Winding Numbers on Graphs

Boyang Liu*
liubyfly@bu.edu
Boston University

Junting Lyu*
lvjt@bu.edu
Boston University

ACM Reference Format:
Boyang Liu and Junting Lyu. 2024. Winding Numbers on Graphs. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM STATEMENT

Spacial segmentation can be performed to cluster data in many different datasets after embedding the data into points in high dimensional spaces. In this project, we aim to generalize the use of winding number into multi-dimensional graphs, utilizing it's nature of segmenting spacial regions, and introducing it as a new feature of spacial segmentation task. Our basic approach is to identify graph data whose structure mimics traditional geometric boundary data, and perform segmentation tasks leveraging the winding number. More specifically, we performed semi-supervised learning on a partially labeled graph with multi-dimensional winding number features generated by different strategies of assigning stroke(i.e. connections between oppositely labelled objects in some learned embedding space) orientations. With different assignments on orientations of strokes, we obtained various topology structures on the graph, which gives us multiple winding number features for each point on the graph. Finally, we performed multi-region segmentation on an embedding space and test winding number's reliability as a feature to predict with.

2 BACKGROUND

The traditional concept of winding number is a signed, integer-valued property of a point p with respect to a closed curve Γ that can be calculated with respect to oriented boundary information. Intuitively, it signifies the number of times Γ wraps around p , which makes it valuable for segmentation purposes, notably in distinguishing the interior and exterior of objects. In previous works, the use of winding number has been extended to discrete triangle meshes[5], and point clouds[1]. The definition of winding number has been further generalized to harmonic function u with jump conditions across Γ [3]. These previous works give us intuitions on generalizing winding number to general graphs, using harmonic function definition with jump conditions specified by strokes (Section 3.2).

In this project we have broadened the application of the geometric winding number to general graphs, utilizing its property to split space and serve as a feature in spacial segmentation problems.

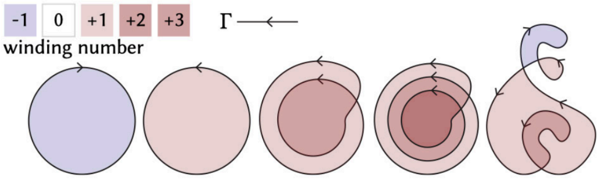

Figure 1: Traditional definition of winding number that separates the 2D space into regions with labels of integer values.[2]

Spacial segmentation in an embedding space is a technique that is commonly used in machine learning and data analysis tasks. Firstly, it requires embedding the original data into a high-dimensional space based on their features or attributes, commonly with deep-learning based methods. Then the space is segmented using various clustering algorithms, i.e. K-means. This segmentation process captures the similarity between the original data through their representation in the embedding space. In our work, the embedding procedure is performed by using Open-clip model [4], in which the labels of the original data are captured by the spacial positions of the points in the embedding space. Furthermore, we built graphs with KNN method and computed the generalized winding numbers based on stroke directions, and used them as features for model training.

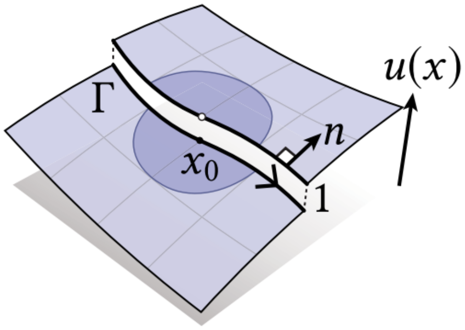

Figure 2: Winding number defined by harmonic function $\Delta u = 0$ on $M \backslash \Gamma$ with specific jump condition $u^+ - u^- = 1$ on Γ, and ensuring continuity with $\partial u^+/\partial n = \partial u^-/\partial n$, on Γ. [3]

3 METHOD

The winding number serves as a topological invariant quantifying the number of times a closed curve winds around a point in a given space, providing crucial information about the topology and connectivity of the underlying structure. In this work, we utilize the

[*] Both authors contributed equally to this research.

topological characteristics inherent in a graph, specifically focusing on winding numbers, to elucidate relationships among points as features. Utilizing the extracted topological features, we further perform clustering and subsequently predict labels for unlabeled points within the graph.

3.1 Graph Generation

We initiate the process with the generation of the graph, focusing on the systematic construction of nodes and edges to represent the underlying data. In order to consider both the randomness and validity of the generated graph, we utilized classical image-text datasets such as MNIST. We employed Open-clip [4] to encode images into high-dimensional embedding vectors. Next, we implement the KNN algorithm on the embeddings and connect each vertex to its k nearest neighbors. Generating the graph in this manner ensures the establishment of topological connections among relatively more similar vertices.

3.2 Topological Features Calculation

The subsequent phase delves into the computation of topological features, a process that can be understood as dimensionality reduction applied to embedding vectors.

Let $G = (V, E)$ be a graph, where V represents the set of vertices and E the set of edges. Let each vertex $v \in V$ have an associated label denoted by $l(v)$, where $l : V \to \mathcal{L}$ is a labeling function mapping vertices to their labels in the set $\mathcal{L}$.

Define the stroke set $S \subseteq E$ as the set of edges $\{s = (u, v) \in E \mid l(u) \neq l(v)\}$, that is, S consists of all edges connecting vertices with different labels. Each edge in this set is referred to as a stroke.

Then we denote the winding number of vertex v_i as w_i. In this work, we emphasize the differences in winding numbers between connected vertices, which allows us to better understand the relationships and similarities among points.

Building upon this premise, we now formalize our approach through the following mathematical definitions:

Suppose Δ is defined as the Laplacian of the winding number at each vertex, we can calculate the winding numbers w by solving the linear system:

$$\begin{cases} \Delta w = 0 \\ w_i - w_j = \sigma \quad \forall e_{ij} \in Stroke \end{cases}$$

where the weights of points in the Laplacian matrix are specified with Gaussian similarity: $\exp\left(-\frac{\|x-x'\|^2}{2s^2}\right)$. And σ is equal to 1 or -1, and the sign is determined by the direction of the stroke between the two labels.

For example, if we define the stroke from label l to label l' as positive, then for any vertices v_i, v_j in the graph, if w_i, w_j correspond to the labels l, l' respectively, the difference $w_i - w_j$ is consistently fixed at 1 or -1 according to the predefined orientation of the stroke.

Then we define the total variance TV as:

$$TV = \sum_{e_{ij} \in E \setminus S} |w_i - w_j| \tag{1}$$

Equation 3.2 quantifies the total topological variance within the same label groups in a graph by summing the absolute differences in winding numbers between vertices connected by edges not classified as strokes. Essentially, it focuses on the internal topological consistency among vertices sharing labels, ignoring the topological differences that define the boundaries between different label groups. This metric highlights the homogeneity or heterogeneity within clusters, shedding light on the graph's underlying structural coherence.

Given that Equation 3.2 excludes strokes, focusing solely on edges that connect vertices of identical labels or with unlabeled vertices, the theoretical optimal value of TV is postulated to be 0. This is predicated on the assumption that vertices sharing the same label or those unlabeled should exhibit identical winding numbers. Given that the value of σ within system 3.2 is determined by the predefined direction of the stroke, it follows that variations in σ can lead to different values of winding numbers and TV. This highlights the sensitivity of TV to the directional attributes assigned to strokes between labels.

For each pair of labels (l_1, l_2) with $l_1, l_2 \in \mathcal{L}$ and $l_1 \neq l_2$, we define two distinct directions: from l_1 to l_2 and from l_2 to l_1. Assuming there are $|\mathcal{L}|$ distinct labels in total, there are $2^{\binom{|\mathcal{L}|}{2}}$ such combinations. We sample a fixed number of these combinations. For each combination, we calculate the total variance TV.

Since we have demonstrated that a smaller TV is preferable, indicating that a lower TV more accurately captures the characteristic topological information of a point, we opt to select the minimum m TV values as representative features of the point. Suppose $TV_{i1}, TV_{i2}, \ldots, TV_{im}$ are the m minimum candidates in the sampled TV values; $w_i(v)$ is the corresponding winding number of vertex v to TV_i. Then, the feature of vertex v f_v is defined as:

$$f_v = \begin{pmatrix} w_{i1}(v) \\ w_{i2}(v) \\ \vdots \\ w_{im}(v) \end{pmatrix}$$

3.3 Clustering and Label Prediction

Based on the point features $F = f_v | v \in V$, we employed the K-means clustering algorithm to assign labels to unlabeled points. K-means is originally designed for unsupervised learning. We modified and designed semi-supervised K-means algorithm to accommodate our semi-supervised problem by incorporating the following adjustments: (1) initializing centroids using the labeled points; (2) fixing the labels of these points while only updating the labels of the unlabeled points. Algorithm 1 provides a detailed description of the semi-supervised K-means clustering algorithm.

4 RESULTS

4.1 Visual Results

Figure 3 displays the results of the winding number calculations. In this figure, the cyan arrows represent strokes, which consistently point from lighter to darker nodes. The direction of the arrows, from higher to lower winding numbers, validates the accuracy of the method.

Figure 4 presents the label prediction outcomes corresponding to those shown in Figure 3. During our experiments, we observed that the accuracy of predictions is unstable in generated simple

Algorithm 1 Semi-supervised K-means Clustering

```
Input: Set of points $P$.
Output: $P$ (with updated $p.predicted\_label$).
procedure SemiSupervisedKmeans($P$)
    Identify labeled points $L = \{p \in P \mid p.\text{labeled} = \text{true}\}$
    Group points in $L$ into clusters $C = \{c_1, c_2, \ldots, c_k\}$ based on their labels, such that all points in $c_i$ have the same label $l_i$.
    Initialize centroids $M = \{\mu_1, \mu_2, \ldots, \mu_k\}$ where $\mu_i.feature$ is the mean of the features of points in $c_i$ and $\mu_i.label = l_i$, which is the label of $c_i$.
    for iterations do
        for each point $p \in P \backslash L$ do
            Compute $dis(\mu_i.feature,\ p.feature)$ for $\mu_i$ in $C$.
            Suppose $\mu_j$ is the nearest centroid.
            $p.predicted\_label \leftarrow \mu_j.label.$
            Update clusters.
        end for
        for $c_i$ in $C$ do
            $\mu_i.feature \leftarrow \text{mean}(\{p.feature \mid p \in c_i\})$
        end for
    end for
    return $P$ (with updated $p.predicted\_feature$)
end procedure
```

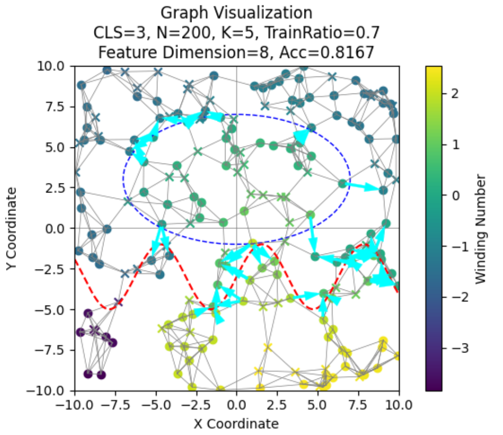

Figure 3: Cyan arrows represent strokes. Each stroke points from points with larger winding numbers (lighter colors) to points with smaller winding numbers (darker colors).

graphs with few points, varying between 0.6 and 0.9. However, the accuracy typically falls within the range of 0.75 to 0.8.

We visualized the winding numbers and features on the graph to demonstrate the accuracy of our winding number calculations, as shown in Figure 5.

We also experimented with various variables: the number of categories, as illustrated in Figures 6 and 7; training rates, shown in Figures 8 and 9; and the number of points, depicted in Figures 10 and 11.

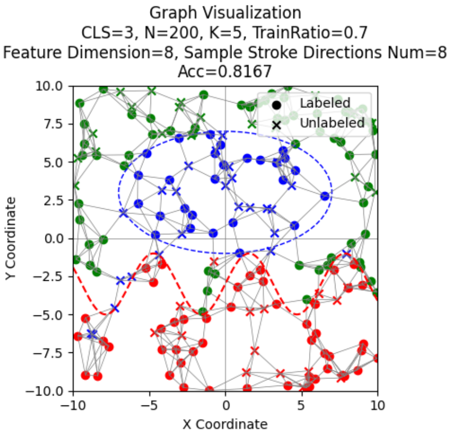

Figure 4: Results for generated simple graphs with complex boundaries and three categories

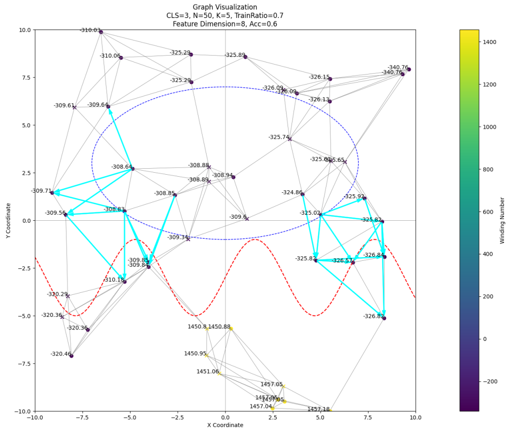

Figure 5: We can visualize the winding number with the smallest total variance. We can also visualize the final feature vectors. It may result in chaos due to the large number of points and long vectors involved; therefore, we do not display it here.

The accuracy in simpler graphs is not stable. We have randomly selected some results that are representative or typical for display. We will discuss these specific results in detail in the forthcoming sections on ablation studies.

4.2 Ablation Studies

We experimented on the MNIST dataset which has 10 different labels with various variables:

- n: the number of categories
- knn_k: the number of edges of each point
- train_ratio: the ratio of labeled training points and unlabeled testing points

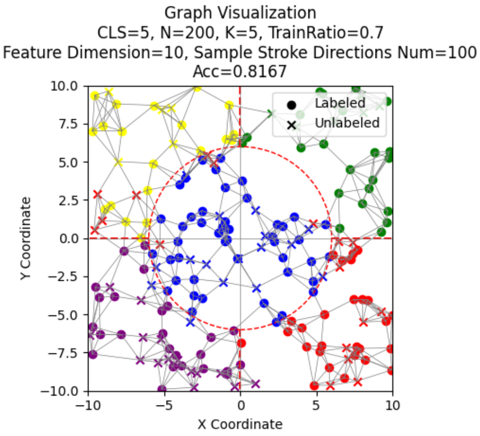

Figure 6: 5 categories

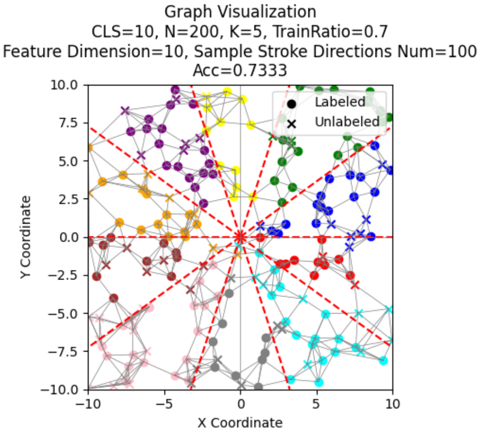

Figure 7: 10 categories

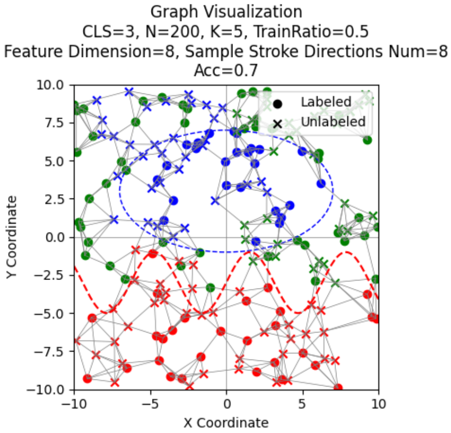

Figure 8: 50% of points are labeled (trained)

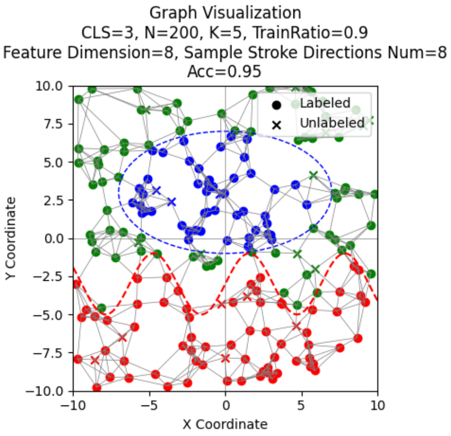

Figure 9: 90% of points are labeled (trained)

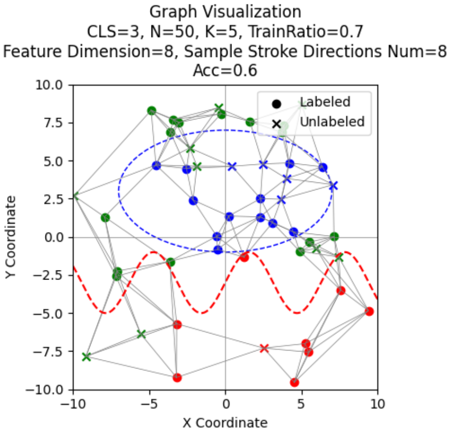

Figure 10: 50 points

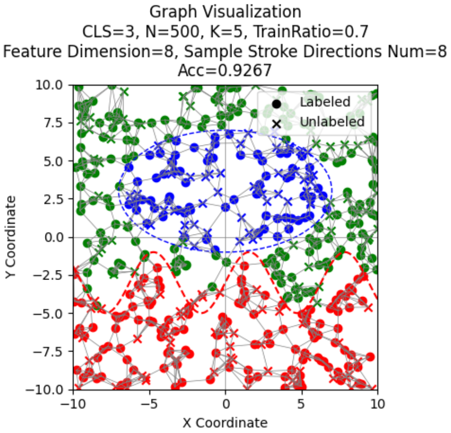

Figure 11: 500 points

- sample_n: the number of sampled stroke directions
- ft_d: feature dimension — the number of minimal winding numbers used to form the feature.

Table 1 displays the results of the ablation studies. We conducted each variable combination trial three times and calculated the average accuracy to mitigate the effects of instability. The relationships between each variable and accuracy are as follows:

- n: Larger number of points gurantees sufficient strokes along boundaries, leading to higher accuracy.
- knn_k: The influence is not very explicit.
- train_ratio: Aligns with intuition; accuracy increases as the training ratio increases.
- sample_n: Accuracy decreases with the increase of the number of sampled stroke directions. Noticing that there exists same total variances, we speculate that with a fixed feature dimension, increased sampling might lead to unrepresentative features with the same winding numbers.
- ft_d: Accuracy increases with the increase of the feature dimension.

Furthermore, we executed a final test with the following parameters: n=10000, knn_k=5, train_ratio=0.7, sample_n=100, ft_d=10. This run took over 8 hours (08:35:25), achieving an accuracy of 87.03%.

n	knn_k	train_ratio	sample_n	ft_d	**acc**	t/s
500	5	0.7	100	10	**0.79**	25.15
200	5	0.7	100	10	**0.70**	4.44
1000	5	0.7	100	10	**0.84**	00:01:20
500	**2**	0.7	100	10	**0.78**	19.37
500	**10**	0.7	100	10	**0.78**	24.22
500	5	**0.3**	100	10	**0.69**	18.02
500	5	**0.5**	100	10	**0.75**	21.73
500	5	**0.9**	100	10	**0.82**	23.19
500	5	0.7	**10**	10	**0.82**	2.57
500	5	0.7	**1000**	10	**0.77**	00:04:11
500	5	0.7	100	**5**	**0.70**	25.32
500	5	0.7	100	**20**	**0.84**	25.87

Table 1: Experimental results

5 REFLECTION

When solving the Laplacian equation, we noted that if there were 3 points with different labels connected to each other by strokes, the condition $w_i - w_j = \sigma$ is cannot be satisfied. Thus we had to use least square solution of the linear system to resolve such conflict. Such solution might cause inaccurate computation for the remaining points, thus affects the prediction results.

Since in our work we use strokes to capture jump conditions between different label spaces, which demonstrates the property of winding number to separate space, we need sufficient information about strokes to correctly reflect such property. This means the graph we generate needs certain amounts of strokes between each pair of label spaces. Specifically, if there is no stroke between 2 certain label spaces, the jump condition will not be specified and thus their winding number will remain harmonic across the boundary. Figures 13 and 14 illustrate this scenario. This condition should be guaranteed with sufficient sample points, training ratio and knn_k to ensure the degree of connection.

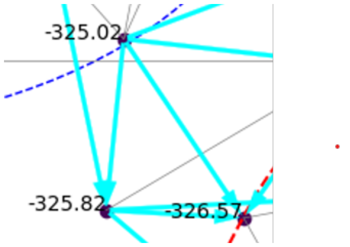

Figure 12: Case in which 3 points with different labels connected by stokes, thus the jump condition cannot be satisfied.

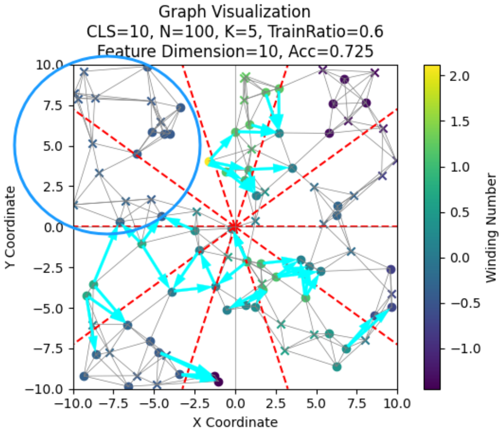

Figure 13: winding numbers will remain harmonic across the boundary.

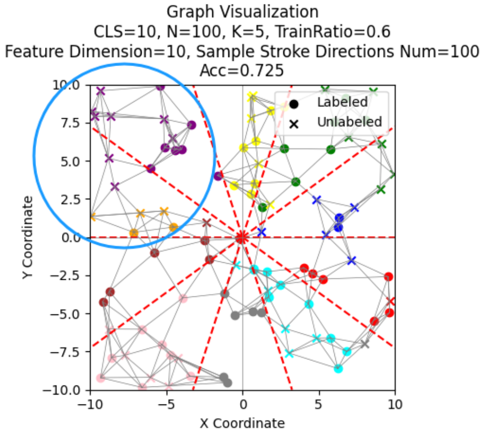

Figure 14: Incorrect prediction results occur when there are no strokes crossing the boundaries.

Also, we observed that if there are only a few edges connecting different parts within the same label space, this method may lead to errors due to the lack of constraints between the parts. This might lead to the algorithm ignoring the harmonic condition within the label space, and thus consider them as different label spaces.

Since our work requires massive computation on the process of computing winding numbers, we might need to implement GPU acceleration to perform our experiments with larger dataset. The aim of our work is to explore the validity of generalized winding number as a feature in the spacial segmentation task. The accuracy of prediction we observed generally falls between 70% to 90%. Although it falls far behind the SOTA of 98%, this feature still proves to be valuable and worth further exploration to incorporate into current methods.

REFERENCES

[1] Gavin Barill, Nia Dickson, Ryan Schmidt, David I.W. Levin, and Alec Jacobson. 2018. Fast Winding Numbers for Soups and Clouds. *ACM Transactions on Graphics* (2018).

[2] Nicole Feng. 2023. *Winding Number on Discrete Surfaces.* https://nzfeng.github.io/research/WNoDS/index.html

[3] Nicole Feng, Mark Gillespie, and Keenan Crane. 2023. Winding Numbers on Discrete Surfaces. *ACM Trans. Graph.* 42, 4, Article 36 (jul 2023), 17 pages. https://doi.org/10.1145/3592401

[4] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. *OpenCLIP*. https://doi.org/10.5281/zenodo.5143773

[5] Alec Jacobson, Ladislav Kavan, and Olga Sorkine. 2013. Robust Inside-Outside Segmentation using Generalized Winding Numbers. *ACM Trans. Graph.* 32, 4 (2013).