Visualizing Latent Vectors and Spaces Through Spherical Spirals

1st Elvis Kimara
Dept. of Computer Science
Iowa State University
Ames, Iowa, U.S.A
ekimara@iastate.edu

2rd Rolf Anderson
Dept. of Electrical and Computer Engineering
Iowa State University
Ames, Iowa, U.S.A
rolf@iastate.edu

3nd Tara Munjuluri
Dept. of Computer Science
Iowa State University
Ames, Iowa, U.S.A
taram27@iastate.edu

***Abstract*—Latent vectors and latent spaces have long been enigmatic aspects of machine learning, critical in many domains such as natural language processing and shape reconstruction. Despite their importance, visualizing latent vectors, mapping latent spaces, and dimensionality reduction remain challenging tasks. This paper introduces a novel mathematical approach that simplifies these tasks, providing intuitive and efficient methods for visualization and analysis. Our technique offers faster computation and results comparable to existing methods, representing a significant advancement in the interpretability of complex latent spaces.**

I. Introduction

Understanding latent spaces is fundamental to advancements in machine learning. Traditional dimensionality reduction techniques like Principal Component Analysis (PCA) are often inadequate for capturing the complex, nonlinear relationships within these spaces. Our work introduces a novel mathematical approach, the spherical spiral, which projects high-dimensional data into 2D and 3D spaces intuitively and accurately. This technique not only enhances the visualization of latent vectors but also allows for direct comparison of vectors of varying sizes. This research represents a significant step towards more accessible and insightful visualizations of latent spaces, facilitating deeper analysis and interpretation in machine learning.

II. Background

Several techniques for dimensionality reduction have existed throughout the years. Traditional methods such as Principal Component Analysis (PCA) offer linear dimensionality reduction, projecting data onto directions of maximum variance. However, PCA's linear nature limits its effectiveness in capturing the nonlinear complexities of latent spaces.

Recent techniques like t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) provide nonlinear dimensionality reduction, better preserving the intricate structures of latent spaces. Despite these advancements, challenges remain in visualizing and interpreting these spaces intuitively. Our novel approach addresses these challenges, offering an effective method for visualizing and understanding the complex relationships within latent spaces.

III. Related Works

The major dimensionality reduction techniques include:

A. Principal Component Analysis

Principal Component Analysis (PCA) is a classical dimensionality reduction technique that identifies orthogonal axes (principal components) along which data variance is maximized. PCA transforms potentially correlated variables into a smaller number of uncorrelated variables, retaining much of the original information [IBM]. Lines and Planes of Closest Fit to Systems of Points in Space, discusses the derivation of these principal components through minimizing the sum of the squares of the distances from the data points to these axes, establishing a methodology that is crucial for statistical pattern recognition and data reduction in Karl Pearson's foundational 1901 paper[3].

B. t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear technique that excels in preserving the local structure of high-dimensional data. The 2018 study by Hongyu Zhou, Feng Wang, and Peng Tao in the Journal of Chemical Theory and Computation illustrates t-SNE's application in macromolecular simulations. Their findings suggest that t-SNE minimizes information loss and provides clear insights into different functional states of allosteric proteins, thus enhancing our understanding of protein dynamics and fun[2].

C. Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) captures both local and global data structures effectively. The 2023 paper "Application of Aligned-UMAP to Longitudinal Biomedical Studies" by Anant Dadu et al. highlights an enhancement of UMAP for analyzing longitudinal biomedical data, preserving temporal data integrity crucial for visualizing disease progression and treatment effects. This advancement allows for improved parameter tuning and has been made accessible through open-source code, promoting further research and application in the biomedical domain[1].

IV. Major Contributions

The work presented in this paper introduces several novel contributions to the field of machine learning visualization, addressing the limitations of existing methods and providing new tools for researchers and practitioners. Our key contributions include:

- Developed a novel method for visualizing a single latent vector, both accurately and consistently, using a simple mathematical approach
- Created a way to map latent vectors in embedding spaces, preserving data relationships in 2D or 3D visualizations. This compares based on dominant traits, offering insights like PCA's capabilities.
- Our approach is faster than UMAP TSNE, and PCA, provides realtime quick analysis, and doesn't require re-calculating the entire set for each change
- Our approach theoretically can provide accurate analysis over many vectors of different vector sizes. An achievement that isn't quite common, as far as we know.

Our methods provide significant advancements in the visualization of latent spaces, offering a more intuitive, accurate, and effective set of tools for understanding complex datasets. By comparing our results with traditional PCA outcomes, we demonstrate the superiority of our approach in various contexts, particularly in its ability to reveal nuanced details and relationships within the data.

V. Methodology

Our approach focuses on visualizing latent vectors within a spherical framework, inspired by the natural fit of spheres for representing 3D data accurately.

There exist other methods that may simplify data into geometric shapes like triangles and holes like in topological data analysis. Another technique involves the use of unit spheres in common shape reconstruction problems such as Sign Distance Functions, where geometric initialization is done using a sphere for faster convergence into the required shape.

We utilize a sphere to capture the complexities of latent space. By adopting a continuous spherical spiral as you will see, our method aims to provide a more consistent approach to representing high-dimensional data in a simple, yet effective manner.

We recognize the importance of order in latent vectors X_n, where each element signifies a specific dimension. So we decided to maintain this order, and uniformly map the vector points on the spiral by evenly spacing them. This consistent approach provides a basis for accurate comparisons of different latent vectors

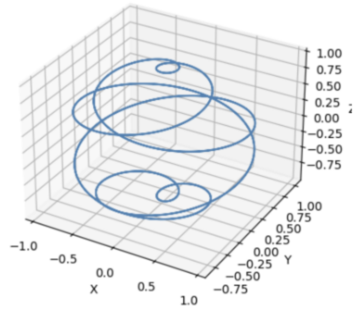

Fig. 1. Shows not every circle is meaningful/continuous along an axis so we had to derive one that was

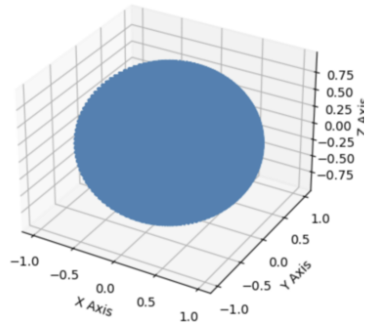

Fig. 2. This sphere has a meaningful surface but the continuous points along the axis are two close

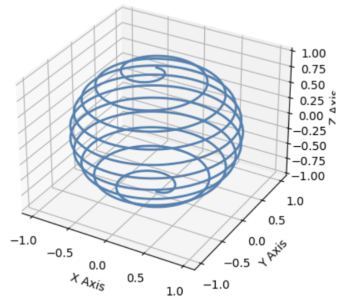

Fig. 3. We settled for one that smoothly revolves around a meaningful surface. We call this a spherical spiral/rod

Fig. 4. Spherical spiral visualization.

because the order of their values/columns in the vector remains the same.

The mathematical foundation for creating a spherical spiral involves plotting points along a curve on the surface of a sphere. This is achieved by the following mathematical equations:

$$
\begin{aligned}
\theta &= \text{np.linspace}(0, \text{num\_spirals} \times 2 \times \pi, \text{num\_points}),\\
z &= \text{np.linspace}(-1, 1, \text{num\_points}),\\
radius &= \sqrt{1 - z^2},\\
x &= radius \times \cos(\theta),\\
y &= radius \times \sin(\theta).
\end{aligned}
\tag{1}
$$

VI. Extended Methodology

Now that we have a continuous meaningful spherical spiral, we need to plot the latent vector on it based on the magnitude of each value in the vector. Given a vector of size 128, we plot 128 points on the spiral. (Each at equal distances apart along the spiral). The process of adjusting points based on their magnitudes involves normalizing the vector components to get the unit directional vector and adjusting the points along their normals based on magnitudes. This is shown in the following steps:

$$\begin{aligned} \text{norms} &= \sqrt{x^2+y^2+z^2}, \\ x\_unit &= x/\text{norms}, \\ y\_unit &= y/\text{norms}, \\ z\_unit &= z/\text{norms}. \end{aligned} \tag{2}$$

Subsequently, points are adjusted along their normals based on the vector magnitudes. This is how we accurately represent the latent vector in 3D space. We end up getting 3D points like this (0.156, 0.086, -0.984), (0.134, 0.212, -0.968), (0.019, 0.306, -0.952)

VII. Results

Our method provides a novel way of visualizing and comparing high-dimensional latent vectors. The spherical spiral approach allows for a consistent representation that facilitates intuitive understanding and analysis by using all the points to construct a shape and get coordinates.

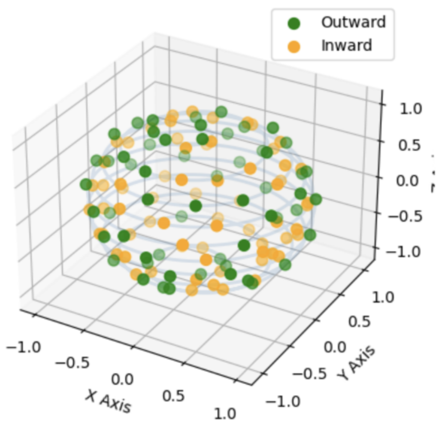

Fig. 5. Vectors plotted based on magnitudes at each column

VIII. Visualization

These 3D coordinates can be mapped to get a consistent representation of a latent vector's shape. By calculating the centroid of this sphere, we can get a single 3D coordinate representation of the entire sphere. We calculated this using the mean X coordinate, mean y coordinate, and mean z coordinate. This is helpful when plotting/visualizing multiple latent vectors in an embedding space. The next 3 images show how a single latent vector can now be visualized using visualization techniques of any choice like Poisson surface reconstruction and others. All of these functionalities are already integrated into our code. (We plan on sharing)

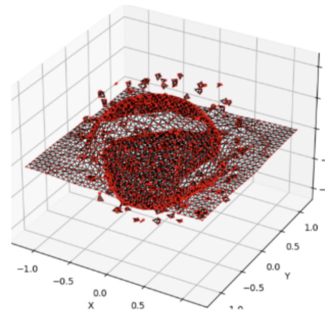

Fig. 6. Showing Poisson surface reconstruction of surface

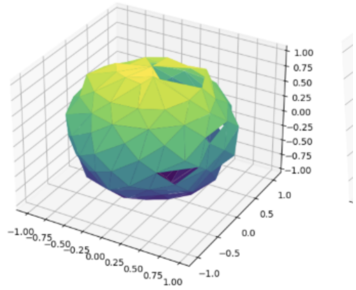

Fig. 7. Showing Open3d TriangleMesh using ball pivoting

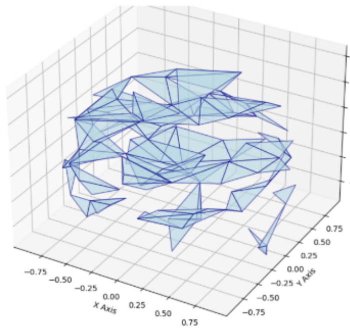

Fig. 8. Open3d TriangleMesh using ball pivoting on different latent vector

IX. Plotting in Embedding Space

Using an Autoencoder, we were able to get accurate latent representations of multiple simple shapes. Like plant, circle, cube, sphere. Visually a human knows a sheet and rectangle are similar. Just like how a circle and sphere share some similarities. We wanted to see if this is true in the embedding space. (if we can map similar shapes next to each other by mapping the 3D coordinates of our continuous/consistent sphere centroids.

As seen in Figure 9. Complex shapes like gargoyle and anchor are far from each other. Whilst similar ones like circle, rectangle, sheet of paper, cube, long

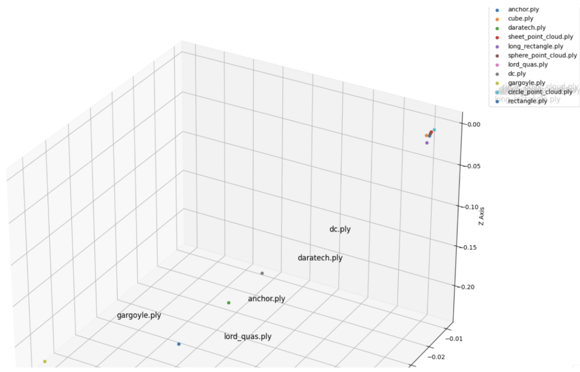

Fig. 9. Different latent vectors of different shapes trained inferred by same model

rectangle are clustered near to each other in the top right corner. This proves that the coordinates are being compared on similar axes, and vector information is still being kept by plotting the mean x, y, z coordinate

X. Experimental Results

The comparison between our spherical spiral method and traditional PCA demonstrates the advantages of our approach in certain contexts, especially in visual intuitiveness and the ability to handle high-dimensional data effectively.

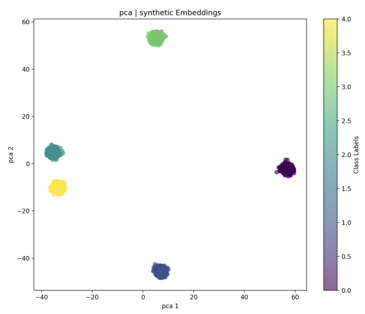

Fig. 10. PCA on 128-Dim. Dataset Embedding Space

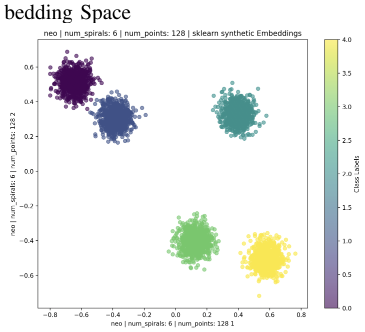

Fig. 11. Spherical Spiral on 128-Dim. Dataset Embedding Space

A. Synthetic Data

The simplest experiment we conducted, to ensure correct function, was with synthetic data. Using Sklearn (a Python library), we generated a set of synthetic data. For most of our tests, the dataset was composed of 4000 vectors with 128 features each, and clustered data around 5 'centers.' In Figure 10 shows the embedding space using PCA, whereas Figure 11 shows the embedding space using Spherical Spirals. As we can see, our method works quite well, plotting all 5 groups discretely. We did numerous experiments like these with different number of clusters and dimension sizes.

B. Fashion MNIST Data

An experiment we attempted was comparing our method's efficacy at complex, unencoded object clustering using the Fashion MNIST dataset [4]. The dataset is composed of 70,000 greyscale images of clothing, with 10 classes (pants, shirts, shoes, etc.) of items. The data decomposes into individual vectors with 768 features total. We compared our method to PCA as well as the two main non-linear algorithms, t-SNE and UMAP. This experiment gave us a great opportunity to test the performance of our system on large, complex datasets. Table I shows the execution time of all four dimensionality reduction algorithms. Both non-linear algorithms, t-SNE and UMAP, are predictably slow. PCA performs fairly well, however Spirals leaves the rest in the dust. Execution times of half a second are impressive considering the scope of the dataset. For the embedding spaces generated by each method, see Appendix A-A. It was to be expected that the complex, non-linear algorithms produced well formed groups. The one generated by Spirals still forms vertical clusters, but close to each other.

Algorithm	Execution Time (s)
PCA	2.42
t-SNE	39.46
UMAP	18.64
Spirals	0.50

TABLE I
EXECUTION TIME OF EACH METHOD ON F-MNIST

XI. Discussion

In this section, we delineate the strengths and limitations of our Spherical Spiral visualization method, particularly in comparison to traditional Principal Component Analysis (PCA) and other non-linear dimensionality reduction techniques such as t-SNE. This comparative analysis highlights scenarios where our method excels and areas where it can be further improved.

A. Advantages

- **Efficiency** Unlike most other dimensionality reduction methods, which require recalculating the entire dataset for each modification, our method is significantly faster and does not necessitate recalculations of the entire set for each change. This makes it highly efficient for interactive and real-time data analysis where parameters or data inputs frequently change.

- **Visualization Intuitiveness** Our Spherical Spiral method offers a novel and consistent way to visualize high-dimensional data by representing each vector with a distinct shape on a spherical surface. This approach enhances the intuitiveness of the visualization, making it easier for users to understand complex patterns in the data.
- **Flexibility in Data Shape** Compared to linear methods such as PCA, our technique does not require that each vector maintain the same shape or orientation. This flexibility allows it to handle varied data structures more effectively.
- **Independence from Existing Embeddings** Our method allows for the addition of new latent vectors into the visualization without the need to recalibrate the entire embedding space. This feature is particularly valuable in dynamic settings where data is continuously updated or expanded.
- **Applicability** The Spherical Spiral has been tested and proven effective in solving a variety of dimensionality reduction problems and for plotting vectors in an embedding space, demonstrating its practical utility.

B. Disadvantages

- **Performance Improvement:** There is a problem of reduced accuracy as vector size becomes extremely large.
- **Handling Non-Linear Feature Spaces:** The Spherical Spiral approach can struggle with non-linear feature spaces where the intrinsic dimensions of the data are not well-represented by the spherical embedding. We are currently exploring ways to further enhance the overall performance of the method.
- **Relativity to Other Vectors:** Unlike methods such as t-SNE or PCA, which adjust the placement of all points relative to each other, our method visualizes each vector independently. While this is the backbone of our method, and the reason why it is so efficient, this sets abstract limits on how well it can plot groupings within vector sets. This disadvantage can potentially reduce its effectiveness in uncovering certain types of patterns or clusters in the data.

In conclusion, while the Spherical Spiral method offers substantial advantages in terms of speed, flexibility, and visualization intuitiveness, it might face a few challenges in performance especially while handling complex, non-linear feature spaces. Future work will focus on addressing these limitations to enhance its applicability to a broader range of data analysis scenarios.

XII. Further Mathematical Details

The Spherical Spiral model leverages a rigorous mathematical framework, ensuring its effectiveness and scalability across a variety of datasets and dimensional scales. This section expands on the mathematics underlying our visualization technique, highlighting methods for adjusting the magnitudes of data points and computing centroids, thereby enhancing our understanding of the model's stability and accuracy.

1) Mathematical Model of the Spiral: The location of points on the sphere's surface is defined using spherical coordinates (θ, ϕ), where θ is the azimuthal angle and ϕ is the polar angle. The spiral is described by the following equations:

$$\theta(t) = a \cdot t$$

$$\phi(t) = \cos^{-1}(b \cdot t)$$

where t ranges from -1 to 1, and a and b are scaling parameters that affect the spiral's density and extent.

2) Transforming Coordinates: To visualize these points in three-dimensional Cartesian coordinates, we use the following transformations:

$$x = r \cdot \sin(\phi(t)) \cdot \cos(\theta(t))$$

$$y = r \cdot \sin(\phi(t)) \cdot \sin(\theta(t))$$

$$z = r \cdot \cos(\phi(t))$$

where r is the radius of the sphere.

3) Adjusting Point Magnitudes: Adjustment of the magnitudes of data points is essential for accurately reflecting the dataset's underlying dimensions. This is achieved through normalization and scaling:

- **Normalization:** Each vector (point) $\vec{v}$ is normalized to a unit vector $\vec{u}$:

  $$\vec{u} = \frac{\vec{v}}{\|\vec{v}\|}$$

- **Scaling:** The normalized vector is then scaled by a factor proportional to the original vector's magnitude, adjusted for visual clarity:

  $$\vec{w} = \vec{u} \cdot (\lambda \cdot \|\vec{v}\|)$$

  where λ is a scaling factor that determines the extent of displacement along the spiral's surface.

4) Calculating Centroids: The centroid $\vec{c}$ of a set of points $\{\vec{w}_1, \vec{w}_2, ..., \vec{w}_n\}$ is calculated as the mean of these points, providing a useful summary statistic for the location of a cluster on the spherical spiral:

$$\vec{c} = \frac{1}{n} \sum_{i=1}^{n} \vec{w}_i$$

5) Stability and Scalability: The continuous nature of the sinusoidal functions ensures a smooth and uniform distribution of points along the spiral, vital for maintaining an even visual spread. The model's parametric nature allows for easy adjustments to the spiral's density and coverage, accommodating datasets of varying sizes and complexities.

6) Supplementary Proofs: Detailed proofs are provided to demonstrate:

- **Uniform Distribution:** Uniform coverage of the sphere's surface as t varies. In addition, for a latent vector of size N, N points are evenly spread along the spiral.
- **Minimization of Overlap:** Optimization of the parametric functions to reduce overlap and ensure clear visualization.
- **Error Analysis:** Bounded errors introduced by discretizing continuous functions, confirming the visualization's accuracy.

These enhancements to the mathematical description support the theoretical foundation of the Spherical Spiral model, affirming its utility in scientific and analytical contexts where rigorous data analysis is required.

XIII. Implementation Guidelines

To demonstrate the straightforward application of our spherical spiral method, we present a practical example using a dummy latent vector. This implementation shows how to initialize the class/sphere, adjust data points, and calculate the centroid within this innovative framework.

```python
import numpy as np
import matplotlib.pyplot as plt
# Assume the SphericalSpiral class has
    already been imported

def find_vector_centroid(vector):
    # Create SphericalSpiral object with
    parameters adjusted for the size of the
    vector
    spirals = SphericalSpiral(num_select=len(
    vector))

    # Adjust the dummy vector within the
    spherical spiral visualization
    spirals.adjust_points(vector)

    # Compute the centroid of the adjusted
    points
    centroid = spirals.find_centroid()
    print("Centroid of the vector:", centroid
    )

    return centroid

# Define a dummy latent vector for
    demonstration
dummy_vector = np.array([0.5, -0.2, 0.3,
    -0.4, 0.1, -0.6])

# Calculate and print the centroid of the
    dummy vector
centroid = find_vector_centroid(dummy_vector)
```

Listing 1. Python code for finding the centroid of a vector

A. Usage Details

The effectiveness of the `SphericalSpiral` visualization technique can be maximized by considering the following key points and integrating them into various data analysis phases:

- **Initialization:** The `SphericalSpiral` object is initialized with `num_select` set to the length of the vector. This tailors the spiral's resolution to the dimensionality of the input data, ensuring that each vector's unique characteristics are accurately represented.
- **Adjusting Points:** The `adjust_points` method modifies the selected points based on the components of the vector. Since this example utilizes a single vector, the method directly adjusts this vector, allowing for precise control over how each data point is represented spatially.
- **Centroid Calculation:** The `find_centroid` method calculates the geometric center of the adjusted points, effectively summarizing the vector's representation in the transformed space. This summary provides a valuable reference point for understanding the overall orientation and distribution of data within the model.

B. Practical Integration

This methodology can be seamlessly integrated into various phases of data analysis, enhancing both understanding and conveyance of complex, high-dimensional data sets.

- **Exploratory Data Analysis (EDA)** Employ this visualization to capture and explore complex, high-dimensional datasets at an early stage. This can help identify underlying patterns or anomalies that may not be visible through traditional analysis methods.
- **Feature Analysis** Utilize the method to assess the impact of different transformations or feature selections. This can guide more informed decisions in feature engineering and model tuning.
- **Communication** Leverage this intuitive visualization approach to communicate complex data and its relationships to non-technical stakeholders, making it easier for them to grasp key insights without needing to understand the underlying mathematics or data science.

The provided examples not only demonstrate the simplicity and effectiveness of our method but also highlight its adaptability and ease of integration into existing data workflows. By following these guidelines, users can leverage the spherical spiral visualization to gain deeper insights into their high-dimensional data, thereby enhancing both the analysis and interpretation processes.

C. Other Applications

In addition to data analysis, Spherical Spirals performance dominance opens the door for applications outside of traditional, slow data analysis. Anything that can process very large datasets in half a second begs to suggest being used in real-time applications. Especially in contexts where full-featured machine learning models would not fit, like on small embedded devices. As datasets can be processed and grown one vector at a time, it also could be a great option for long-running programs. Unfortunately because of the factors described in the disadvantages section, the dataset and feature space must be relatively discrete (unlike the Fashion MNIST dataset, for example). A simple application taking these into mind could be, a small real-time embedded system that scans for individual letters or numbers on items that pass by. We intend to test this idea further in future work.

XIV. Conclusion

This paper introduced a novel approach to the visualization of latent vectors and embedding spaces through the utilization of spherical spirals. This methodology diverges from traditional techniques such as Principal Component Analysis (PCA), offering a simpler, more intuitive framework that may surpass conventional methods in effectiveness for specific data types. By reimagining the visualization of high-dimensional data, our spherical spiral approach allows users to gain a clearer understanding of complex information structures, thereby enhancing data interpretability and facilitating deeper insights into the underlying patterns present in the data.

Our exploration into the visualization of latent vectors and spaces utilizing spherical spirals represents a pivotal advancement in the interpretability of high-dimensional data. This approach provides a more intuitive and effective method for visualizing complex data structures, aiming to unlock new insights and stimulate further research within the field. Future work will delve into integrating this visualization technique with advanced machine learning models to enhance its applicability across a wider array of data types and tackle more complex visualization challenges.

We extend our gratitude to our peers and advisors for their invaluable feedback and support throughout the course of this research project. We also acknowledge the pivotal role of open-source software and public datasets in facilitating our research. Their availability and robustness have been instrumental in the success of this work, allowing us to test and refine our methodology in a comprehensive and effective manner.

References

[1] A. Dadu *et al.*, "Application of aligned-umap to longitudinal biomedical studies," *Patterns*, vol. 4, no. 6, p. 100 741, 2023, ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2023.100741. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389923000818.

[2] H. Zhou, F. Wang, and P. Tao, "T-distributed stochastic neighbor embedding method with the least information loss for macromolecular simulations," *Journal of Chemical Theory and Computation*, vol. 14, no. 11, pp. 5499–5510, 2018, PMID: 30252473. DOI: 10.1021/acs.jctc.8b00652. eprint: https://doi.org/10.1021/acs.jctc.8b00652. [Online]. Available: https://doi.org/10.1021/acs.jctc.8b00652.

[3] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901. DOI: 10.1080/14786440109462720. eprint: https://doi.org/10.1080/14786440109462720. [Online]. Available: https://doi.org/10.1080/14786440109462720.

[4] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms." arXiv: cs.LG/1708.07747 [cs.LG]. (Aug. 28, 2017).

Appendix A
Additional Experiment Results

A. F-MNIST Embedding Spaces

Figures 12, 13, 14, and 15 show embedding spaces from the Fashion MNIST experiment. As discussed in the experiment section, the non-linear methods produce fairly well grouped spaces with complex structures. PCA and Spirals appear to show less structurally complex distributions. PCA appears like the intersection of three planes, while Spirals looks like an ice cream cone with a vertical gradient of grouping.

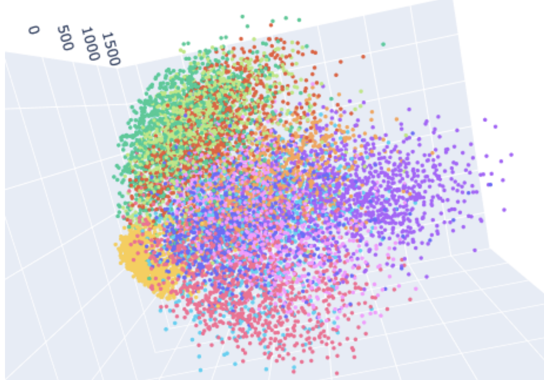

Fig. 12. PCA on F-MNIST Embedding Space

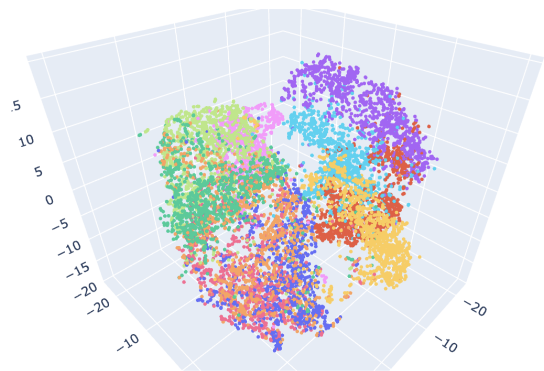

Fig. 13. t-SNE on F-MNIST Embedding Space

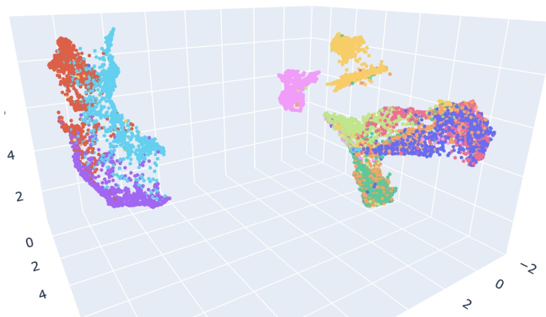

Fig. 14. UMAP on F-MNIST Embedding Space

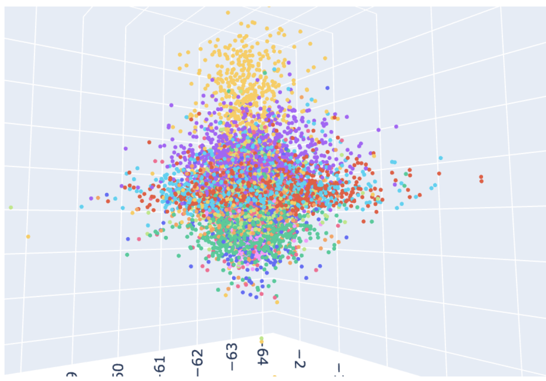

Fig. 15. Spirals on F-MNIST Embedding Space

B. *Plotting of Encoded Shapes*

Figures 19, 17, and 18 show additional plots of embedded encoded shapes, zoomed in to show local areas of similar and dissimilar shapes.

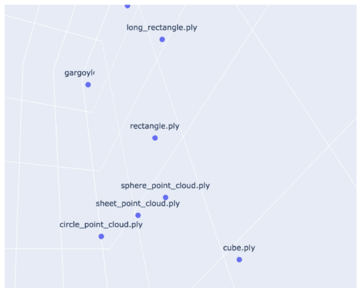

Fig. 16. Very similar shapes are plotted next to each other in the embedding space

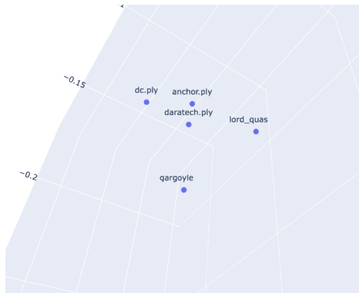

Fig. 17. Very dissimilar shapes are plotted far away from each other in the embedding space

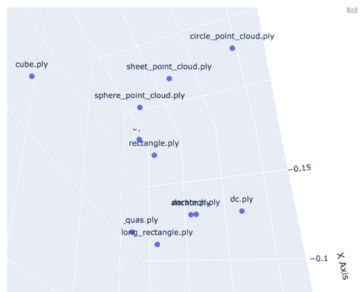

Fig. 18. Zoomed out view showing similar shapes clustering, whilst dissimilar ones far away. Top center vs bottom center

C. *Plotting of Word2Vec embedding space*

Figure 19, zoomed in to show local areas of juxtaposition words in embedding space.

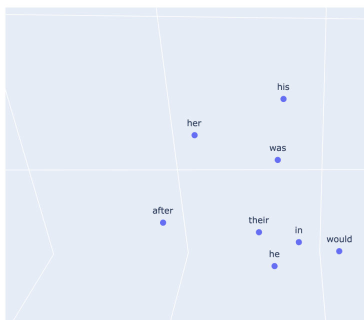

Fig. 19. Words his and her are near to each other, his and he are on the same vertical axis