Solid Freeform Fabrication 2022: Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

A Data-Driven Approach for Multi-Lattice Transitions

Martha Baldwin,[1] Nicholas A. Meisel,[2] Christopher McComb[1]

[1]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213
[2]School of Engineering Design, Technology, and Professional Programs, The Pennsylvania State University, University Park, PA 16802

Abstract

Additive manufacturing is advantageous for producing lightweight components while maintaining function and form. This ability has been bolstered by the introduction of unit lattice cells and the gradation of those cells. In cases where loading varies throughout a part, it may be necessary to use multiple lattice cell types, also known as multi-lattice structures. In such structures, abrupt transitions between geometries may cause stress concentrations, making the boundary a primary failure point; thus, transition regions should be created between each lattice cell type. Although computational approaches have been proposed, smooth transition regions are still difficult to intuit and design, especially between lattices of drastically different geometries. This work demonstrates and assesses a method for using variational autoencoders to automate the creation of transitional lattice cells. In particular, the work focuses on identifying the relationships that exist within the latent space produced by the variational autoencoder. Through computational experimentation, it was found that the smoothness of transition regions was higher when the endpoints were located closer together in the latent space.

1. Introduction

Additive manufacturing (AM) is renowned for producing lightweight components while addressing complex geometry and strength requirements. This ability has been bolstered by the introduction of unit lattice cells, which have enabled designers to significantly reduce the weight of a part, while maintaining needed stiffness and geometry[1,2]. Proposed applications of lattices are often centered around light weighting or improving the stiffness to weight ratios of components manufactured [1]. Lattices are also discussed as impact reduction applications, such as the crumple zone in a car [1] or surgical implants that experience variable loading [3]. Recent work has indicated that graded lattice structures are the next step for improving upon uniform lattice structures [4–8]. These structures vary the thickness of individual strut-based unit cells to accommodate areas of high or low stress. Graded lattice structures have been shown to significantly outperform the stiffness of uniform lattice structures of comparable weight [5,6,8]. Given the unique deformation properties of graded lattices, it is believed that they would be more beneficial than uniform lattices in impact reduction applications [9]. Further research has explored the possible improvements afforded by multi-lattice structures, which are structures comprised of multiple types of unit cell topologies [3,10–13]. Similar to graded lattice structures, multi-lattice structures have higher strength and stiffness than uniform lattice structures of comparable density[9,12]. However, in multi-lattice structures, abrupt transitions between lattice types may cause stress concentrations, which are detrimental to overall part strength [12]. One method to address abrupt transition regions is to interpolate between the faces of two lattice cell types [10]. This creates a smooth transition which evenly distributes the stresses throughout the structure. However, smooth transition regions are currently difficult to intuit and design, especially between lattices of drastically different geometries.

Recent work has utilized machine learning to create interpolations between unit cells by optimizing the properties of the cells [14,15]. While optimization-based methods are powerful, they leave little space for input from the user. This work explores the effectiveness of convolutional Variational Autoencoders (VAEs) for automating the creation of transition regions in multi-lattice structures. Specifically, we design a VAE system that can predict transitions between different unit lattice cells and produce transition regions with any length. Our primary research questions address the unknowns about the effects of dimensionality reduction with respect to lattice topologies:

1. How does distance in the latent space and transition length affect the smoothness of interpolations?
2. How do differences in unit cell topology affect smoothness?

2. Background

2.1 Unit Cells for Design

This section defines lattice unit cells and their usefulness to the research and design community, as well as establishing the motivation for using multi-lattice structures. In this paper, the focus will be on periodic lattice structures, which are created by repeating a unit cell. A *uniform lattice* structure consists of one type of unit cell, where all the cells in the structure have the same density and size. The stress strain curve of uniform lattices matches standard plastic deformation, where there is an elastic region, followed by a plastic plateau, until the object reaches full densification [7]. The properties of each type of unit lattice are unique [1], making it important to select the correct type of lattice for a design. A *graded lattice* structure consists of one type of unit cell, but the cells in the structure can vary in volume fraction [6]. It has been proven that functionally graded lattice structures perform better with respect to stiffness [5,6,8]. The benefits of graded structures are the additional design freedom that can be used to customize failure modes and stress-strain responses [7,8]. It was suggested that such properties would be beneficial in applications where there are dynamic loads, such as surgical implants [3] or impact protection equipment [9]. These structures are also relatively simple to create. Li et al. created graded structures by using relative density mapping from a topology optimization and assigning densities based on stress requirements [4]. However, the key restriction in graded lattice structures is their property dependence on the type of unit cell. It was found that surface-based unit cells outperformed strut-based cells due to their connectivity[8], as well as the degree of gradation greatly affecting the cumulative energy absorption for body-centered cubic lattices, but not Schwarz-P lattices [2].

The term "*multi*-lattice" structure is being defined as a structure that contains multiple types of unit cells. This term is being proposed in this work to differentiate from the other types of mesostructures that are possible. The motivation for this work revolves around the implementation of multi-lattice structures where the transitions between unit cells are continuous. Many studies that explore multi-lattice structures have utilized unit cells with matching boundaries to avoid developing a method to connect nodes [3,9,12,13]. Although this is not an area of interest, it is important for understanding the behavior of multi-lattice structures on a rudimentary level. To create the structures, a relative density mapping from a topology optimization was executed to assign 2.5-dimensional unit cells based on the stresses in the system [9,12,13]. As a result, they were able to prove that trusses utilizing 2.5-dimensional multi-lattice structures outperform uniform lattice structures regarding stiffness and strength[9,12,13]. Another study was conducted by Gok that proved that using a multi-lattice hip implant reduced the maximum stress and weight

compared to conventional hip implants [3]. Although the work does not provide comparison between other types of mesostructured patterning, it does demonstrate a strong use case for multi-lattice structures. It should be noted that the study was also restricted to unit cells that had overlapping boundaries, so the transition regions were inherently smooth.

Although multi-lattice structures are more beneficial, there are some potential drawbacks. Kang et al. concluded that the boundaries between unit cell types caused stress concentrations [12]. Which indicates that there is a need for smooth transitions between unit cells to appropriately distribute stress through structures. Sanders et al. created transition regions by using signed distance functions with respect to the boundaries of each strut [10]. By combining transition regions with functional grading, they are able to prove the effectiveness of multi-lattice structures. Their work studies topology optimization of multi-lattice structures that use functionally graded interpolated transition regions. Although this technique can be applied to "unit cells composed of noncylindrical bars or plates", the interpolations between each pair of unit cells must be computed individually. For those looking to optimize both the macrostructure and the mesostructure, this method is extremely repetitive and tedious. Wang et al. have also regarded the creation of transition regions as "a challenging problem involving complex inverse design at the microscale, costly nested optimization at the macroscale, and boundary matching between neighboring microstructures [14]." Which is why more recent research has explored how to create smooth transitions between unit cells using machine learning methods [10].

2.2 Machine Learning Methods

Recent approaches to creating continuous transitions in multi-lattice structures have utilized machine learning. This section establishes the two notable machine learning methods which have been used for creating multi-lattice structures: Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANS).

Generative Adversarial Networks

Generative Adversarial Networks (GANs) utilize two models that work in parallel to ultimately create a single model for generating data [16]. These two models are known as the generator and discriminator, which work against one another in effort to beat the other. The generator produces fake data that is learned from real data, while the discriminator learns to sort the fake and real data. As training progresses, each model progressively becomes better at its purpose until the discriminator is unable to distinguish between real and fake data produced by the generator.

Wang et al. used an Inverse Homogenization GAN to generate a multi-lattice structure that reduced stress concentrations by nearly 80% [15]. This method used a topology optimization of the design space to create a relative density mapping, while considering the type of unit cells being used. This serves as the main motivation for this work, as it proves there is a benefit to creating multi-lattice structures. The advantage to using a GAN, was that it could restrict designs based on manufacturability [15].

However, this work was limited by the type of unit cells that could be interpolated. The focus was on triply periodic minimal surfaces, as their physical characteristics could be described by 6 principal components. Indicating that only structures generated using functions would be suitable for this method, which limits the freedom for using personalized unit cells. The primary

drawback to this method was that it was only intended to organize the types of unit cells, but only based on performance rather than shape. Therefore, interfaces between the cells were not perfect, and a second interpolation had to be performed to create smooth transition regions. Indicating that the IH-GAN was not ideal for creating interpolations, but better suited for optimization of unit cell types.

Variational Autoencoders

A variational autoencoder (VAE) is a machine learning model that learns how to perform data-driven dimensionality reduction [17]. Reducing the dimensionality of data can make it more computationally efficient to perform inferences, such as interpolations. What makes VAEs unique from other techniques is that they perform a non-linear dimensionality reduction, which is advantageous for reducing complex data types [18]. The reduced data is combined to establish the reduced dimensionality latent space, which can be used to represent complex mechanical parts [19] and performance characteristics of engineered systems [20].

A VAE consists of two models, an encoder, and a decoder, which work in series. Where the encoder acts as a recognition model and the decoder as the generative model [16,18]. The encoder performs a dimensionality reduction which creates a latent representation of the data. The decoder uses the information from the latent space to reconstruct the data. Finally, the VAE is trained by minimizing the error between the decoder model and the original data provided.

Wang et al. constructed a VAE to perform multi-lattice interpolations between 2D lattices [14]. This work created interpolations by traveling along the shortest paths of trained points in the latent space. Another application of VAEs in AM has been describing the properties of randomly generated unit cells [18]. Xue et al. used a principal component analysis (PCA) reduction of their generated 2D unit cells to classify them based on physical performance [18]. Their work proved that location in the latent space correlated with young's modulus, and shear modulus. Given the nature of lattice topologies, VAEs are a good candidate for dimensionality reduction of these structures. Wang et al. identify 3 benefits to the latent space created by VAEs [14]: interpolations can be created by traveling in the latent space, shape similarity can be evaluated based on distance in the latent space, and the latent space can be used to map the physical properties of the unit cells.

In contrast to existing work, our focus is to establish a methodology that permits a designer-centered approach rather than a fully automated VAE optimization. Embracing a human-in-the-loop paradigm permits the designer to inject stylistic elements as well as easily account for aspects that would be difficult to define numerically. This interaction will currently consist of selecting the endpoint unit cells and the length of the transition region.

3. Methodology

In this section, the process of data generation, development of the VAE model, and execution and evaluation of interpolations are detailed. The current work specifically deals with interpolation of 2-dimensional lattice unit cells in order to simplify the data needs and training time of the VAE model. The methodological approach to testing the hypotheses of our research questions is visualized in Figure 1. Section 1 will describe how artificial data was generated to use for testing. Section 2 outlines the hyperparameters and key characteristics of the VAE used for

training. Section 3 shows the process for creating interpolations in the latent space. Section 4 will develop the smoothness metric to measure the performance of the interpolations.

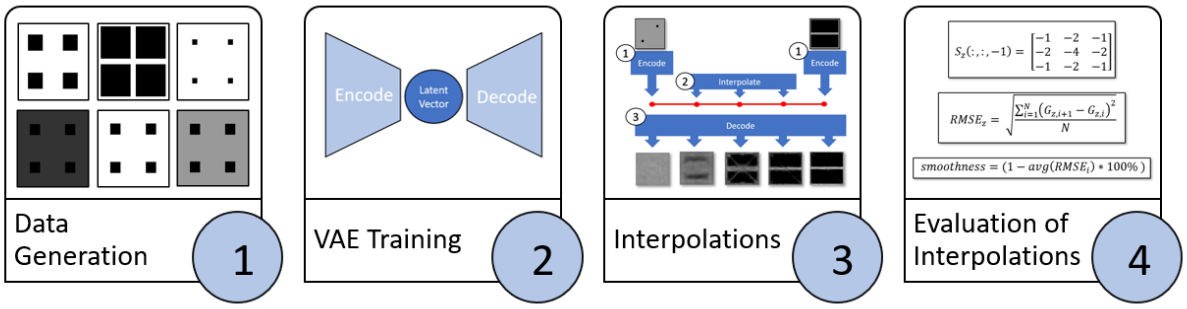


Figure 1: Illustration of overall methodologies

3.1 Synthetic Data Generation

To begin training the VAE, a significant amount of data was necessary to produce a sufficiently dense latent space for interpolation. Synthetic data was generated based on a series of 12 shapes (Figure 2) that mimicked other additive manufacturing literature[9,13]. Although the dataset does not contain a wide variety of shapes, it will allow for thorough testing of the latent space itself, as our focus is on how the quality of interpolations change within the latent space.

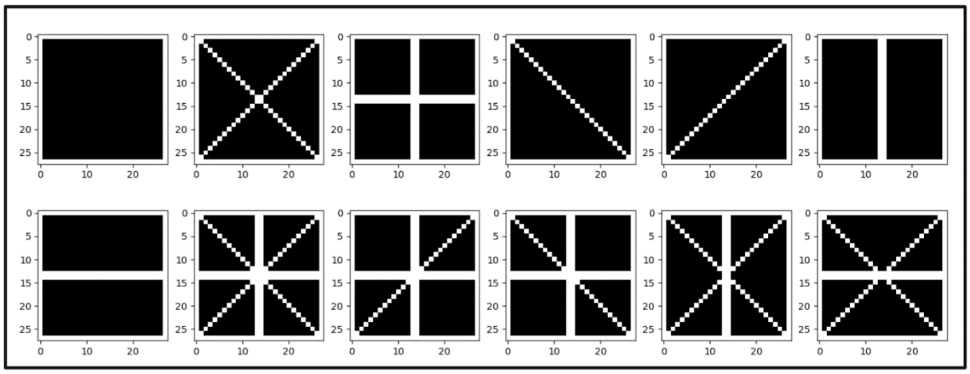


Figure 2: Synthetic Shape Types

To add variation to the data, the thickness of the shapes and densities were varied. Where the thickness of the shapes represents the increasing size of the struts. This was done stepwise by expanding upon the existing shapes (See top of Figure 3). The density represents the value of each pixel between 0 and 1, which are black and white respectively. This representation of density is indicative of the likelihood of a pixel being present (See bottom of Figure 3). Once all the shapes were generated, there were a total of 415 data points. Although limited, this dataset proved sufficient for the purposes of this paper.

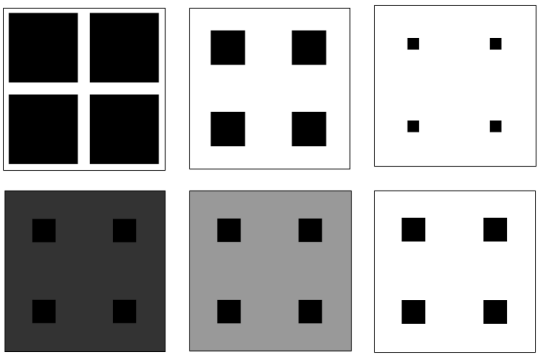


Figure 3: Original shape with possible density and pixel intensity variations

3.2 Training the Variational Autoencoder

This section outlines how the VAE was designed for this research. When the VAE was created, there were many factors that had to be considered, as there are many hyperparameters of VAEs that affect their performance. First, the overall framework of the VAE affects the performance and outputs from the system (See Figure 4). The framework shown was designed to mimic the VAEs used to organize images. Another important hyperparameter is the latent dimensionality. In this case, the latent dimensionality was chosen based on the qualitative difference between the original and decoded unit cells. A qualitative study of latent space dimensionality revealed that a dimensionality of 4 provided a desirable balance between training time and reconstruction accuracy.

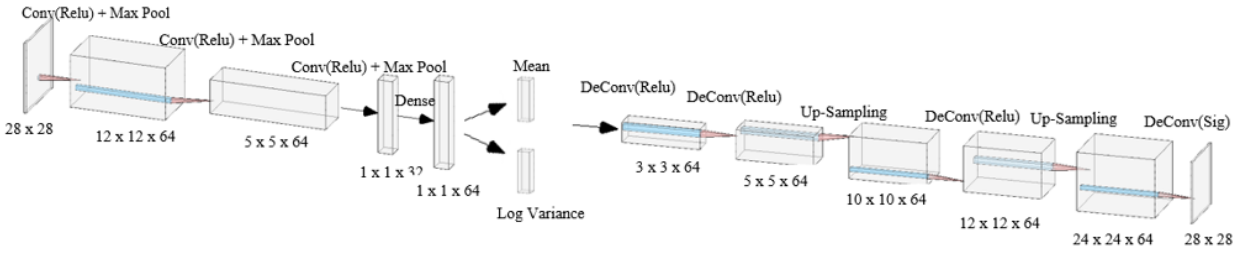


Figure 4: Framework of VAE[1]

There are many additional features that can be incorporated into a VAE to improve performance. In this case, training was terminated early if the loss failed to improve after several epochs and the autoencoder was also designed to save the weights from the best iteration. The system used a batch size of 32 to and the adam optimizer. Given this framework, the VAE used 85% of the dataset to train and 15% to validate and test the performance of the VAE.

3.3 Creating Interpolations

Once trained, the autoencoder has established a latent space in which interpolations can be performed. The latent space represents a combination of the dimensionally reduced training data. The encoder serves to recognize the unit cell and assign it a predicted latent point, and the decoder then recreates the unit cell using the predicted latent point. These functions would be used for creating interpolations between the different unit cells. As mentioned previously, interpolations require an encoding and decoding to produce desired results. The unit cells at the end of the desired

[1] Generated with this site: https://dropsofai.com/variational-autoencoders-and-image-generation-with-keras/

transition region are first provided to the encoder, which calculates the latent points of the two cells. The transitions points are then interpolated between the encoded endpoints in the latent space. Finally, the decoder generates the cells that correspond to those latent points (See Figure 5).

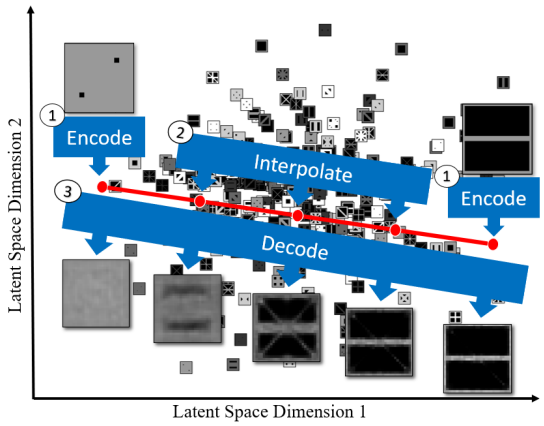


Figure 5: Diagram demonstrating interpolation process

3.4 Evaluating Smoothness

To accurately gauge the quality of an interpolation, a metric was designed to measure the smoothness of the transition region, where smoothness needed to (1) measure how the interpolation changed both within images and across images and (2) penalize pixels disconnected from the main structure, as we did not want to encourage the generation of floating pixels. Finally, (3) the metric should account for pixel intensity, allowing it to account for the gradation of pixels, as interpolations will often produce non-binary images. Related metrics have been published in the literature [15,21–24], but each of these metrics fails to satisfy at least one of the aforementioned criteria. A custom measurement metric was designed to emphasize change along the edges of the unit cells using a 3-dimensional Sobel filter [25,26]. This makes it possible to resolve both within image changes (the x and y directions shown in Figure 6) and between-image changes (the z direction shown in Figure 6) using a single operation.

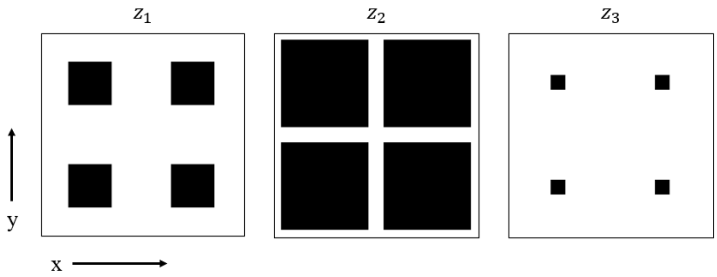


Figure 6: Visualization of the model convolved with Sobel filter

Specifically, we use the formulation of 3D Sobel operators defined by Amin et al. [26]:

$$S_x(:,:,-1) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_x(:,:,0) = \begin{bmatrix} -2 & 0 & 2 \\ -4 & 0 & 4 \\ -2 & 0 & 2 \end{bmatrix}, S_x(:,:,1) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$
$$S_y(:,:,-1) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, S_y(:,:,0) = \begin{bmatrix} -2 & -4 & -2 \\ 0 & 0 & 0 \\ 2 & 4 & 2 \end{bmatrix}, S_y(:,:,1) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (1)$$
$$S_z(:,:,-1) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{bmatrix}, S_z(:,:,0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, S_z(:,:,1) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

where $S_x, S_y, and\ S_z$ are the Sobel operators in the x, y, and z directions respectively.

$$\begin{aligned} G_{x,i} &= S_x(:,:,-1)conv(I_i) + S_x(:,:,0)conv(I_{i+1}) + S_x(:,:,1)conv(I_{i+2}) \\ G_{y,i} &= S_y(:,:,-1)conv(I_i) + S_y(:,:,0)conv(I_{i+1}) + S_y(:,:,1)conv(I_{i+2}) \\ G_{z,i} &= S_z(:,:,-1)conv(I_i) + S_z(:,:,0)conv(I_{i+1}) + S_z(:,:,1)conv(I_{i+2}) \end{aligned} \quad (2)$$

where $G_x, G_y, and\ G_z$ are the gradient array components in the x, y, and z directions respectively, I is the image, and i is the index between each gradient array and their respective images.

To directly compare the gradients, the x, y, and z components of the arrays were flattened. Then the root mean squared error (RMSE) between the consecutive gradients was measured, as that would be the indicator of "smoothness" between each dimension (Eq 3).

$$\begin{aligned} RMSE_{x,i} &= \sqrt{\frac{\sum_{j=1}^{N}\left(G_{x,i+1,j} - G_{x,i,j}\right)^2}{N}} \\ RMSE_{y,i} &= \sqrt{\frac{\sum_{j=1}^{N}\left(G_{y,i+1,j} - G_{y,i,j}\right)^2}{N}} \\ RMSE_{z,i} &= \sqrt{\frac{\sum_{j=1}^{N}\left(G_{z,i+1,j} - G_{z,i,j}\right)^2}{N}} \end{aligned} \quad (3)$$

where $RMSE_x, RMSE_y, and\ RMSE_z$ are the root mean squared errors of a pair of gradients in the x, y, and z directions respectively, j is the index that identifies the specific term in the gradient array, and N is the number of terms in a single gradient array.

Once the RMSE was calculated for the x, y, and z components, the average RMSE was calculated and normalized to create a final "smoothness value".

$$RMSE_i = \frac{RMSE_{x,i} + RMSE_{y,i} + RMSE_{z,i}}{3 \cdot RMSE_{max}} \quad (4)$$

where $RMSE_{max}$ is the maximum possible $RMSE_i$ which is calculated based on the filter used. The normalization of the RMSE allowed for the smoothness value to be represented as a percentage.

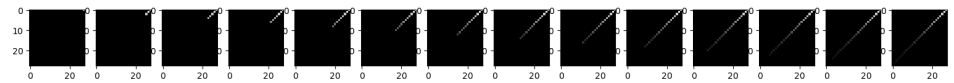
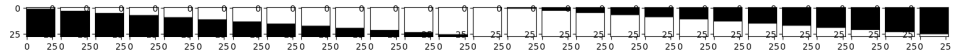
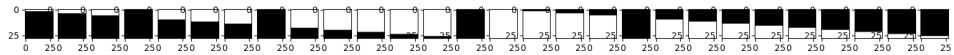
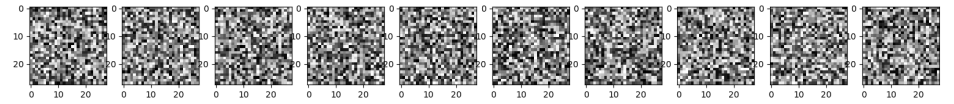
$$smoothness = (1 - avg(RMSE_i)) \cdot 100 \quad (5)$$

To validate and determine the limits of this measurement, multiple baseline evaluations were conducted. These evaluations consisted of the following (Table 2):

1. 1-pixel step-wise interpolation with decreasing pixel intensity values
2. Row-step-wise interpolation with transition occurring at fully solid shape
3. A "smooth" interpolation with abrupt changes
4. Random pixel arrays

Evaluation 1 was to determine the best possible interpolation value, as it is impossible to achieve 100% smoothness unless the unit cells are identical. Evaluation 2 was simply a test interpolation to determine the performance of the metric if the pixels were purely black and white. Evaluation 3 was to determine how the interpolations would be affected if a "bad" transition step was introduced in random locations. Evaluation 4 was chosen to evaluate the metric when the transitions lack a clearly defined edge.

Table 1: Smoothness baseline metric evaluations

Interpolation	Smoothness
	98.797%
	80.5%
	66.775%
	79.879%

4. Results

This section reports the performance of the trained VAE, assesses the qualitative and quantitative results from interpolations, and addresses the research questions outlined in the introduction. Overall, the purpose of this experiment was to demonstrate the feasibility of using VAE's to interpolate 2D unit cells and determine a metric for evaluating such interpolations.

4.1 Performance of Convolutional Variational Autoencoder (VAE)

Before using the results from a VAE, it should be determined whether the VAE is overfitting or underfitting the data. The performance of the VAE was measured using loss and coefficient of determination (R^2 value), which can be seen in Figure 7. The R^2 value was in excess of 95%, which indicates that the model is a good fit of the training data. The significant decrease in loss indicates that the generated unit cells began to match the original unit cells more closely. Given these results, the trained VAE was considered satisfactory for testing.

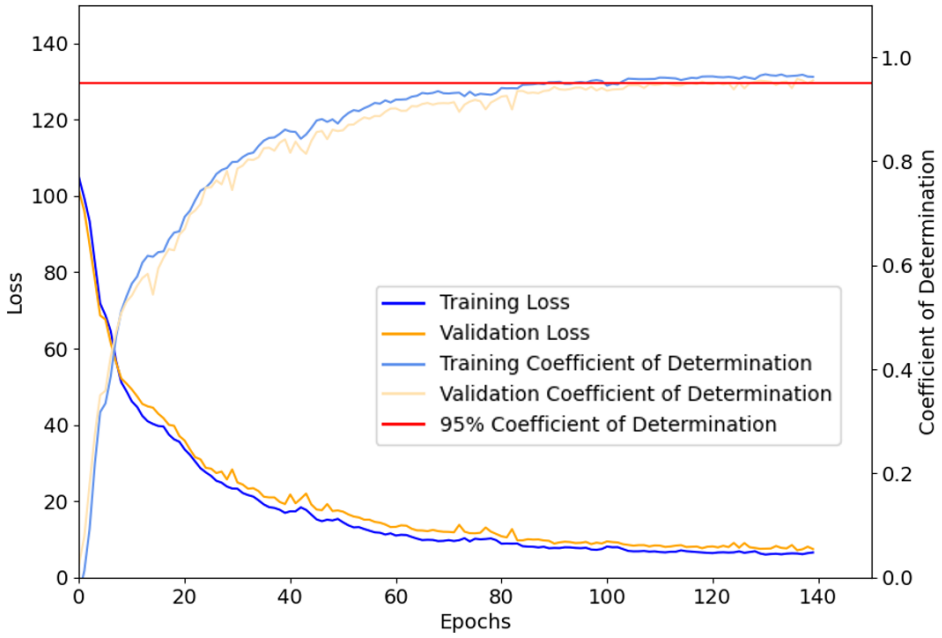


Figure 7: Performance of VAE with a latent dimensionality of 4

4.2 Interpolation Results

This section demonstrates how the smoothness evaluation is executed, and what it outputs. To address the research questions in the introduction, this section also shows the testing of different types of interpolations. Note that the decoded endpoint unit cells do not perfectly match the user provided unit cells (See Figure 8(a)), this was due to use of the test unit cells for interpolation. Since the VAE is trained on 85% of the data, it created the best reconstructions possible to match the provided test points. It was expected that the encoder would not be capable of perfect recreations of test points since the VAE contains loss. Whereas Figure 8(b) shows a method for visualizing the latent space, while helping to show where the interpolation(a) is oriented with respect to the rest of the latent space.

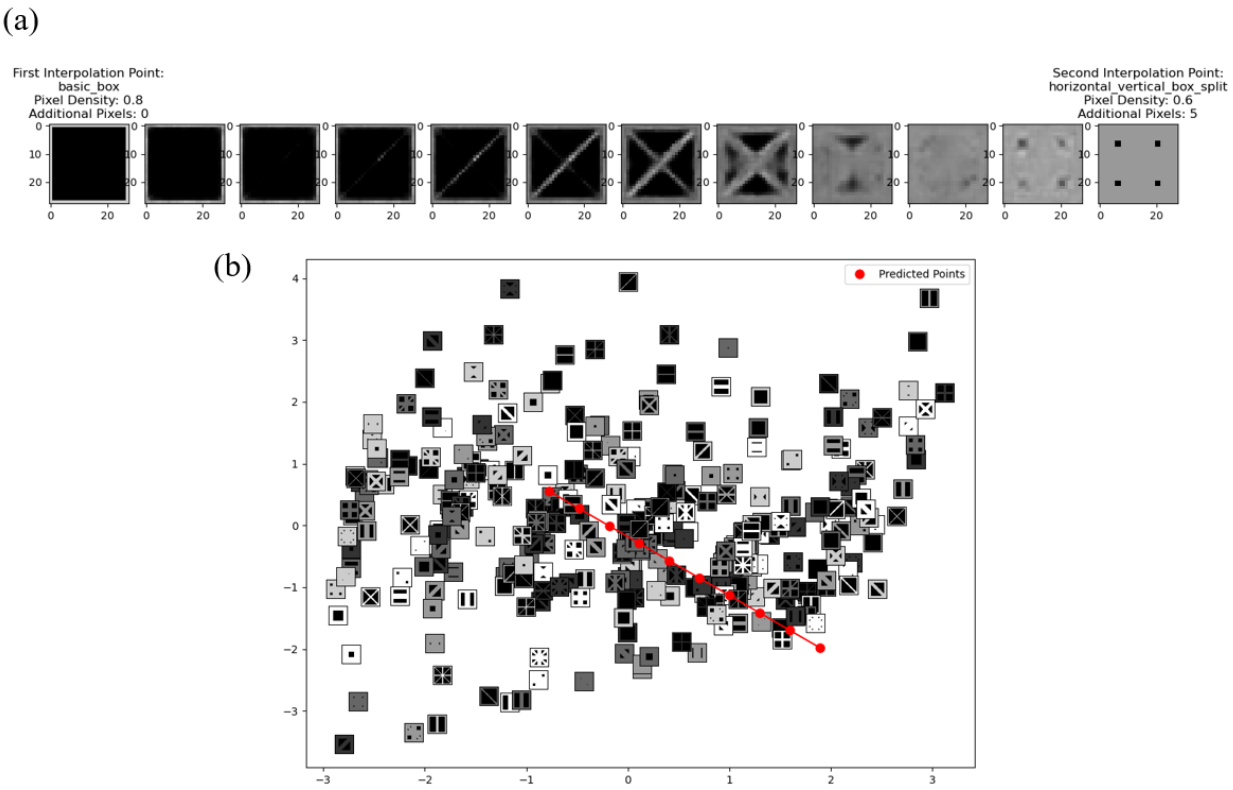
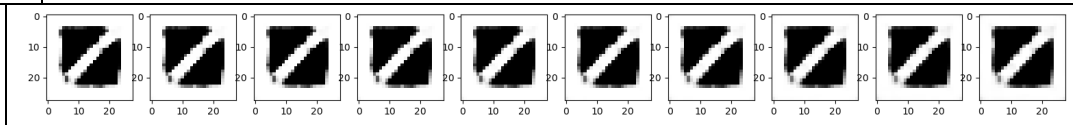
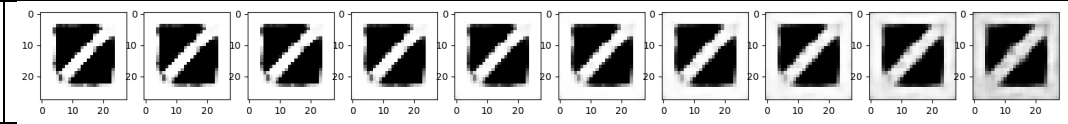
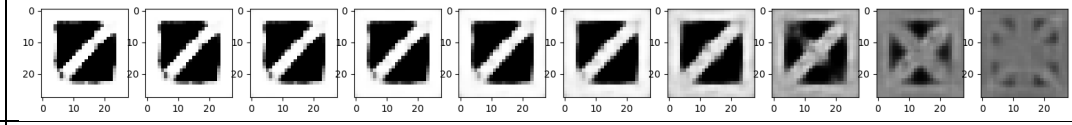
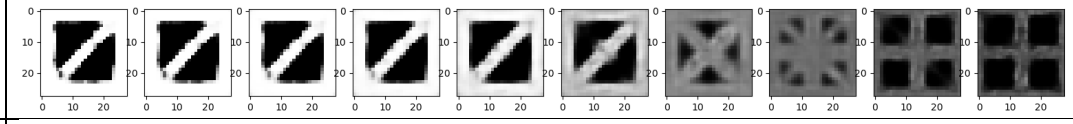
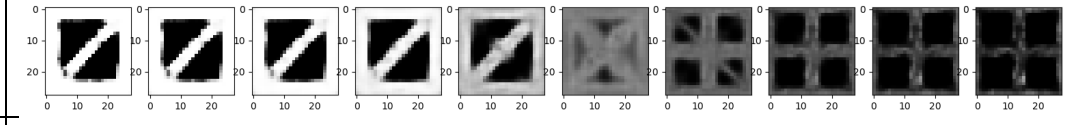
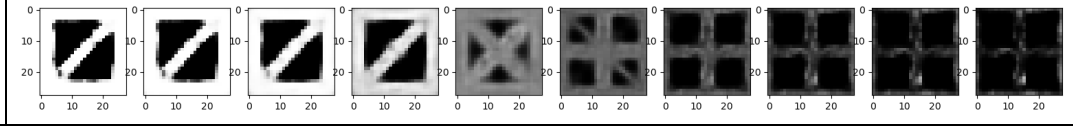


Figure 8: Demonstration of Interpolation: (a) Interpolation (b) PCA reduced latent space plot with the interpolation points super imposed in the space

How does distance in the latent space and transition length affect smoothness?

It was hypothesized that the distance in the latent space would result in a decrease in smoothness as points would become spread further apart. Where the length of the transition region would have a proportional relationship with the smoothness of the interpolations. To test this hypothesis, a variety of different interpolations were defined and assessed. First, the mean and standard deviation of the latent space were calculated, since the latent space is unitless, the standard deviations within the space serve as the best measurement for comparison between trained VAEs. Interpolations were performed from -3 standard deviations to 3 standard deviations from the mean. We tested sets of interpolations consisting of 5, 10 and 15 transition points. A sample of the interpolations can be seen in Table 2 for transition length 10.

Table 2: Visualization of transitions based on number of standard deviations with 10 interpolation points

Number of Standard Deviations Between Endpoints	Resulting Interpolations
1	
2	
3	
4	
5	
6	

Visually, the unit cells are most similar when there are more interpolation points and 1 standard deviation between the endpoints. To confirm this, the smoothness evaluation was conducted on all interpolations (See Figure 9).

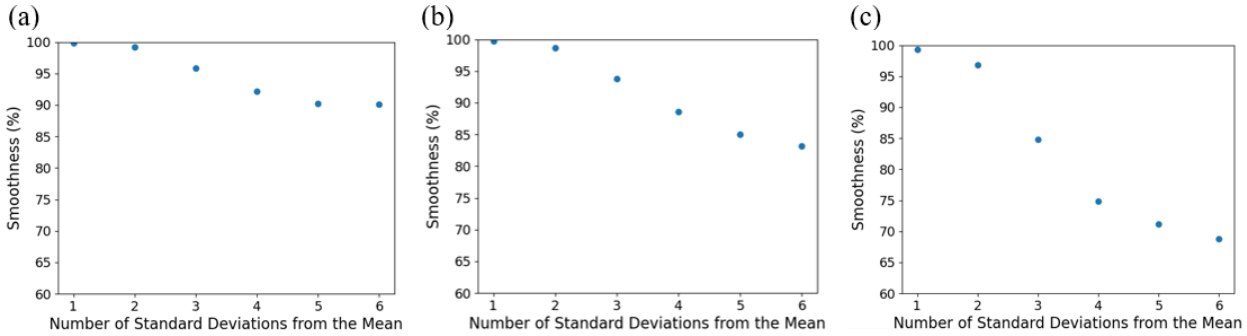


Figure 9: Smoothness vs Number of Standard Deviations in the Latent Space. (a) Analysis based on 15 interpolation points - More interpolation points in the latent space result in slightly poorer transitions as distance increases. (b) Analysis based on 10 interpolation points (c) Analysis based on 5 interpolation points - Fewer interpolation points result in significantly rougher transitions over a longer distance in the latent space.

To fully confirm the results, an ordinary least squares (OLS) regression was executed on the data gathered in Figure 9, where the dependent variable was percent smoothness, and the independent variables were transition length and distance in the latent space. This regression was found to be highly statistically significant ($p < 0.001$) as well as practically significant ($R^2 = 0.924$). The p-values in Table 4 confirmed that as the distance between endpoints increases, the smoothness decreases. The distance in the latent space has the most significant effect on smoothness. However, the results also indicate that transition length alone is unlikely to influence the smoothness value. Consequently, the combination of the two variables, noted by the cross-term in Table 4, does directly affect the smoothness. This indicates that the transition length only affects the smoothness when distance is also accounted for.

Table 3: Ordinary Least Square Regression

	Coefficient	Standard Error	p
Constant:	108.6807	3.882	$p < 0.0001$
Number of Standard Deviations (Distance):	-8.8616	0.997	$p < 0.0001$
Transition Length:	-0.4182	0.359	0.264
Cross-Term:	0.4601	0.092	$p < 0.0001$

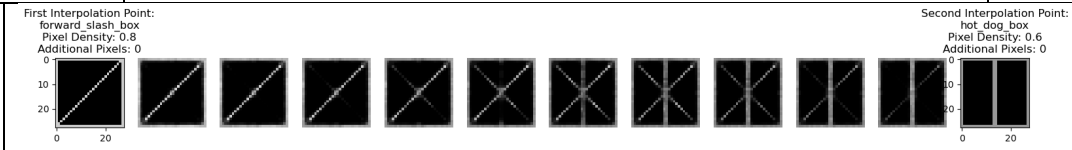
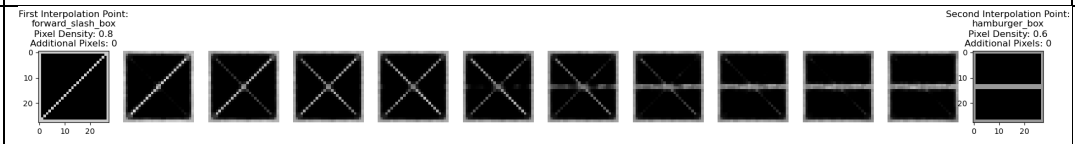
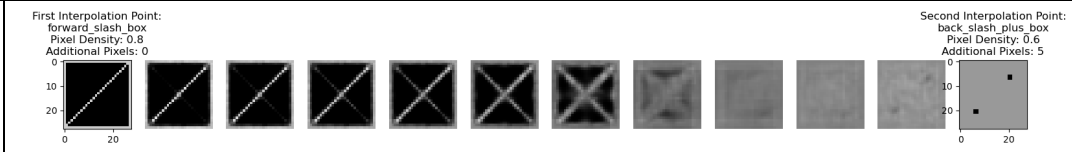
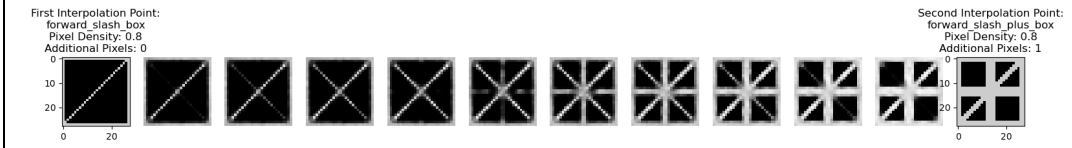
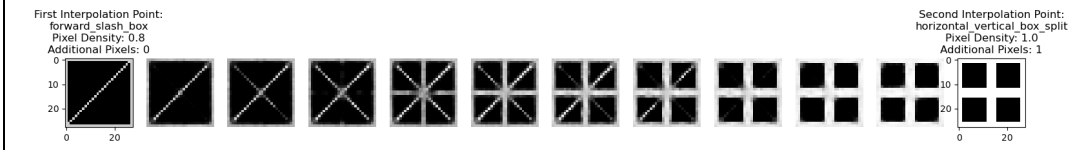
Given these results, the best interpolations can be produced by reducing the distance traveled in the latent space as much as possible. In cases where the distance is vast between endpoints, increasing the number of transitions will only improve the smoothness by a small amount. The smoothness is heavily reliant on distance within the latent space.

How do differences in unit cell topologies affect smoothness?

This section will explore the effects of the qualitative differences in unit cell shape with respect to smoothness. It is hypothesized that as the endpoints of an interpolation become more "different", their smoothness should decrease. Given the complexity of the latent space, visual

inspection may be deceptive, which is why it is important to address the unexpected. Table 5 consists of an array of various interpolations, all of which have the same initial interpolation point. This was to ensure that the interpolations were directly comparable to one another. Although the distances between the endpoints are not the same, this series of interpolations are ordered based on smoothness performance, where the best interpolation is Example 1.

Table 4: Examples of different topologies and their effect on smoothness of the interpolation

Example	Example Interpolations	Smoothness (%)
1		95.946
2		95.398
3		91.576
4		90.928
5		88.141

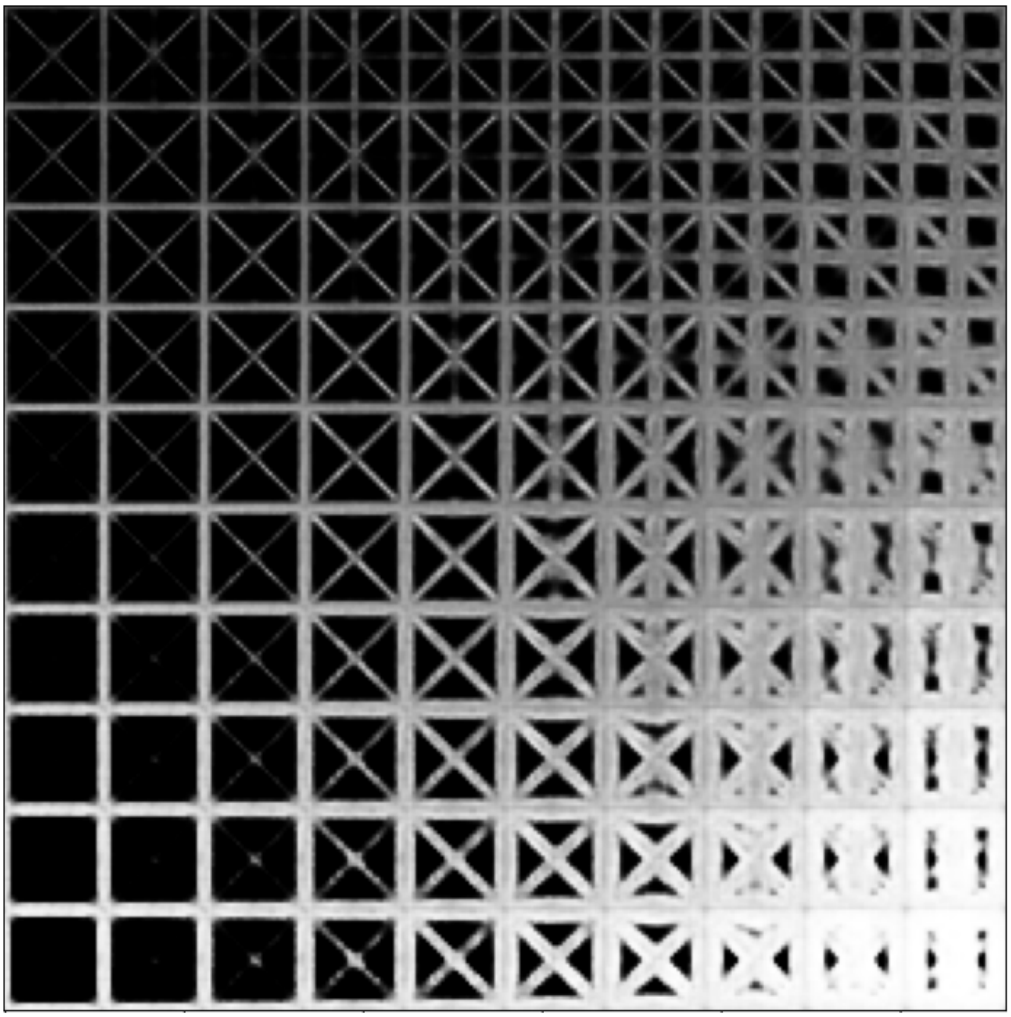
The results from Table 5 were somewhat as expected, but this section will outline why the results here were reasonable. From a qualitative standpoint, the rankings were expected to order Example 1 > 2 > 4 > 5 > 3, however, the smoothness metric is based on the variation along all the unit cells, not solely the endpoints. Example interpolations 1 and 2 performed similarly well, as both were a rotation of the initial endpoint. Example 3 performed slightly worse than 1 and 2, but there were a lot of pixels that needed to be activated. However, based on the qualitative evaluation it would appear that 3 should have performed the worst. Since the smoothness evaluation is highly dependent on the amount of similarity between consecutive unit cells, the endpoint unit cells in Example 3 would have increased the smoothness percentage significantly. Finally, example 4 performed slightly better than example 5, as the diagonal was not removed in example 4, meaning that fewer changes occurred.

5. Discussion

This section will discuss the results obtained from a transition region developed between four different types of unit cells, while highlighting the many possibilities that are afforded by VAEs and their future work. This work underscores the potential of VAEs for supporting transition region design. However, the smoothness of the transition region is primarily predicted by the distance in the latent space. Therefore, our hypothesis is partially refuted, as the transition length did not have the expected impact on smoothness.

Figure 10(a) shows how powerful VAEs are by demonstrating a smooth transition region between four distinct unit cells. Such a structure represents how a transition boundary would be developed between four different lattice topologies. Multi-lattice designs incorporating more than two types of lattices could expand design possibilities but creating transition regions has been a main restriction. As far as we know, this type of transition has only been reproduced using a VAE, as it is extremely computationally intensive using other methods. Figure 10(b) represents the locations of each of the predicted points in the PCA reduced latent space. As established previously, the closer points will have smoother transitions based on the smoothness metric. This type of visual aid could be utilized to quickly determine which transition regions perform poorly.

(a)



(b)

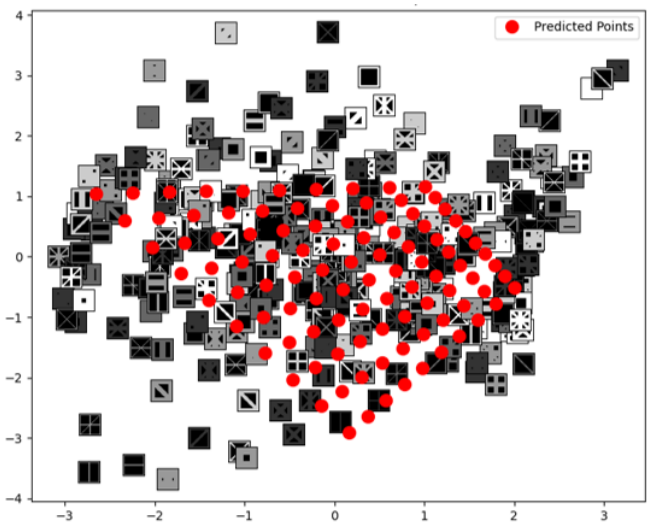


Figure 10: (a) Example of mesh grid interpolation (b) PCA reduced representation of the mesh grid interpolation in the latent space

6. Conclusion

In cases where the loading varies throughout a part, it may be necessary to use multiple lattice cell types, also known as multi-lattice structures. In such structures, abrupt transitions between lattice types may cause stress concentrations, making the boundary a primary failure point; thus, transition regions should be created between each lattice cell type. This work demonstrates and assesses a method for using variational autoencoders to automate the creation of transitions amongst multi-lattice structures. In comparison to other computational approaches, this work focused on analyzing how a VAE latent space affected unit cell interpolations by answering two research questions.

First, we answered how distance in the latent space and the number of transitions affects the smoothness of an interpolation. Specifically, we assessed the smoothness of lattice interpolations produced using a VAE, evaluated using 3D Sobel filters. Using an OLS regression, it was found that distance between endpoints in the latent space had the most significant impact on smoothness, whereas the number of steps in the transition were unlikely to affect the smoothness alone. Working in unison, the two variables only had some impact on the smoothness value. Given these results, the best interpolations can be produced by reducing the distance traveled in the latent space as much as possible. This indicates the need for using latent space optimization methods to minimize the distance between desired end points during training.

Secondly, we qualitatively evaluated comparable interpolations to determine whether their performance met our expectations. The results from these tests indicate that human qualitative similarities may not perform as well as similarities made by the model. Based on the smoothness evaluations of our interpolations, humans cannot accurately and precisely determine which set of interpolations is most smooth.

Future work will also focus on modifications to be made that encourage manufacturability of transition regions. This will require that the connectivity between cells can be ensured, as the physical properties of the unit cells are dependent on their ability to be created. This will be followed by determining the best method for analyzing the functionality of these transition regions, as they are extremely complex structures which require physical analyses to be useful to designers. It is important to determine if the smoothness metric has any correlation to physical properties within a transition region. As discussed previously, other works have incorporated physical properties into their latent space optimization [14], which may prove to be the best option for organizing the latent space.

In order to take advantage of the latent space created by VAEs, we must determine how to address the effects on distance within that space. Given that the distance has such a huge impact on smoothness, the unit cells that perform best will exist in the center of the space where they are close to all the other unit cell. Unit cells on the boundaries of the latent space have less information, so they are difficult to decode without significant loss of the original unit cell. Furthermore, the resolution of the outputs is highly dependent on the resolution of the training data. The data size increases by an exponent of the data dimensionality. Consequentially, as this work expands into the 3-dimensional space, we will have to address extreme increases in data size and therefore training times of the VAE.

References:

[1] Hanks, B., Berthel, J., Frecker, M., and Simpson, T. W., 2020, "Mechanical Properties of Additively Manufactured Metal Lattice Structures: Data Review and Design Interface," Addit Manuf, **35**.

[2] Plocher, J., and Panesar, A., 2020, "Effect of Density and Unit Cell Size Grading on the Stiffness and Energy Absorption of Short Fibre-Reinforced Functionally Graded Lattice Structures," Addit Manuf, **33**, p. 101171.

[3] Gok, M. G., 2021, "Creation and Finite-Element Analysis of Multi-Lattice Structure Design in Hip Stem Implant to Reduce the Stress-Shielding Effect."

[4] Li, D., Dai, N., Tang, Y., Dong, G., and Zhao, Y. F., 2019, "Design and Optimization of Graded Cellular Structures with Triply Periodic Level Surface-Based Topological Shapes," Journal of Mechanical Design, Transactions of the ASME, **141**(7).

[5] Wang, Y., Zhang, L., Daynes, S., Zhang, H., Feih, S., and Wang, M. Y., 2018, "Design of Graded Lattice Structure with Optimized Mesostructures for Additive Manufacturing," Mater Des, **142**, pp. 114–123.

[6] Plocher, J., and Panesar, A., 2019, "Review on Design and Structural Optimisation in Additive Manufacturing: Towards next-Generation Lightweight Structures," Mater Des, **183**(108164).

[7] Maskery, I., Hussey, A., Panesar, A., Aremu, A., Tuck, C., Ashcroft, I., and Hague, R., 2017, "An Investigation into Reinforced and Functionally Graded Lattice Structures:," Journal of Cellular Plastics, **53**(2), pp. 151–165.

[8] Panesar, A., Abdi, M., Hickman, D., and Ashcroft, I., 2018, "Strategies for Functionally Graded Lattice Structures Derived Using Topology Optimisation for Additive Manufacturing," Addit Manuf, **19**, pp. 81–94.

[9] Wang, C., Zhu, J. H., Zhang, W. H., Li, S. Y., and Kong, J., 2018, "Concurrent Topology Optimization Design of Structures and Non-Uniform Parameterized Lattice Microstructures," Structural and Multidisciplinary Optimization, **58**, pp. 35–50.

[10] Sanders, E. D., Pereira, A., and Paulino, G., 2021, "Optimal and Continuous Multilattice Embedding," Sci Adv, **7**(16), p. 4838.

[11] Sahariah, B. J., Namdeo, A., and Khanikar, P., 2022, "Composite-Inspired Multilattice Metamaterial Structure: An Auxetic Lattice Design with Improved Strength and Energy Absorption," Mater Today Commun, **30**, p. 103159.

[12] Kang, D., Park, S., Son, Y., Yeon, S., Kim, S. H., and Kim, I., 2019, "Multi-Lattice Inner Structures for High-Strength and Light-Weight in Metal Selective Laser Melting Process," Mater Des, **175**, p. 107786.

[13] Wang, L., van Beek, A., Da, D., Chan, Y.-C., Zhu, P., and Chen, W., "Data-Driven Multiscale Design of Cellular Composites with Multiclass Microstructures for Natural Frequency Maximization."

[14] Wang, L., Chan, Y. C., Ahmed, F., Liu, Z., Zhu, P., and Chen, W., 2020, "Deep Generative Modeling for Mechanistic-Based Learning and Design of Metamaterial Systems," Comput Methods Appl Mech Eng, **372**.

[15] Wang, J., Chen, W. (Wayne), Da, D., Fuge, M., and Rai, R., 2022, "IH-GAN: A Conditional Generative Model for Implicit Surface-Based Inverse Design of Cellular Structures," Comput Methods Appl Mech Eng, **396**, p. 115060.

[16] Regenwetter, L., Nobari, A. H., and Ahmed, F., 2022, "Deep Generative Models in Engineering Design: A Review," Journal of Mechanical Design, **144**(7).

[17] Kingma, D. P., and Welling, M., 2014, "Auto-Encoding Variational Bayes," *Conference Proceedings: Papers Accepted to the International Conference on Learning Representations (ICLR) 2014*.

[18] Xue, T., Wallin, T. J., Menguc, Y., Adriaenssens, S., and Chiaramonte, M., 2020, "Machine Learning Generative Models for Automatic Design of Multi-Material 3D Printed Composite Solids," Extreme Mech Lett, **41**, p. 100992.

[19] Murphy, C., Meisel, N., Simpson, T. W., and McComb, C., 2018, "Using Autoencoded Voxel Patterns to Predict Part Mass, Required Support Material, and Build Time," *Solid Freeform Fabrication 2018: Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium*.

[20] McComb, C., "Toward the Rapid Design of Engineered Systems Through Deep Neural Networks," *Design Computing and Cognition 2018*.

[21] Cristovao, P., Nakada, H., Tanimura, Y., and Asoh, H., 2020, "Generating In-Between Images Through Learned Latent Space Representation Using Variational Autoencoders," IEEE Access, **8**, pp. 149456–149467.

[22] Vyas, S., Chen, T. J., Mohanty, R. R., Jiang, P., and Krishnamurthy, V. R., 2021, "Latent Embedded Graphs for Image and Shape Interpolation," CAD Computer Aided Design, **140**, p. 103091.

[23] Hore, A., and Ziou, D., 2010, "Image Quality Metrics: PSNR vs. SSIM," *2010 20th International Conference on Pattern Recognition*, IEEE, Istanbul, Turkey.

[24] Amanatiadis, A., and Andreadis, I., 2009, "A Survey on Evaluation Methods for Image Interpolation," Meas. Sci. Technol, **20**(104015).

[25] Shrivakshan, G. T., and Chandrasekar, C., 2012, "A Comparison of Various Edge Detection Techniques Used in Image Processing," International Journal of Computer Science Issues, **9**(5), pp. 269–276.

[26] Amin, A., and Deriche, M., 2015, "A New Approach for Salt Dome Detection Using a 3D Multidirectional Edge Detector," Applied Geophysics, **12**(3), pp. 334–342.