

# *Honors Project Report*

## *- Motion Projector*

Zelong Jiang

**Abstract**—Roboticians have traditionally compared and visualized robot motions by observing 3D representations. While this method offers direct insights into robot poses, it often requires significant time for users to identify interesting poses and presents challenges in comparing motions at different timestamps. To address these issues, we have developed an interactive visualization tool that leverages modern dimensionality reduction algorithms. This tool empowers roboticists to analyze, explore, and communicate high-dimensional robot joint motion data effectively. Additionally, a diagnostic tool is provided to assess the reliability of the 2D embeddings generated by dimensionality reduction algorithms.

### I. INTRODUCTION

Understanding robot motions is crucial for programming robots or developing planning algorithms. Roboticians can watch a single motion of robots with high-dimensional joint space like humanoid robots or robotics arms using visualization tools such as Rviz. However, understanding multiple motions, including their similarities, difference, or overall trends, can be difficult and time-consuming.

In this paper, we leverage recent advancements in dimensionality reduction technologies to create an interactive visualization tool, Motion Projector <sup>1</sup>, that assists roboticists to efficiently understand multiple motions. Conventionally, people often rely on 3D representations to analyze joint motions effectively. However, when dealing with lengthy robot motions, pinpointing specific poses of interest becomes challenging.

The approach we use to address this issue involves presenting multiple dimension-reduced traces on a 2D graph alongside the corresponding robot poses in a 3D view. When two traces overlap or intersect on the 2D graph, it indicates a high probability that robot motions also exhibit similar behavior. Using the interactive visualization tool, users can select a region on the 2D graph where such overlaps or intersections occur and examine the corresponding robot traces in 3D. This integrated approach provides a comprehensive understanding of the relationship between the 2D scatter plots and the original data, facilitating deeper insights into the motion patterns.

The reduced 2D graph representations alongside 3D robot views can also help users compare robot motions at different timestamps. As robot motion data inherently consist of time-series information, comparing one motion with itself solely through observation of 3D representations can be challenging. However, our tool, which offers a 2D projection of all robot motions at all timestamps, makes it much easier

to compare robot motions across different timestamps by selecting specific regions.

Our input data is the robot motion joint data, represented as time-series data. If the robot moves without teleporting, the input data should be continuous traces. It is expected that the traces after the reduction can be as continuous as possible. However, dimensionality reduction algorithms may lead to discontinuities in the resulting traces. For instance, it is plausible for such algorithms to map two points that are close in the high-dimensional space to distant locations in 2D, or two points that are distant in the high-dimensional space to close locations in 2D. While rectifying these errors entirely is challenging, we have developed a diagnostic tool to detect and pinpoint any inconsistencies that occur. This tool assists users in identifying which segments of the embedding are more reliable, aiding in the interpretation of the visualization.

### II. RELATED WORK

The program builds on Motion Comparator [1] which is an online tool to visualize and compare robot motions. Using dimensionality reduction algorithms, the tool is able to map high dimensional robot joint data into 2D embeddings which are shown in a graph implemented using Plotly.js [2].

#### A. Dimensionality Reduction Algorithms

We experimented with some common dimensionality reduction algorithms that have been used to map high-dimensional trajectories to 2D trajectories for visualization, including Multidimensional scaling (MDS) [3] t-distributed stochastic neighbor embedding (t-SNE) [4], uniform manifold approximation and projection (UMAP) [5], and Parametric UMAP [6]. The default dimensionality reduction algorithm we use in the program is Parametric UMAP because it generates embeddings that exhibit higher continuity compared to other methods. Sainburg [7] also utilized Parametric UMAP successfully to create continuous projections of vocalization data. However, there is a lack of research evaluating the performance of dimensionality reduction algorithms in terms of the continuity of traces post-reduction. In addition, unlike many other dimension reduction algorithms that focus on preserving only the local or only the global structure, Parametric UMAP can balance both local and global structure [8].

#### B. Using Dimensionality Reduction algorithms

Our display devices are physically limited in directly displaying structures with dimensions exceeding two or

<sup>1</sup>Source code: <https://github.com/uwgraphics/RobotJointMotionReduction>

three. It is, therefore, common to visualize high-dimensional data by reducing its dimensionality. However, dimension-reduced data may contain distortions, and visual patterns in the reduced data may not accurately represent patterns in the original high-dimensional space [9]. Consequently, it is crucial to implement features to assess the reliability of the embeddings [9]. Since our tool, Motion Projector, uses dimensionality reduction algorithms, it also provides features that can help users interpret the embeddings.

1) *Preprocessing*: Research has demonstrated that transforming input datasets into a new set of vectors with the same dimensionality can enhance dimensionality reduction performance [10]. We have implemented preprocessing techniques specifically tailored to improve the embeddings of robot motion data, which is time-series data.

2) *Parameter tuning*: Hyperparameters play a significant role in influencing the embeddings generated by various dimensionality reduction algorithms. Appleby [11] trains a neural network to tune the hyper parameters in multiple dimensionality reduction algorithms. Our tool give users many options to tweak the parameters of the dimensionality reduction algorithms. For UMAP, the users can modify the  $n$ -nearest neighbors, minimum distance, spread, and random seed. For Parametric UMAP, the embeddings are calculated based on a neural network.

3) *Interactive tool*: Many interactive tools are built to help the users better understand the embeddings generated by the dimensionality reduction algorithms. Hinterreiter [12] build an interactive tool for reducing trajectory dimension for decision-making paths. Furthermore, according to Xia [13], it is important to enable interaction for dimension reduction tools so that users can tweak the parameters to generate different embeddings. In addition, interactive tools can also help users visualize, understand and explore long time-series data [14]. However, none of these tools specifically address dimensionality reduction for robot motion data. In contrast, our tool offers numerous interactive features designed to assist users in comprehending the reduced robot motion data, as elaborated later in the paper.

### C. Dimensionality Reduction for Robotics

A lot of robotic data is high-dimensional, so roboticists sometimes use dimensionality reduction techniques to visualize the data. Masumori [15] use UMAP to visualize robot memory and actual motion data which has 43 dimensions. UMAP is also used to visualize latent representation of robot surgical scenes [16]. Interaction with UMAP is also helpful as Jaunet [17] build an interaction tool with UMAP to help visualize sim2real gap. However, no existing research on dimension reduction methods specifically target robot joint motion data, which our tool addresses. Understanding robot joint motions is important for roboticists involved in robot programming or planning algorithms.

### D. Visualization for robotics

Roboticists frequently employ visualization techniques to observe and analyze robot motions, aiding them in making

informed decisions regarding robot movement [18] [19]. Our tool utilizes a combination of 2D and 3D views to visualize robot motions, as elaborated later in the paper.

## III. MOTION PROJECTOR

The Motion Projector incorporates many features. It pre-processes the data, offers interactive and diagnostic tools, and empowers users to control parameters and compare motions effectively.

### A. Preprocessing

First, the robot joint motions data is preprocessed to generate better UMAP embeddings. As UMAP calculates  $n$ -nearest neighbors to derive embeddings, filtering and interpolating the data can change the neighboring points and thus improve the embeddings.

1) *Filtering*: Throughout the entire robot motion, there may be brief periods where the robot remains stationary. When the robot is stationary, the data will contain a series of identical consecutive points. As most neighboring points in this scenario are also repetitions of the same point, they tend to be mapped to regions distant from the rest of the traces, despite their proximity in the original high-dimensional space. Our approach is to filter the data by excluding identical consecutive points from the input to UMAP. After UMAP calculates the embeddings, the repetitive points will be assigned the same values as their predecessors. In this way, the data is filtered so that the repetitive points are removed, preventing the aforementioned distortion in the resulting visualization.

2) *Interpolation*: By interpolating the data, the  $n$ -nearest neighbor of any point will be closer to this point compared to the  $n$ -nearest neighbor in the original data. In this way, the points are closer in the embeddings, resulting in less gaps and smoother traces.

### B. Interactive tool

Our tool contains interactive features to help users interpret dimension reduction results.

1) *Schneiderman's Mantra*: Our tool adheres to Schneiderman's mantra, allowing users to first obtain an overview, then zoom in and apply filters, and finally access detailed information on demand [20].

2) *Link to 3D view*: In order to better interpret the result of the 2D embeddings, we also offer two different ways to display these robot poses in 3D views.

a) *Single Scene View*: Place all robot poses within a single scene. Each robot pose is superimposed so that it is easier to detect subtle differences.

b) *Grid-like Window View*: Display these robot poses in a grid-like window view, with each robot pose presented in its own scene. It facilitates pairwise robot poses comparison. In particular, the grid-like window view is especially helpful when the users want to compare the robot pose corresponding to one point on the graph with the robot poses of its neighbors. In this setup, the center window of the grid displays the robot pose corresponding to the selected point,

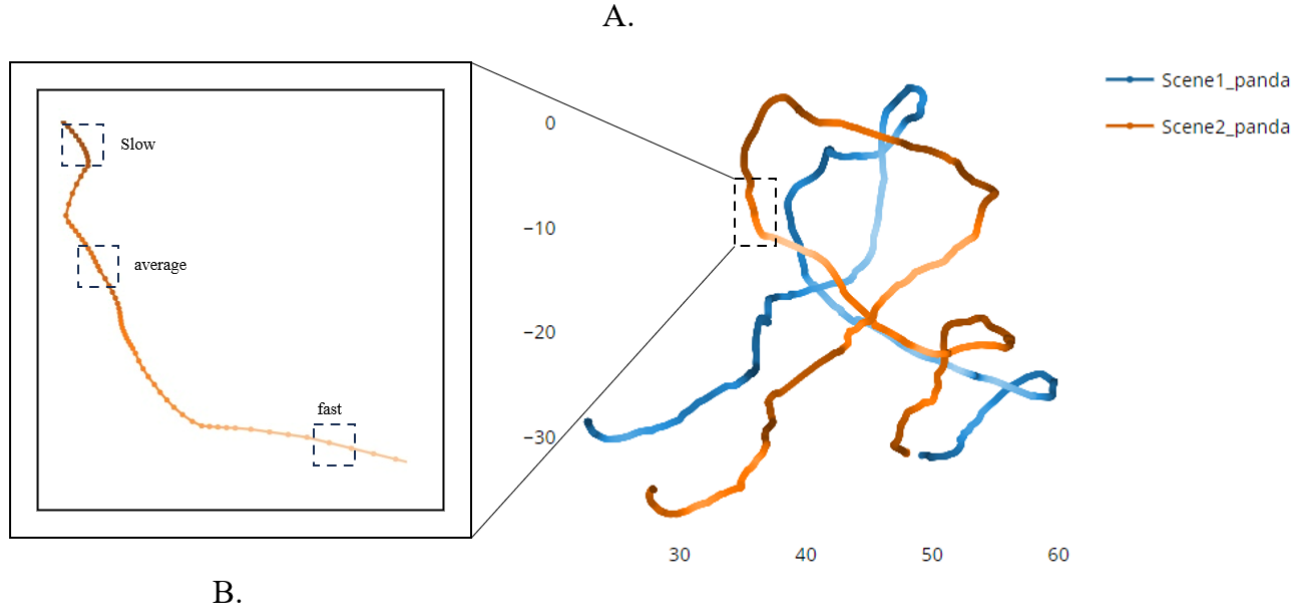


Fig. 1: (A) Motion Projector can visualize the relative velocity of the trajectory in joint space. (B) Segments with lighter color indicate higher relative velocity.

while its surrounding windows display the robot poses of its neighbors.

### C. Enrichment

1) *Velocity*: Velocity is also an important feature that can help users understand robot motions. Therefore, our tool also visualizes the velocity in the joint space on the 2D graph, where each segment is assigned a different lightness corresponding to its relative velocity. Higher relative velocities are represented by lighter colors.

### D. Diagnostic tool

Certain dimension reduction algorithms (such as UMAP, Parametric UMAP, t-SNE, and MDS) we have used cannot preserve the continuity of data post-reduction. This can lead to scenarios where two points closely situated in high-dimensional space are depicted as distant in 2D, or vice versa. The diagnostic tool is implemented to help understand how good/trustworthy a dimensional reduction result is.

1) *Display Gaps*: We define gaps as the spaces between two adjacent points within a single trace in the original high-dimensional space that exhibit considerable distance after dimension reduction. The program allows users to specify the minimum 2D distance threshold that defines a gap and visualizes all such gaps in the 2D graph.

2) *Display Stretches*: We define stretches as pairs of points that are close (with a distance smaller than 0.1) in the original high-dimensional space but exhibit a significant distance after dimension reduction.

3) *Display Folds*: We define folds as pairs of points that are close (with a distance smaller than 0.1) in the 2D embedding but exhibit a significant distance in the original high-dimensional space.

The tool allows users to specify the minimum distance threshold that defines a gap, stretch, or folds and highlights all such gaps, stretches, and folds in the graph, enabling users to identify regions with potentially less reliable data as shown in 3. Adjacent to the graph, a legend displays a sorted list of these anomalies based on their distances, arranged in descending order. By clicking on a specific gap, stretch, or fold in the legend, users can visualize its position within the graph and explore the corresponding robot poses in a 3D view. This functionality aids in determining whether these anomalies are a result of UMAP's dimension reduction process or errors in the original data, such as instances where the robot appears to "teleport". It's important to note that the points of stretch or folds can belong to different traces.

4) *Display N Nearest Neighbors*: Users can select a point in the graph to highlight its  $n$ -nearest neighbors within the visualization. Additionally, employing the  $k$ -medoids algorithm, the tool identifies and displays eight points from the neighborhood, displaying their corresponding robot poses along with the pose of the selected point in separate 3D scenes as mentioned in III-B.2. In the grid window view, the center window will display the robot pose corresponding to the clicked point. Users can toggle between viewing the "before reduction" and "after reduction" states of the neighbors. Given that UMAP operates based on  $n$  nearest neighbors, this functionality allows users to assess the effectiveness of UMAP in preserving the relative distances from a point to its neighbors. By comparing the neighbors before and after reduction, users can evaluate the performance of the dimension reduction process.

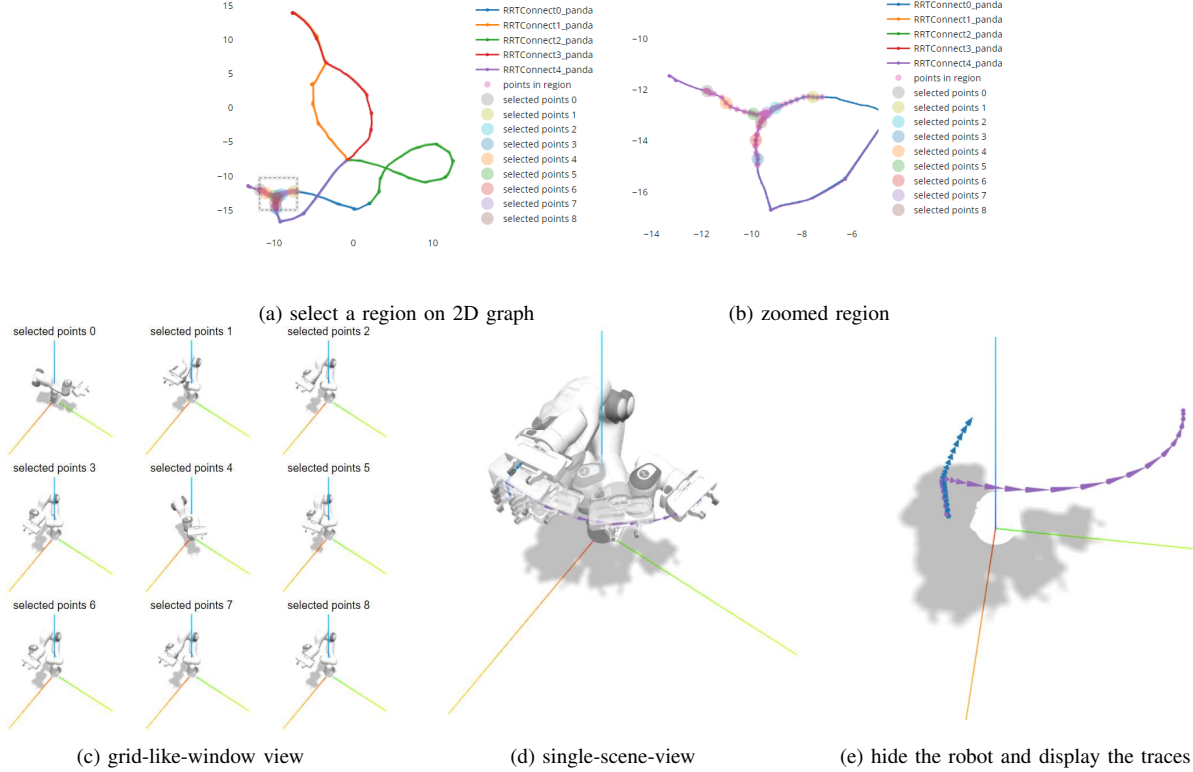


Fig. 2: Motion Projector enables users to select a region on 2D graph (a), selects nine points from all the points in the region using  $k$ -medoids (b), shows the robot poses of these nine points in both nine-window view (c) and one window view (d), and visualizes the traces of all points from the selected region in 3D (e).

### E. Compare Motions

Presently, scientists commonly visualize and contrast multiple robot motions through observation of 3D representations. However, this approach has two limitations. Firstly, due to the potential length of robot motions, users may spend considerable time locating interesting robot motions, such as similar robot poses, by playing back the motions. Additionally, it is also challenging to compare a robot motion with itself at different timestamps using this method.

Our tool addresses both issues by projecting robot motions onto a 2D graph. Within this graph, users can identify intersections or overlaps among multiple traces. Furthermore, since all robot motions are displayed on the same graph, users can easily compare a trace with itself.

However, merely examining the 2D graph may not provide sufficient information. To better visualize the robot motions, our tool can display 3D robot poses alongside the 2D graph. First, users can select a region in the 2D embedding. Subsequently, nine points will be highlighted using the  $k$ -medoids algorithm, and the corresponding robot poses will be displayed in 3D scenes as mentioned in III-B.2. Note that we use  $k$ -medoids instead of  $k$ -means because the points returned by  $k$ -medoids are guaranteed to be part of the input data, whereas points returned by  $k$ -means may not necessarily belong to the input data. In the meantime, users can select a robot joint and examine the robot joint traces

affiliated with the points from the selected region. These features empower users to compare multiple robot motions in 3D, particularly when they observe several traces in close proximity within the 2D embedding as shown in 2.

### F. Important Parameters

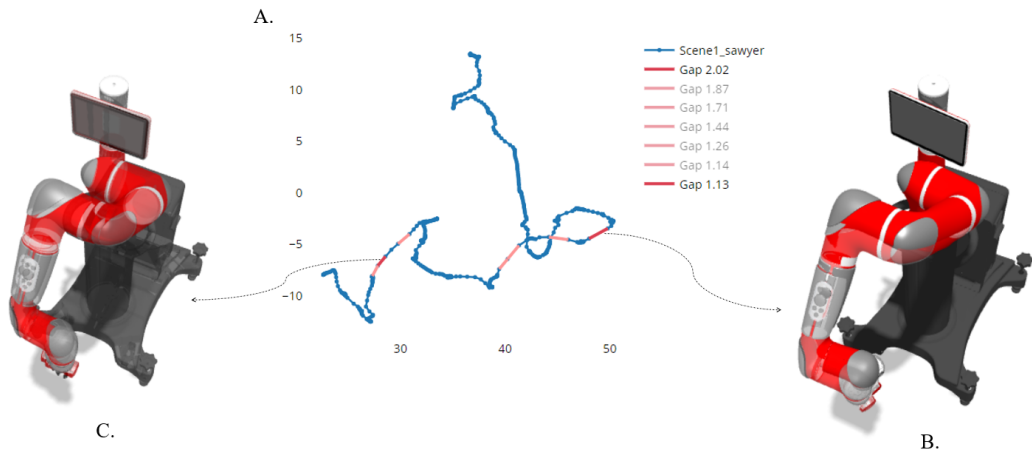
Since we use UMAP as the dimensionality reduction algorithm, adjusting its parameters can impact the embeddings. To provide users with flexibility, we have integrated parameter tuning features into our program, allowing them to modify the parameters and observe different embeddings.

1) *UMAP vs. Parametric UMAP*: While both methods might result in gaps in the traces after reduction, parametric UMAP generally yields smoother traces compared to UMAP. Users have the option to toggle between UMAP and parametric UMAP, with the latter being the default setting.

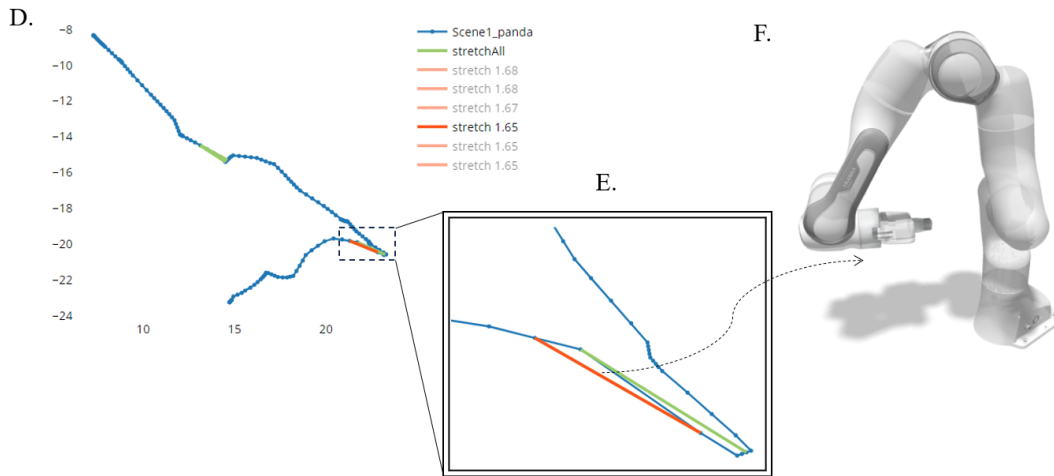
2)  *$n$  nearest neighbors*: The users are able to adjust the number of nearest neighbors for UMAP when computing the embedding. By default, this value is set to 10% of the size of the dataset.

3) *loss weight*: Users can adjust the loss weight parameter to control the balance between Parametric UMAP's preservation of local and global structure.

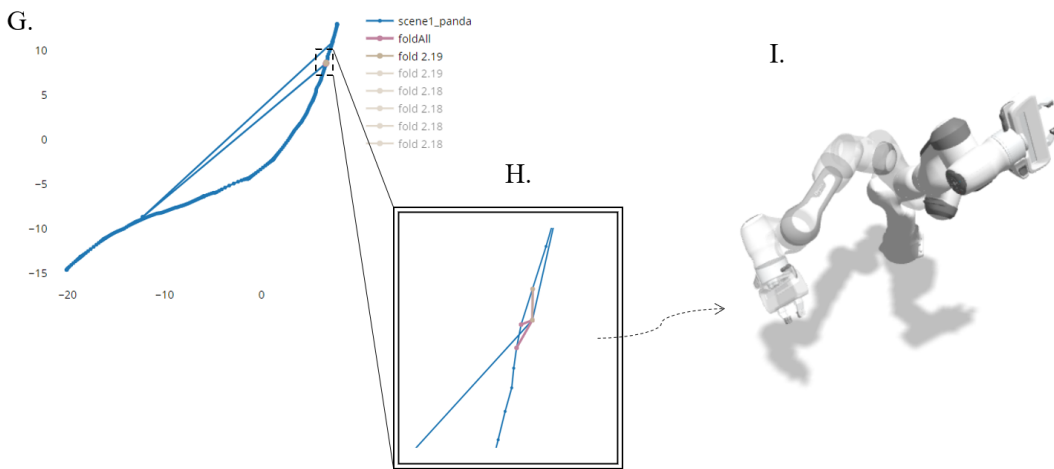
4) *random seed*: Utilizing the same random seed can assist users in reproducing the 2D embeddings generated by UMAP.



(A) Our tool identifies all gaps in the trajectory where distances exceed a threshold specified by the user. (B) After users click on the largest gap on the legend, the gap is highlighted in red on the graph, and the tool displays the corresponding robot poses in 3D. (C) The smallest gap is displayed and it indicates that the robot poses are not so different.



(D) Our tool identifies all stretches in the trajectory where distances exceed a threshold specified by the user. (E) After users click on a stretch on the legend, the stretch is highlighted in red on the graph. (F) The tool displays the corresponding robot poses in 3D. In this example, the two robot poses are not so different but the distance of their reduced points on the graph is big.



(G) Our tool identifies all folds in the trajectory where distances exceed a threshold specified by the user. (H) After users click on a fold on the legend, the fold is highlighted in brown on the graph. (I) The tool displays the corresponding robot poses in 3D. In this example, the two robot poses are very different but the distance of their reduced points on the graph is small.

Fig. 3: The Diagnostic tool can display the gaps, stretches and folds in both 2D and 3D views.

#### IV. CONCLUSION

We introduce a visualization tool, Motion Projector, which uses the dimensionality reduction algorithm to transform high-dimensional robot joint motion data into a 2D representation displayed on a graph. The tool presents all reduced robot motion data across timestamps on the 2D graph, accompanied by 3D views of selected robot poses. In this way, our tool effectively addresses two limitations of conventional robot motion visualization: the time-consuming process of identifying interesting robot poses and the difficulty in comparing robot motions at different timestamps. Due to the limitations of any dimensionality reduction algorithm, the resulting traces may contain distortions, including gaps, stretches, or folds. As such, our tool incorporates a diagnostic feature to visualize these distortions stemming from dimensionality reduction. This diagnostic tool enables users to evaluate the reliability of various regions of the graph.

#### V. DISCUSSION

Our tool has a number of limitations that requires future research and development. Further work could be done to utilize the background color of the graph to show more information, such as the stress and strains of the data. Furthermore, even though a diagnostic tool is created to evaluate the trustworthiness of the embeddings after dimensionality reduction, we do not attempt to create a dimensionality reduction algorithm that can preserve the continuity of the traces. In addition, the joint motion data can be preprocessed by assigning a weight to each joint to map joint configurations to a more meaningful high-dimensional space. Moreover, we want to add some background points around the traces to explore the joint space not used by the trace, but these background points might affect the embedding of the UMAP since they are also the neighbors of the points on the traces.

#### VI. ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Yeping Wang, Connor Bailey, and Michael Gleicher for their invaluable feedback and assistance throughout the course of my research project. Their insightful suggestions and guidance significantly contributed to the completion and success of this work.

I am truly grateful for their dedication, expertise, and unwavering support. Their mentorship has been instrumental in shaping this project and enhancing its quality. I am fortunate to have had the opportunity to collaborate with such outstanding individuals.

#### REFERENCES

- [1] Y. Wang, A. Peseckis, Z. Jiang, and M. Gleicher, "Motion comparator: Visual comparison of robot motions."
- [2] P. T. Inc. (2015) Collaborative data science. Montreal, QC. [Online]. Available: <https://plot.ly>
- [3] M. C. Hout, M. H. Papesh, and S. D. Goldinger, "Multidimensional scaling," *Wiley Interdiscip Rev Cogn Sci*, vol. 4, no. 1, pp. 93–103, 2013.
- [4] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [5] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.
- [6] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric umap embeddings for representation and semisupervised learning," *Neural Computation*, vol. 33, no. 11, pp. 2881–2907, 2021.
- [7] T. Sainburg, M. Thielk, and T. Q. Gentner, "Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires," *PLoS computational biology*, vol. 16, no. 10, p. e1008228, 2020.
- [8] H. Jeon, H.-K. Ko, S. Lee, J. Jo, and J. Seo, "Uniform manifold approximation with two-phase optimization," in *2022 IEEE Visualization and Visual Analytics (VIS)*. IEEE, 2022, pp. 80–84.
- [9] L. G. Nonato and M. Aupetit, "Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2650–2673, 2018.
- [10] B. Rafieian, P. Hermosilla, and P.-P. Vázquez, "Improving dimensionality reduction projections for data visualization," *Applied Sciences*, vol. 13, no. 17, p. 9967, 2023.
- [11] G. Appleby, M. Espadoto, R. Chen, S. Goree, A. C. Telea, E. W. Anderson, and R. Chang, "Hypermp: Interactive visual exploration of multidimensional projection hyperparameters," in *Computer Graphics Forum*, vol. 41, no. 3. Wiley Online Library, 2022, pp. 169–181.
- [12] A. Hinterreiter, C. Steinparz, M. Schöfl, H. Stütz, and M. Streit, "Projection path explorer: exploring visual patterns in projected decision-making paths," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 3–4, pp. 1–29, 2021.
- [13] J. Xia, L. Huang, W. Lin, X. Zhao, J. Wu, Y. Chen, Y. Zhao, and W. Chen, "Interactive visual cluster analysis by contrastive dimensionality reduction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 734–744, 2022.
- [14] M. Ali, M. W. Jones, X. Xie, and M. Williams, "Timecluster: dimension reduction applied to temporal data for visual analytics," *The Visual Computer*, vol. 35, no. 6–8, pp. 1013–1026, 2019.
- [15] A. Masumori, N. Maruyama, and T. Ikegami, "Personogenesis through imitating human behavior in a humanoid robot "alter3"," *Frontiers in Robotics and AI*, vol. 7, p. 532375, 2021.
- [16] J. Y. Wu, A. Tamhane, P. Kazanzides, and M. Unberath, "Cross-modal self-supervised representation learning for gesture and skill recognition in robotic surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, pp. 779–787, 2021.
- [17] T. Jaunet, G. Bono, R. Vuillemot, and C. Wolf, "Sim2realviz: Visualizing the sim2real gap in robot ego-pose estimation," *arXiv preprint arXiv:2109.11801*, 2021.
- [18] A. Orthey, B. Frész, and M. Toussaint, "Motion planning explorer: Visualizing local minima using a local-minima tree," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 346–353, 2019.
- [19] D. Hägele, M. Abdelaal, O. S. Oguz, M. Toussaint, and D. Weiskopf, "Visual analytics for nonlinear programming in robot motion planning," *Journal of Visualization*, pp. 1–15, 2022.
- [20] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The craft of information visualization*. Elsevier, 2003, pp. 364–371.