector lengths and the lengths of their neighboring vectors. In addition, once the vectors were optimized in one geometry, the optimized parameters could be applied to similar vectors in a different geometry. It was shown that a generalizable vector descriptor allowed for the transferability of the parameter sets. In general, the most homogeneous total time molten was obtained when the vectors were grouped with similar vectors in a more discrete manner in the vector descriptor space.

An area of future work includes generating and fitting a more meaningful optimization criterion. Currently, the optimization simply targets a specific total time molten for a layer based on domain knowledge, but it is not certain what this means for the properties of the part. A suggested criterion, presented in the previous paper,<sup>2</sup> would be the principal component analysis space of thermal history. This space can capture the full thermal history instead of just the total time melted, allowing for direct linking between the optimization and part properties. However, it is not currently known how updating the scan parameters for a particular vector will alter this space, and an investigation will need to take place to understand this effect. Once this effect is understood, optimization can be applied to shift the points in the PCA space to create the desired cluster(s).

## CONFLICT OF INTEREST

The authors have no conflicts of interest.

## REFERENCES

1. J.-Y. Lee, J. An, and C.K. Chua, *Appl. Mater. Today* 7, 120 (2017).
2. S. Srinivasan, B. Swick, and M.A. Groeber, *JOM* 72, 4393 (2020).
3. J. Jiang and Y. Ma, *Micromachines* 11, 633 (2020).
4. L. Englert, S. Czink, S. Dietrich, and V. Schulze, *J. Mater. Process. Technol.* 299, 117331 (2022).
5. M.J. Matthews, G. Guss, S.A. Khairallah, A.M. Rubenchik, P.J. Depond, and W.E. King, *Acta Mater.* 114, 33 (2016).
6. E.J. Schwalbach, S.P. Donegan, M.G. Chapman, K.J. Chaput, and M.A. Groeber, *Addit. Manuf.* 25, 485 (2019).
7. B. Stump, A. Plotkowski, and J. Coleman, *Model. Simul. Mater. Sci. Eng.* 29, 035001 (2021).
8. J. Lin, E. Keogh, S. Lonardi, B. Chiu, in *DMKD '03 Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (2003).
9. E. Duong, L. Masseling, C. Knaak, P. Dionne, and M. Megahed, *Addit. Manuf. Lett.* 3, 100047 (2022).
10. R.L. Iman, *Encyclopedia of Quantitative Risk Analysis and Assessment* (John Wiley & Sons, Ltd, 2008).
11. H. Yeung, B. Lane, and J. Fox, *Addit. Manuf.* 30, 100844 (2019).
12. C.L. Druzgalski, A. Ashby, G. Guss, W.E. King, T.T. Roehling, and M.J. Matthews, *Addit. Manuf.* 34, 101169 (2020).
13. T. Szandala, in *Bio-Inspired Neurocomputing*, ed. by A.K. Bhoi, P.K. Mallick, C.-M. Liu, and V.E. Balas (Springer, Singapore, 2021), pp. 203–224.
14. A.J. Smola and B. Schölkopf, *Stat. Comput.* 14, 199 (2004).
15. D. Berrar, in *Encyclopedia of Bioinformatics and Computational Biology*, ed. by S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach (Academic Press, Oxford, 2019), pp. 542–545.
16. K. Weiss, T.M. Khoshgoftaar, and D. Wang, *J. Big Data* 3, 9 (2016).
17. L. Torrey and J. Shavlik, in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, ed. by E.S. Olivas, J.D.M. Guerrero, M. Martinez-Sober, R. Magdalena-Benedito, and A.J.S. Lopez (IGI Global, Hershey, 2010), pp. 242–264.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.