2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)
August 28 - September 1, 2024. Bari, Italy

Knowledge Constrained Deep Clustering for Melt Pool Anomaly Detection in Laser Powder Bed Fusion

Erfan Ziad, Zhuo Yang, Yan Lu, Feng Ju

Abstract— **The rapid expansion of the manufacturing sector has brought laser-based metal additive manufacturing, like laser powder bed fusion, to the forefront of innovation. Yet, its widespread acceptance hinges on overcoming numerous obstacles, including uncertainties regarding part quality when employing standardized materials in additive manufacturing procedures. Clustering techniques are essential in uncovering patterns within data sets, particularly in the field of additive manufacturing, where understanding the behavior of meltpool images is crucial for process optimization. Traditional hierarchical clustering methods often lack the ability to incorporate domain-specific knowledge, limiting their effectiveness in this field. In this study, we propose a novel approach that integrates knowledge-constrained hierarchical clustering with encoded meltpool image sequences. By incorporating domain-specific constraints, our approach aims to enhance clustering accuracy and provide more interpretable cluster assignments. Specifically, our approach demonstrates improvements in clustering performance, as measured by the Calinski-Harabasz index. The base model achieved an index of 2.07, while the Constrained model attained 5.47, indicating a substantial improvement in clustering structure. This enhanced performance enables a more accurate evaluation of printed parts' quality based on melt pool image sequences.**

I. Introduction

Additive manufacturing, particularly in the context of metal 3D printing, relies on a deep understanding of the melt pool dynamics captured in image sequences [1], [2]. Hierarchical clustering techniques offer a promising avenue for analyzing these sequences, but their effectiveness can be hindered by the lack of incorporation of domain-specific knowledge. Clustering techniques play a crucial role in various domains by grouping data points based on similarities [3], [4]. In the field of additive manufacturing, anomaly detection based on clustering methods is instrumental in analyzing complex data generated from processes like laser-based metal additive manufacturing. However, traditional clustering algorithms often face challenges when applied to such data due to their high dimensionality and intricate patterns. Integrating domain knowledge into clustering algorithms has shown promise in addressing these challenges, enabling more accurate and interpretable results [5]. Moreover, the advent of neural network-based approaches has provided new avenues for enhancing clustering performance by leveraging deep learning capabilities to extract intricate features and patterns from high-dimensional data [6].

Erfan Ziad and Feng Ju are with the School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281 USA.

Zhuo Yang and Yan Lu are with Engineering Lab, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA.

Please send all correspondance to Dr. Feng Ju at fengju@asu.edu.

In recent years, paper [7] proposed a deep learning-based framework to detect anomalies and monitor the laser powder bed fusion additive manufacturing process in real-time using sequences of melt pool images. In [8], authors extracted motion features from consecutive melt pool images and performed k-means clustering to identify different melt states under varied processing parameters. Researchers proposed a method that used standard K-Means and Gaussian Mixture Model (GMM) clustering algorithms to segment the latent space into distinct clusters based on the characteristics of the input data [9]. Scholars used k-means unsupervised clustering algorithm to identify similar gradient fields in [10].

A knowledge-infused 3D U-Net model is developed using posterior regularization to incorporate separable and non-separable anatomical constraints in [11]. Also, a Posterior-Regularized Bayesian Neural Network (PR-BNN) framework for incorporating different types of soft and hard knowledge constraints was presented in [12]. Another study presents a multi-stage inference framework for modeling and predicting sequential processes using knowledge-guided Bayesian neural networks (BNNs) [13]. In [14], scholars present a Posterior Regularization (PR) framework for incorporating structural constraints into probabilistic models with latent variables.

This paper introduces a novel hierarchical clustering algorithm designed to distinguish different meltpool image sequences in laser-based metal additive manufacturing. The algorithm leverages a knowledge constraint based on the Euclidean distance between the center of meltpools and their contours, significantly enhancing clustering accuracy and anomaly detection. By integrating domain-specific knowledge, our method addresses the limitations of traditional clustering algorithms and provides more meaningful groupings of data points. Our technical contributions are as follows:

1) Development of a novel custom hierarchical clustering algorithm that integrates domain-specific knowledge to improve clustering accuracy.

2) Demonstrate the effectiveness of our approach through real-world data from laser powder bed fusion additive manufacturing processes.

The rest of this paper is organized as follows: Section II presents a description of the data. Section III discusses the detailed data processing steps. The developed methodologies are given in Section IV. Further, in the next section, the numerical results are discussed. Finally, Section VI concludes this paper with a key findings summary and future work.

II. Data Description

An additive manufacturing process monitoring dataset captured on the Additive Manufacturing Metrology Testbed (AMMT) at the National Institute of Standards and Technology (NIST) is used for this work. The dataset provides comprehensive in-situ process monitoring data for additive manufacturing, particularly focusing on the melt pool images. The images from the melt pool monitoring are saved in two different file formats: AVI video files and TIF image stack files. The frame rate of the video is 10,000 Hz, meaning each TIF image represents 0.1 ms of the corresponding video. The images are nominally oriented and aligned with the machine coordinate system, eliminating the need for transformations. The dataset facilitates various analyses, including general process monitoring, sensor fusion, part-to-part comparison, and in-situ process signature to ex-situ part quality comparison. Additionally, it provides the foundation for developing data-driven predictive models and further ex-situ characterization of the fabricated parts [15].

III. Data Processing

In the initial step, the frames extracted from the videos undergo preprocessing. Min-max normalization, as depicted in Equation 1, is applied to standardize the pixel values of the frames/images between zero and one. Here, x^* denotes the normalized pixel value, and x represents the original value. Furthermore, to mitigate the impact of the scanning pattern and bolster the analysis's robustness, a threshold is meticulously chosen to binarize the images. This step aims to alleviate the appearance of 'tails' in the meltpool images.

$$x_i^* = \frac{x_i - \min(x_1, x_2, ..., x_n)}{\max(x_1, x_2, ..., x_n) - \min(x_1, x_2, ..., x_n)} \tag{1}$$

Considering the use of a co-axial camera for image capture, it's anticipated that the center of each meltpool aligns with the image's center in terms of both width and height. However, slight deviations from this expectation are observed. To pinpoint the precise center point, various settings are tested to determine the optimal configuration that consistently represents the fixed center point across all images and scanning patterns.

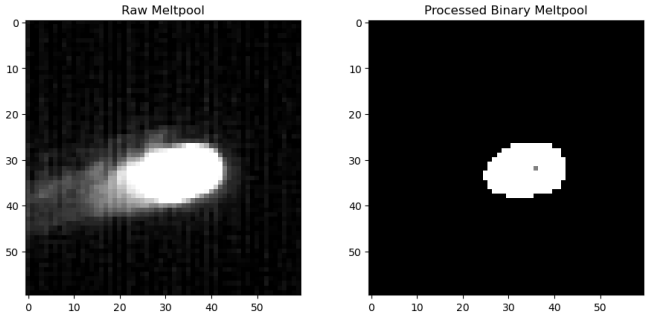
The outcome of these preprocessing stages is illustrated in Figure 1. To incorporate the temporal dimension of the data, image sequences comprising 10 frames are created from the videos. Subsequently, the dataset is partitioned into a 75% training set (x_train) and a 25% testing set (x_test). This systematic pipeline facilitates the efficient preparation of sequential image data for subsequent modeling endeavors.

IV. Methodology

A. Knowledge Constraint

For each processed meltpool image, a distance distribution is computed using the Euclidean distance metric, as outlined in Equation 2. This metric serves as the knowledge constraint, which is subsequently integrated into the model.

To derive the distance distribution, we first establish a fixed center point (x_{center}, y_{center}) as the reference for distance



(a) Raw Image (b) Processed Image and Center

Fig. 1: Meltpool Images Processing

calculations, as described in the preceding section. Next, to systematically cover the entire image, we divide the full 360 degrees into 36 ten-degree angles, effectively creating 36 rays or directions emanating from the fixed center point. Along each direction, we calculate the Euclidean distance between the fixed center point and the farthest pixel or point, denoted as (x, y). This process is repeated for all directions in each image.

For clarity in understanding, Figure 2 visually presents this methodology, utilizing only four angles instead of the complete set of 36. In the visualization, white lines depict the directions corresponding to the selected angles, while gray regions represent the meltpool.

$$d = \sqrt{(x - x_{\text{center}})^2 + (y - y_{\text{center}})^2} \tag{2}$$

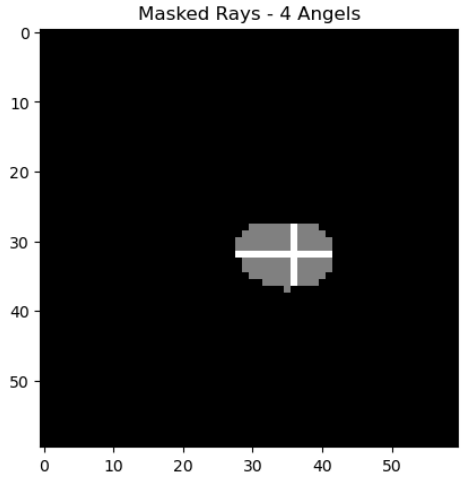


Fig. 2: Distance from the center to the contour of meltpool

We can then find the distance distribution for each image and observe the normal melt pool behavior as opposed to the abnormal ones as in Figure 3. The horizontal axis shows various angles, and the vertical axis shows the Euclidean distance. It's worth noting that since we have different scanning patterns and the shape of the meltpool can vary, the overall shape of the distribution curve might appear to shift on the horizontal axis. To make accurate comparisons, all of the curves are aligned so that the peak (maximum distance),

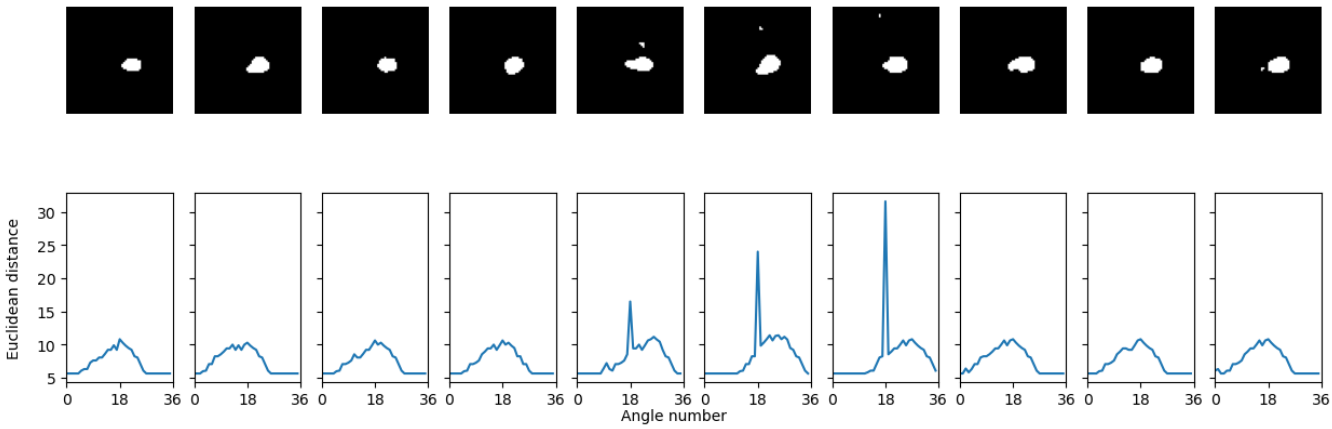


Fig. 3: Distance Distribution of Ten Sample Images

regardless of the scanning pattern, is in the middle of the horizontal axis (angles axis). This shift is reasonable since we are in a polar coordinate system, and we can consider the sequence of distances as a circular sequence. This allows for a more accurate comparison among the different meltpool images. It can be concluded from Figure 3 that normal melt pool distributions look like a bell curve while the abnormal ones look different, for example, by having multiple peaks and higher maximum distance in general. Overall, this approach provides a systematic method for analyzing the spatial distribution of pixels within the meltpool images, which can be valuable for understanding the meltpool shape. Previously, researchers used metrics such as the size or area of the melt pools to describe the morphological features.

B. Auto-encoder

Autoencoders are neural networks with two primary parts: encoders and decoders. The encoder processes the input data into a different representation, while the decoder converts the new representation back into the original format. The main objective of autoencoders is to retain as much information as possible during this process while ensuring that the new representation exhibits certain desired properties [16]. By training the network in this way, autoencoders can be used for tasks like data compression, noise reduction, and image generation [17]. An autoencoder tailored for processing sequential image data is used, with the same architecture as in [7]. The training data (x_train) consists of sequences of raw meltpool images. By utilizing raw images without any preprocessing, we grant the neural network (NN) access to the complete range of visual information, potentially resulting in improved performance in detecting and analyzing melt pools. The encoder initiates with a series of convolutional layers, each followed by a normalization layer to enhance the stability and robustness of the network. Specifically, the initial layers consist of a 16-filter convolution with a 5x5 kernel, followed by an 8-filter convolution with a 3x3 kernel, and another 8-filter convolution with a 2x2 kernel. Each convolutional layer uses the ReLU activation function and appropriate strides to reduce dimensionality while maintaining critical spatial information.

Following the convolutional layers, the model incorporates ConvLSTM2D layers to capture the spatio-temporal dependencies within the data. These layers sequentially have 64, 32, and 16 filters, respectively, each with a 2x2 kernel and ReLU activation. Layer normalization is again applied after each ConvLSTM2D layer to stabilize the learning process.

The decoder mirrors the encoder's structure, reversing the encoding process to reconstruct the input data from its compressed representation. It begins with Conv2DTranspose layers, featuring 8 filters with a 2x2 kernel, followed by another 8 filters with a 3x3 kernel, and concludes with 16 filters with a 5x5 kernel. The final layer uses a convolution operation with a 5x5 kernel and a sigmoid activation function to reconstruct the original spatial dimensions of the input data. Figure 5 depicts the architecture of our neural network.

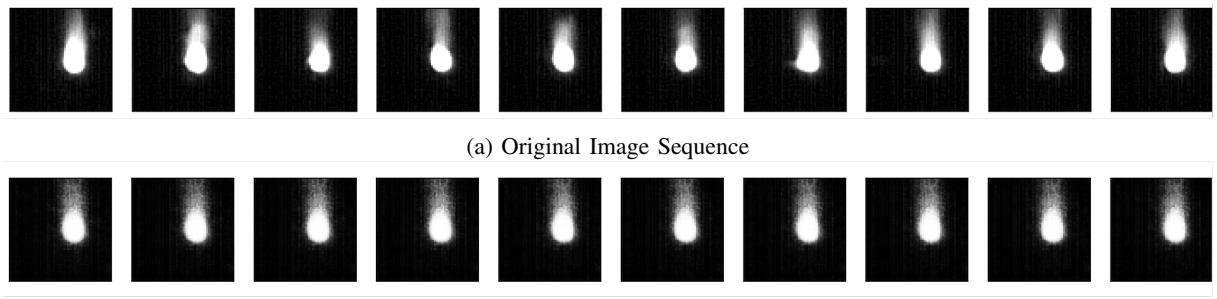
The training procedure employs the Adadelta optimizer and binary cross-entropy loss function, optimized for deep neural networks. An early stopping mechanism is implemented to monitor the validation loss, halting training if there is no improvement over 10 consecutive epochs. This prevents overfitting and ensures the model's ability to generalize to unseen data. The model is trained for up to 300 epochs with a batch size of 32, and the training data is shuffled before each epoch to mitigate the risk of the model memorizing the sequence of examples. Validation is conducted using a separate dataset (x_test) to evaluate the model's performance throughout the training process.

Overall, this autoencoder architecture effectively captures both spatial and temporal features from sequences of images, enabling robust encoding and decoding processes. Figure 4 shows a sample sequence and its decoded sequence.

C. Clustering

In this section, we describe a novel methodology called a constrained hierarchical clustering algorithm designed for the analysis of image sequence data. The algorithm integrates considerations of variance and distributional divergence to guide the clustering process.

The algorithm is initialized with two essential components: encoded melt pool images and the distance distribution functions associated with each image sequence. During the



(a) Original Image Sequence

(b) Decoded Image Sequence

Fig. 4: Auto-encoder Prediction

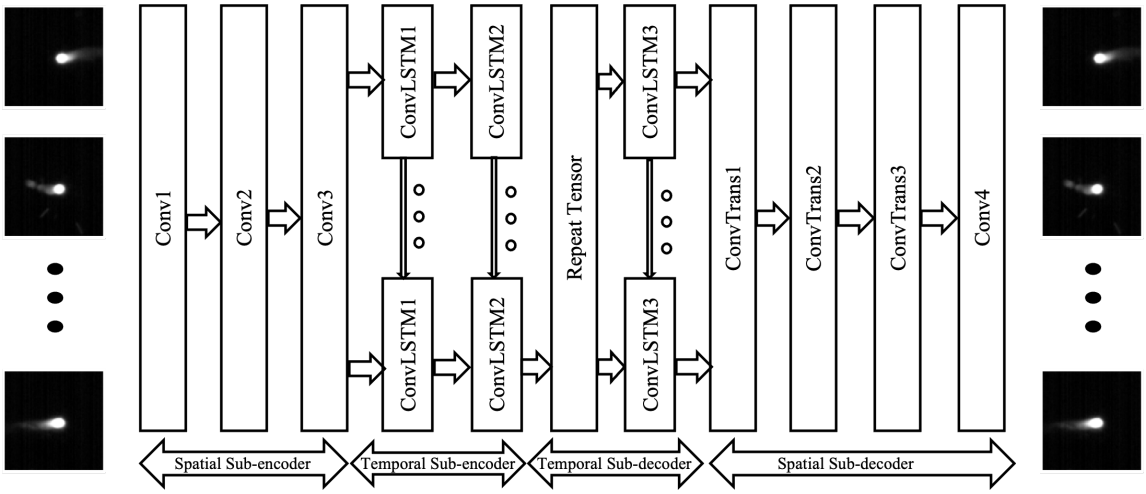


Fig. 5: Neural Network Architecture

clustering process, the algorithm calculates the variance for each cluster at every merge step. This calculation involves computing the variance of the merged clusters and comparing it with the individual variances of the constituent clusters. In this approach, we compute the variance of pixel values across all images, treating them collectively as a single data point. This algorithm tries to merge clusters to minimize within-cluster variance. By quantifying the increase in variance resulting from cluster merging, the algorithm assesses the stability and coherence of the resulting clusters.

In addition to the variance term, the algorithm incorporates a penalty term based on the divergence of distance distributions to ensure the similarity of distributional characteristics among merged clusters. The Jensen-Shannon divergence metric is employed to measure the divergence between the distance distributions of cluster data points. This penalty term encourages the formation of cohesive and homogeneous clusters by penalizing significant differences in distributional profiles. This metric has advantages over the Kullback-Leibler divergence [18]. Firstly, a symmetric metric is needed since there is no difference between the two candidate clusters. Secondly, in some cases, the traditional Kullback-Leibler divergence results in an infinity value due to zero probabilities in the distribution. The Jensen-Shannon divergence is symmetric and results in finite values for the divergence between the distributions [19]. Kullback-Leibler divergence is shown in Equation 3 where P_i and Q_i represent the probabilities of the i-th bin in the histograms of distributions P and Q, respectively. Also, the Jensen-Shannon divergence metric can be seen in Equation 4.

$$D_{KL}(P\|Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{3}$$

$$D_{JS}(P\|Q) = \frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(Q\|M) \tag{4}$$

where $D_{JS}(P\|Q)$ is the Jensen-Shannon divergence between probability distributions P and Q, $D_{KL}(P\|M)$ is the Kullback-Leibler divergence between P and M = $\frac{1}{2}(P+Q)$, and $D_{KL}(Q\|M)$ is the Kullback-Leibler divergence between Q and the average distribution M. To obtain P and Q distributions, the average distance distributions of all the meltpool images in the candidate cluster are taken, and then its histogram is calculated for use in penalty calculations.

Lastly, the total cost function combines the contributions from variance increase and distributional divergence penalty to determine the overall cost associated with merging the two candidate clusters. The algorithm iteratively evaluates potential merge operations and selects the merge that minimizes the total cost, thereby guiding the hierarchical clustering process. Since the increase in variance and penalty term have different orders of magnitude, in our case, the penalty term usually dominates the variance term. The solution to this problem is to normalize both terms in order to leverage both terms together. Our normalization technique

is once again min-max normalization as shown in Equation 1, which results in numbers ranging from zero to one for both the penalty and the variance. x^* is the normalized penalty/variance and x is the original value of it in this case.

At each iteration, the algorithm identifies the pair of clusters with the lowest total cost and merges them into a single cluster. This merging operation involves combining the data points from the selected clusters to form a new cluster. The process continues iteratively until the predefined number of clusters (num_clusters) is achieved as the stopping condition. This algorithm can be seen in Algorithm 1. It's important to note that clustering algorithms usually aim to create clusters that are internally compact (with low within-cluster variance) while being well-separated from each other (with high between-cluster variance). By doing experiments with different coefficients for between-cluster variance terms in the total cost, it is observed that this term misguides the clustering algorithm and results in worse performance evaluation metrics. As a result, the between-cluster dispersion is not considered in the algorithm.

Algorithm 1: Custom Hierarchical Clustering Algorithm

Data: Input data X representing the encoded image sequences
Data: Distance Distribution functions Y corresponding to each data point in X

Function `Clustering`(*num_clusters*) **:**
  Initialize clusters as individual data points
  while *the number of clusters is greater than num_clusters* **do**
    Compute the total cost of merging all pairs of clusters by adding up contributions from variance increase and distributional divergence penalty
    Identify the pair of clusters with the lowest total cost
    Merge the selected clusters into a single cluster
    Update the clusters list accordingly
  end
Result: Final clusters

V. Results

Two variations of the custom hierarchical clustering were compared. The first model only incorporated the variance term, while the second model incorporated both the variance and the knowledge constraint. Both models were executed until the total number of clusters reached 12 on 100 sample image sequences. The number of clusters chosen during the clustering process can have a significant impact on the results obtained. It's important to select an appropriate value to prevent issues like underfitting or overfitting. Striking a balance to capture the underlying structure of the data without introducing unnecessary complexity or loss of information is crucial. Using the total number of data points we had, we tested different cluster sizes and found that 12 worked well. The results of the clustering are shown in Figure 7. For both models, the larger clusters mainly consisted of normal sequences, while the smaller ones mainly showed abnormalities. The Calinski-Harabasz Index (CHI), also known as the variance ratio criterion, is utilized to compare the performance of the two clustering algorithms [20]. CHI is a relative measure for comparing different clustering solutions on the same dataset. Higher values of the index generally indicate better clustering structures, but the magnitude of the index does not have a specific quantitative interpretation beyond that. The base model achieved an index of 2.07 while the Constrained model got 5.47, implying the better performance of the second model. Equation 5 shows the equation for CHI.

$$\mathrm{CHI} = \frac{B}{W} \times \frac{N-k}{k-1} \tag{5}$$

Where CHI is the Calinski-Harabasz Index, B is the between-cluster dispersion, W is the within-cluster dispersion, N is the total number of data points, and k is the number of clusters.

For example, as shown in Figure 6, these two sequences are misclassified in the base model into the large normal sequence clusters, while the constrained clustering model accurately clustered them together into a separate cluster from the rest. Overall, the model with constraints is more accurate in separating normal sequences from abnormal sequences, proving the value of embedding knowledge into models.

In the absence of definitive ground truth labels for meltpool images due to their high-frequency nature, evaluating clustering methods relied on visual inspection and quantitative comparison using the Calinski-Harabasz Index (CHI). The effectiveness of the clustering results is assessed subjectively, primarily based on the visual coherence of clusters. However, to ensure a more rigorous evaluation, we also employed the CHI score, where higher values indicate better clustering structures. By clustering structure, we refer to the formation of compact and well-separated clusters.

The use of a constrained model proved beneficial in this scenario due to its flexibility and ability to adapt to different types of knowledge. The model's versatility allows it to incorporate domain-specific knowledge, improving its effectiveness. Although the constrained model showed improved CHI scores, indicating better clustering structures, it's important to note that the direct connection between meltpool image clusters and part quality is somewhat indirect and open to interpretation. Therefore, despite the promising performance of the constrained model in classifying meltpool sequences, further validation against quality measures will be necessary to confirm the model's practical usefulness. Finally, regarding the time and space complexity of the algorithm, it was observed that on an average personal computer, the constrained model takes approximately 1 minute to execute and uses about 170 MB of memory, which could be improved through further refinements.

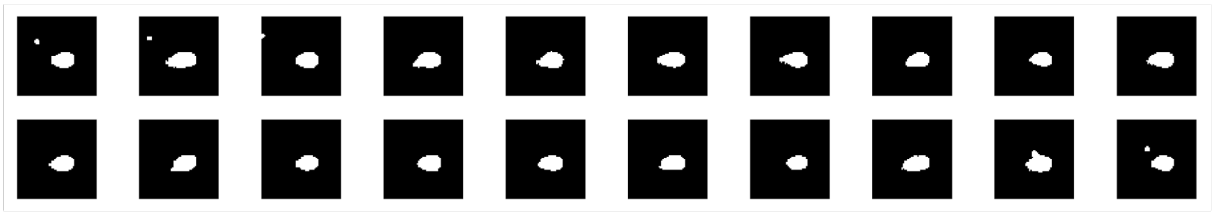
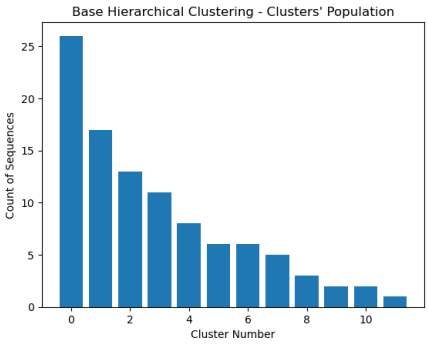
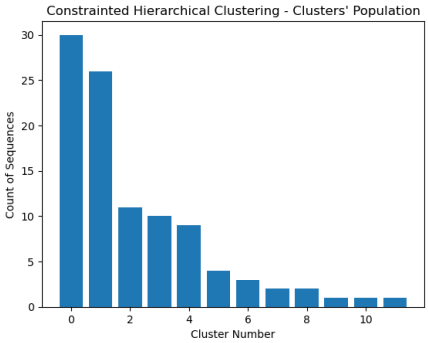


Fig. 6: Two Sample misclassified sequences using the base model



(a) Base Clustering



(b) Constrained Clustering

Fig. 7: Clusters' Population - Number of Clusters=12

VI. Conclusion

A hierarchical clustering technique incorporating knowledge constraints improved clustering accuracy on encoded image sequences in this study. This approach not only facilitates a more meaningful interpretation of the resulting clusters but also contributes to the overall robustness and reliability of the clustering outcomes. The findings highlight the importance of incorporating domain expertise as constraints into clustering methodologies, offering valuable insights for enhancing the analysis and interpretation of complex data sets in various domains. In future work, we could investigate the impact of meltpool images on final part quality by analyzing ex-situ monitoring data such as XCT image datasets. This will help us understand how different meltpool shapes affect the quality and structure of the final part. Furthermore, we need to improve the algorithm's scalability. One way to achieve this could be to incorporate knowledge via posterior regularization into the neural network framework.

References

[1] S. Fathizadan, F. Ju, and Y. Lu, "Deep representation learning for process variation management in laser powder bed fusion," *Additive Manufacturing*, vol. 42, p. 101961, 2021.

[2] Z. Lotfizarei, A. Mostafapour, A. Barari, A. Jalili, and A. E. Patterson, "Overview of debinding methods for parts manufactured using powder material extrusion," *Additive Manufacturing*, vol. 61, p. 103335, 2023.

[3] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[4] F. Nielsen, *Hierarchical Clustering*, pp. 195–211. 02 2016.

[5] B. Vogel-Heuser, F. Ju, C. Fantuzzi, Y. Lu, and D. Hess, "Knowledge-based automation for smart manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 2–4, 2021.

[6] V. Azimirad, S. V. Sotubadi, and A. Nasirlou, "Vision-based learning: a novel machine learning method based on convolutional neural networks and spiking neural networks," in *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pp. 192–197, IEEE, 2021.

[7] S. Fathizadan, F. Ju, Y. Lu, and Z. Yang, "Deep spatio-temporal anomaly detection in laser powder bed fusion," *IEEE Transactions on Automation Science and Engineering*, 2023.

[8] X. Lin, Q. Wang, J. Y. H. Fuh, and K. Zhu, "Motion feature based melt pool monitoring for selective laser melting process," *Journal of Materials Processing Technology*, vol. 303, p. 117523, 2022.

[9] X. Zhao, A. Imandoust, M. Khanzadeh, F. Imani, and L. Bian, "Automated anomaly detection of laser-based additive manufacturing using melt pool sparse representation and unsupervised learning," in *2021 International Solid Freeform Fabrication Symposium*, University of Texas at Austin, 2021.

[10] L. Scime and J. Beuth, "Using machine learning to identify in-situ melt pool signatures indicative of flaw formation in a laser powder bed fusion additive manufacturing process," *Additive Manufacturing*, vol. 25, pp. 151–165, 2019.

[11] J. Huang, H. Yan, J. Li, H. M. Stewart, and F. Setzer, "Combining anatomical constraints and deep learning for 3-d cbct dental image multi-label segmentation," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 2750–2755, IEEE, 2021.

[12] J. Huang, Y. Pang, Y. Liu, and H. Yan, "Posterior regularized bayesian neural network incorporating soft and hard knowledge constraints," *Knowledge-Based Systems*, vol. 259, p. 110043, 2023.

[13] S. Chung, C.-H. Chou, X. Fang, R. Al Kontar, and C. Okwudire, "A multi-stage approach for knowledge-guided predictions with application to additive manufacturing," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1675–1687, 2022.

[14] K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar, "Posterior regularization for structured latent variable models," *The Journal of Machine Learning Research*, vol. 11, pp. 2001–2049, 2010.

[15] B. Lane and H. Yeung, "Process monitoring dataset from the additive manufacturing metrology testbed (ammt): Overhang part x4," *Journal of research of the National Institute of Standards and Technology*, vol. 125, pp. 1–18, 2020.

[16] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[18] I. Csiszár, "I-divergence geometry of probability distributions and minimization problems," *The annals of probability*, pp. 146–158, 1975.

[19] M. Menéndez, J. Pardo, L. Pardo, and M. Pardo, "The jensen-shannon divergence," *Journal of the Franklin Institute*, vol. 334, no. 2, pp. 307–318, 1997.

[20] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.