Appendix

1. Graph Convolution

Algorithm 1 shows the graph convolution operation.

Algorithm 1: Graph convolution.

Input: node embedding $\mathbf{F}_{u,j}$ at layer j, a set of neighbor embeddings $\{\mathbf{F}_v | v \in \mathcal{N}(u)\}$, a set of neighbor weights $\mathbf{W}$ and bias b, an activation function $\gamma(\cdot)$, an aggregator $\alpha(\cdot)$, and a norm function $N(\cdot)$.

Output: node embedding $\mathbf{F}_{u,j+1}$ at layer $j+1$.

$\mathbf{F}_{u,j+1} \leftarrow \gamma(\{\mathbf{W}\mathbf{F}_{v,j} + b) | v \in \{\mathcal{N}(u), u\}\})$

$\mathbf{F}_{u,j+1} \leftarrow \alpha(\mathbf{F}_{u,j+1})$

$\mathbf{F}_{u,j+1} \leftarrow \mathbf{F}_{u,j+1} / N(\mathbf{F}_{u,j+1})$

2. Additional Results

In Figure 1, we show more results with different data sets using SurfNet for stream surface clustering. Each column shows node clustering results for different surfaces of the same data set. These results further confirm the reliability of SurfNet in clustering the nodes of stream surfaces. We found that for complex flows (e.g., solar plume), the clustering results generated by SurfNet include some minor errors, but the main surface structures can be detected correctly.

3. Dimensionality Reduction and Clustering

Dimensionality reduction. To transform the learned features into a 2D space, we experiment with four dimensionality reduction methods: t-SNE [vdMH08], UAMP [MHM18], MDS [Kru64], and Isomap [TdSL00]. Among them, t-SNE is a neighborhood-preserving method, UMAP is a global-preserving method, and MDS and Isomap are distance-preserving methods. The brushing and linking results are shown in Figure 2. Both MDS and Isomap cannot group similar nodes in the 2D projection space, while t-SNE and UMAP meet our expectations. In addition, in the t-SNE projection, the nodes are more separated than those in the UMAP projection. Therefore, we choose t-SNE as the dimensionality reduction algorithm to project the learned features.

Clustering. To group the points in the 2D projection space, we study three representative clustering algorithms: DBSCAN (density-based), k-means (partition-based), and agglomerative clustering (hierarchy-based). The clustering results are shown in Figure 3. DBSCAN performs best by detecting the lobster's claws, body, and tail while the other two clustering algorithms mix these structures.

Feature space vs. projection space. We also conduct a study to justify clustering the node embeddings in the projection space (i.e., the space generated by t-SNE) instead of the feature space. As shown in Figure 4, the results of clustering nodes directly in the feature space do not make sense, and only several isolated parts are grouped. The possible reason for this unsatisfactory result is that in the feature space, data are sparse, which is problematic for any method requiring statistical significance. Moreover, these data are dissimilar in many ways, preventing the traditional clustering algorithm from working efficiently. However, by clustering the node embeddings in the t-SNE space, we keep all important information and decompose co-related factors, ensuring that the clustering algorithm works well.

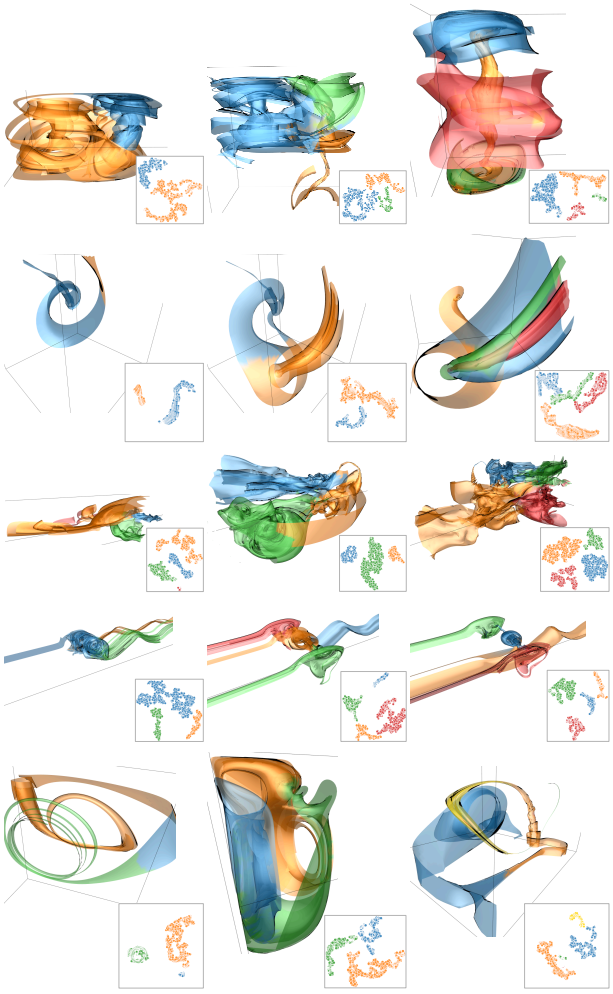


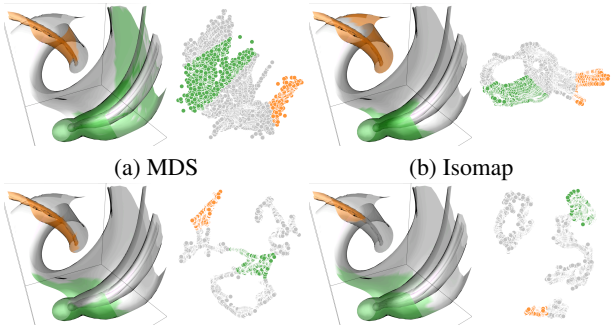
Figure 1: *SurfNet node clustering results of a stream surface. Top to bottom: Bénard flow, five critical points, solar plume, square cylinder, and two swirls.*

Therefore, in this paper, we choose the combination of t-SNE for dimensionality reduction and DBSCAN for node clustering and perform clustering of node embeddings in the t-SNE space.

4. SurfNet Analysis

To evaluate SurfNet, we analyze the following hyperparameter settings: mesh simplification, network depth, training stability, embedding strategy, embedding dimension, feature initialization, and training samples. We focus on node clustering results, from which surface embedding (representative selection) results are derived. For fair comparisons, all remaining parameters use the same settings (e.g., the number of clusters).

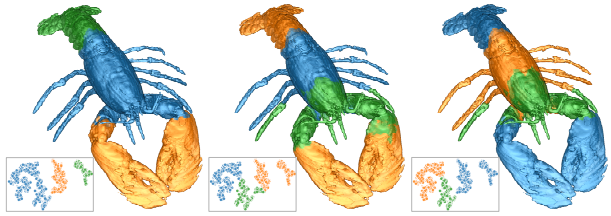
Mesh simplification. To study how mesh simplification impacts the quality of node clustering results, we apply different mesh simplification thresholds ε [GH97] (larger ε leads to more simplifica-



(a) MDS (b) Isomap

(c) UMAP (d) t-SNE

Figure 2: *Comparison of different dimensionality reduction methods via brushing and linking using the five critical points data set. The unselected nodes are colored in gray.*



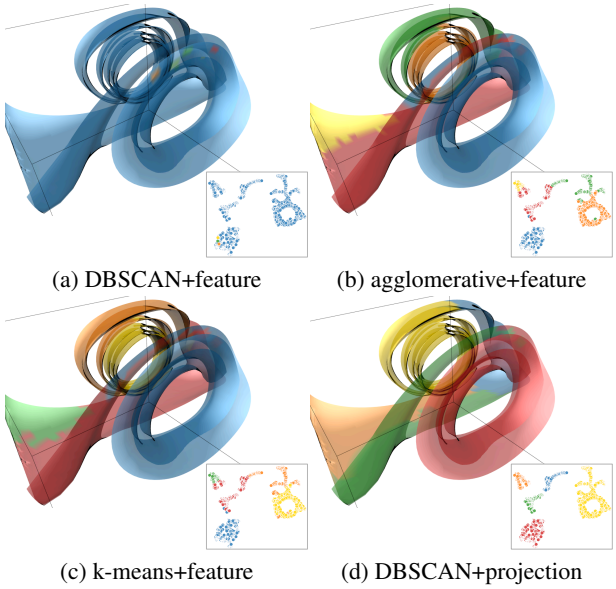
(a) DBSCAN (b) k-means (c) agglomerative

Figure 3: *Comparison of different clustering algorithms under t-SNE projection using the lobster data set. Each produces* 3 *clusters.*

tion) to simplify stream surfaces. We then use the simplified surfaces to train SurfNet using the two swirls data set. The results are shown in Figure 5. Under $\varepsilon = 1.0$, SurfNet cannot detect the bridge that connects the left and right spirals. In addition, some blue surface patches are mixed with the orange ones. Under $\varepsilon = 0.5$ and $\varepsilon = 0.2$, the clustering results have no significant differences. Both can separate the three spirals and the bridge well. Note that all the surfaces displayed in the figures are the original surface without simplification. Simplified surfaces are only used for SurfNet training. Hence, we suggest applying $\varepsilon = 0.5$ to simplify the surfaces.

Network depth. Given a mesh simplification threshold ε (i.e., 0.5), we evaluate how the network depth affects the performance of node embedding. We use 3, 6, and 9 layers to optimize SurfNet using the two swirls data sets. The node clustering results are displayed in Figure 6. Only using 3 layers cannot produce meaningful node embeddings since it groups one swirl and its bridge into one cluster. However, applying 6 and 9 layers can discover the two swirls (i.e., the blue and red parts) and two bridges (i.e., the orange and green parts). Therefore, under $\varepsilon = 0.5$, using SurfNet with 6 layers is sufficient to generate meaningful node embeddings.

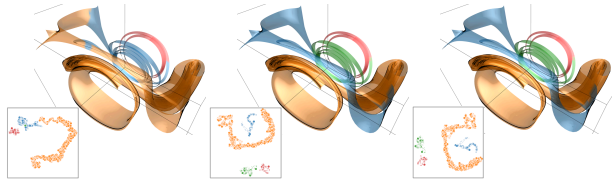
Training stability. To investigate the stability and sensitivity of the training process, we independently train SurfNet four times. The parameters are randomly initialized using He et al. [HZRS15]. The results are displayed in Figure 7. There is no significant difference among these results, and all results can discover the bonsai and its basin. Note that the t-SNE projections of the four results are not similar due to random initialization of the 2D points in the t-SNE algorithm. However, the relative positions of these points



(a) DBSCAN+feature (b) agglomerative+feature

(c) k-means+feature (d) DBSCAN+projection

Figure 4: *Comparison of different clustering algorithms under different spaces (feature vs. projection) using the two swirls data set.*

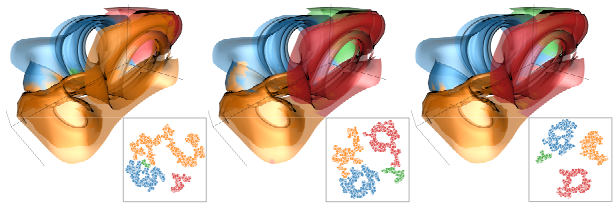


(a) $\varepsilon = 1.0$ (b) $\varepsilon = 0.5$ (c) $\varepsilon = 0.2$

Figure 5: *SurfNet node clustering results of a stream surface under different thresholds for mesh simplification using the two swirls data set.*

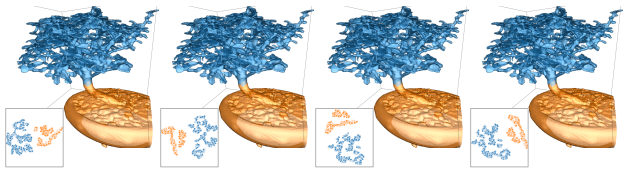
are consistent. Therefore, SurfNet is stable in terms of training and node clustering.

Embedding strategy. To confirm the effectiveness of generating node embeddings using SurfNet, we compare the clustering results generated using node embedding with SurfNet and directly using the shortest-path length as the distance measure, as shown in Figure 8. The clustering results of SurfNet outperform those of direct embedding as the number of clusters increases. One possible explanation is that the features generated by SurfNet include Euclidean



(a) 3 layers (b) 6 layers (c) 9 layers

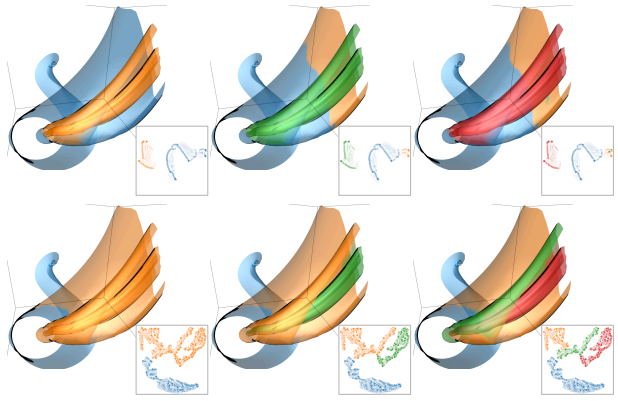
Figure 6: *SurfNet node clustering results of a stream surface under different network depths using the two swirls data set.*



(a) (b) (c) (d)

Figure 7: *SurfNet node clustering results of an isosurface using the bonsai data set. (a) to (d) show the node clustering results from SurfNet trained four times independently.*

information (i.e., node position) and geodesic information (i.e., loss optimization). The former can offer cues between nearby nodes, while the latter can help separate spatially close nodes with large geodesic distances [HBS*21]. Therefore, the embedding generated by SurfNet better captures the surface structure and leads to better clustering results.

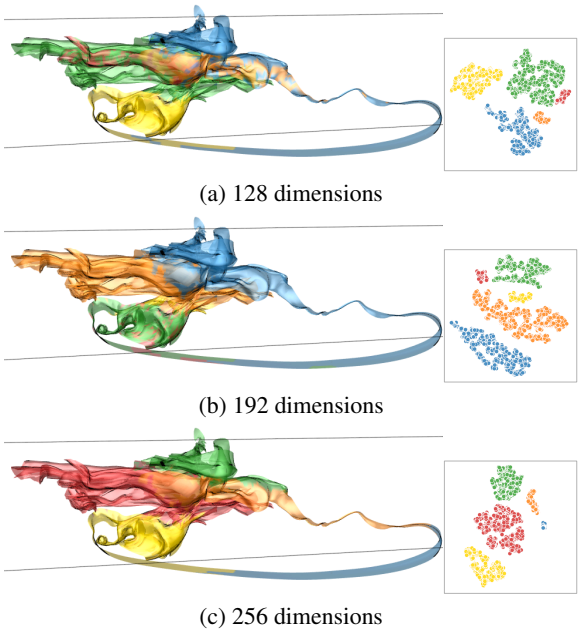


(a) 2 clusters (b) 3 clusters (c) 4 clusters

Figure 8: *Direct embedding and SurfNet embedding of a stream surface under t-SNE projection using the five critical points data set. Top: direct embedding, and bottom: SurfNet embedding.*

Embedding dimension. To determine the appropriate node embedding dimension, we set it to 128, 192, and 256, respectively, to train SurfNet. The node clustering results are shown in Figure 9. The clustering results are not satisfactory under 128 and 192 embedding dimensions. For example, under 128 dimensions, green, red, and orange parts are mixed, and under 192 dimensions, green and red parts cannot be separated well. However, using 256 embedding dimensions, all the structures are clearly separated. In addition, we also use 384 and 512 dimensions to embed node information; however, there is no significant difference compared to the result of 256. Therefore, we set the node embedding dimension to 256 for SurfNet.

Node information initialization. To investigate how to initialize the input surfaces' features, we train SurfNet with different feature initialization options (e.g., position, normal, velocity) using the square cylinder data set. As shown in Figure 10, we display the surface clustering results under different feature initializations. Leveraging position as input features, SurfNet achieves the best clustering performance. For normal, SurfNet cannot separate meaningful



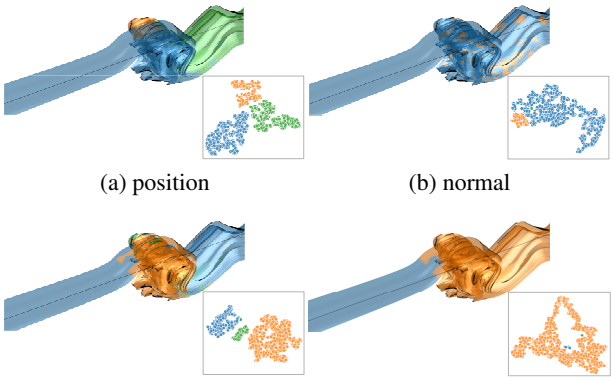
(a) 128 dimensions

(b) 192 dimensions

(c) 256 dimensions

Figure 9: *SurfNet node clustering results of a stream surface under different numbers of embedding dimensions using the solar plume data set.*

parts. For velocity, SurfNet detects major structures but still misclassifies some of the orange parts into the green part, as shown in Figure 10 (c). In addition, if we use both position and velocity as input features, the clustering quality does not improve. In summary, using normal, velocity, or position+velocity as the initialized features does not lead to good node clustering results. This is because SurfNet is optimized using the shortest-path loss. Only position-related information can provide an appropriate initialization for SurfNet optimization. Hence, we suggest only using position as the initialized feature for SurfNet training.



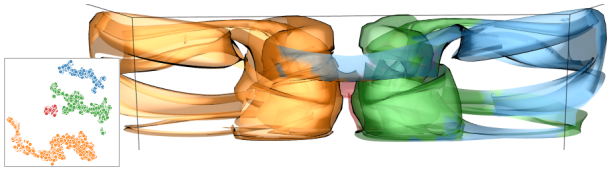
(a) position (b) normal

(c) velocity (d) position+velocity

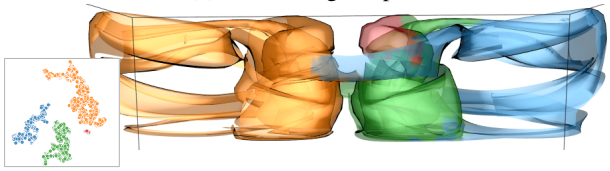
Figure 10: *SurfNet node clustering results of a stream surface under different feature initializations using the square cylinder data set.*

Training samples. We evaluate the influence of the number of

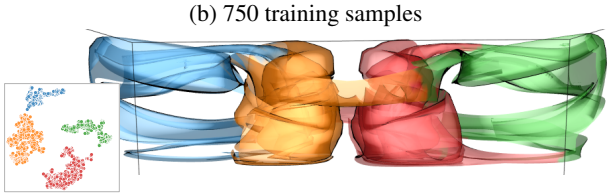
training samples on clustering quality using the Bénard flow data set. We use 500, 750, and 1,000 training samples to train SurfNet. The clustering results are shown in Figure 11. It is clear that using 500 or 750 training samples, SurfNet cannot detect the four major patterns of the stream surface. However, with 1,000 training samples, these patterns can be discovered. Besides, our experiment shows that using more than 1,000 training samples does not further improve clustering quality. Hence, we suggest using 1,000 stream surfaces for training.



(a) 500 training samples



(b) 750 training samples



(c) 1,000 training samples

Figure 11: *SurfNet node clustering results of a stream surface under different numbers of training samples using the Bénard flow data set.*

References

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of ACM SIGGRAPH Conference* (1997), pp. 209–216. 1

[HBS*21] HU Z., BAI X., SHANG J., ZHANG R., DONG J., WANG X., SUN G., FU H., TAI C.-L.: VMNet: Voxel-mesh network for geodesic-aware 3D semantic segmentation. In *Proceedings of International Conference on Computer Vision* (2021). 3

[HZRS15] HE K., ZHANG X., REN S., SUN J.: Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of IEEE International Conference on Computer Vision* (2015), pp. 1026–1034. 2

[Kru64] KRUSKAL J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27. 1

[MHM18] MCINNES L., HEALY J., MELVILLE J.: UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). 1

[TdSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323. 1

[vdMH08] VAN DER MAATEN L. J. P., HINTON G. E.: Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. 1