

1 **NeLo + Mapper: Learned Laplacian-driven 3D mesh segmentation**
2
3

4 BLAŽ BULIĆ, University of Ljubljana, Slovenia
5
6

7 This work introduces a 3D point cloud segmentation method by integrating Neural Laplacian Operator (NeLo) with the Mapper
8 algorithm. We replace Mapper's manual filter functions with NeLo's learned spectral embeddings to automate the selection of filter
9 function. Evaluated on ShapeNet objects, our approach produces finer segmentations than spectral clustering and manual Laplacian
10 baselines, especially for geometrically distinct features.

11 CCS Concepts: • Computing methodologies → Mesh geometry models; Point-based models; Topological data analysis; Segmentation.
12 Additional Key Words and Phrases: NeLo, Laplacian Operator, Mapper, Point Cloud Segmentation
13

14 **ACM Reference Format:**
15 Blaž Bulić. 2025. NeLo + Mapper: Learned Laplacian-driven 3D mesh segmentation. 1, 1 (May 2025), 6 pages. <https://doi.org/10.1145/nmnnnnnn.nmnnnnnn>
16
17

18 **1 Introduction**
19

20 3D point clouds appear increasingly often as a data representation in various domains, including autonomous navigation,
21 robotics, urban mapping, and biomedical imaging. Despite their popularity, the usage of point clouds comes with multiple
22 challenges due to noise, irregular sampling, and the lack of explicit connectivity of the sampled object. Traditional
23 approaches for processing 3D point clouds rely on triangulation-based methods or handcrafted features, to first construct
24 a mesh from the points, which often introduce inconsistency when dealing with thin structures or sharp features of
25 the clouds. These limitations make tasks such as segmentation, shape analysis, and object recognition challenging for
26 objects that exhibit the unwanted properties.
27

28 Recent advances in deep learning have taken steps towards overcoming these challenges. In particular, Neural
29 Laplacian Operators (NeLo) [4] provide a new solution by learning the Laplacian operator directly from point cloud data.
30 NeLo constructs a k -nearest neighbor (KNN) graph from the raw data and uses a Graph Neural Network (GNN) to learn
31 appropriate edge weights, used for approximating the continuous Laplacian operator. The spectral embeddings we get
32 from NeLo provide us with intrinsic geometric features, making them well-suited for various downstream applications.
33

34 Similarly, the Mapper algorithm, a tool from topological data analysis (TDA), has been used to generate simplified
35 representations of high-dimensional data. The mapper works by clustering data based on a chosen filter function
36 and building a nerve (graph) that reflects the global topological structure. However, the quality of Mapper's output is
37 highly sensitive to the choice of filter function, which is typically selected manually. This manual selection can lead to
38 inconsistent results and limits the method's applicability.
39

40 The goal of our project is to integrate NeLo with Mapper for 3D point clouds, by using the spectral embeddings from
41 NeLo as the filter function in Mapper. This substitution transforms the Mapper pipeline into an end-to-end framework
42

43 Author's Contact Information: Blaž Bulić, bb53717@student.uni-lj.si, University of Ljubljana, Ljubljana, Slovenia.
44

45 Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not
46 made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components
47 of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on
48 servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
49

50 © 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
51

52 Manuscript submitted to ACM
53

54 Manuscript submitted to ACM
55

53 that uses learned geometric features instead of relying on handcrafted choices. We test our approach on of the most
 54 basic use-cases, that is 3D point cloud segmentation, a critical step for many downstream tasks.
 55

56 2 Related Work

57 The field of 3D data analysis and neural operator learning has evolved in recent years. Our work builds upon several
 58 key contributions listed below.

59 *Laplacian for Geometry Processing:* Laplacian operators play a central role in many geometry processing tasks. Botsch
 60 et al. [1] provide an overview of Laplacian-based methods in their book *Polygon Mesh Processing*. They detail how the
 61 Laplacian can be used for mesh smoothing, parameterization, and other use cases.

62 *Neural Laplacian Operators:* Pang et al. [4] introduced the Neural Laplacian Operator for 3D point clouds. Their
 63 method uses a GNN to learn edge weights on a k -nearest neighbour graph, producing spectral embeddings that robustly
 64 capture intrinsic geometric features even in the presence of noise and sparsity. This data-driven approach has set a new
 65 benchmark for point cloud processing. The authors get promising results for tasks such as head diffusion, smoothing
 66 and others.

67 *Topological Data Analysis and Mapper:* Singh et al. [7] introduce topological method for high-dimensional data
 68 analysis with the development of the Mapper algorithm. Mapper creates a simplified graph representation by clustering
 69 data based on a continuous filter function. Although Mapper has proven effective in revealing global structures, its
 70 reliance on manually chosen filters often presents the biggest challenge.

71 *Neural Operator Learning:* Li et al. [3] developed the Fourier neural operator, which uses frequency-domain techniques
 72 to achieve discretization-independent learning. This highlights the effectiveness of spectral methods in capturing global
 73 features, a concept that motivates our use of NeLo's spectral embeddings as a filter in Mapper.

74 *Point Cloud Segmentation:* PointNet, introduced by Qi et al. [5], was among the first architectures to directly process
 75 point cloud data for segmentation and classification. Although PointNet achieves competitive performance, it relies on
 76 engineered features. In contrast, our approach utilizes learned spectral embeddings to drive segmentation, providing a
 77 more robust and natural grouping of points.

78 PointNeXt [6] builds on the foundations of PointNet++ by introducing design principles for constructing improved
 79 point-based backbones. It simplifies the architecture while preserving or improving accuracy, using residual blocks,
 80 local aggregation, and normalization strategies. PointNeXt achieves state-of-the-art results on several segmentation
 81 benchmarks while maintaining computational efficiency. Although effective, PointNeXt, like its predecessors, relies on
 82 local geometric structures and handcrafted inductive biases, whereas our method leverages global spectral embeddings
 83 learned through a neural Laplacian operator.

84 These contributions have individually advanced the fields of geometric deep learning and topological data analysis.
 85 Our project aims to combine parts of different approaches for a specific use case of 3D point cloud segmentation.

86 3 Method

87 In this section we describe how we compute the end segmentation of 3D point cloud. In section 3.1 we show the process
 88 of extracting the Laplacian matrix from NeLo framework. In sections 3.2 and 3.3 we describe the two methods we use
 89 for segmentation. Lastly we describe the manually computed Laplacian matrix we use for comparison in section 3.4.

105 **3.1 Learned Laplacian Extraction**

106 Because the NeLo framework output is not a Laplacian operator in its matrix form, but is represented as a graph with
 107 weighted edges, we have to first extract the matrix so we can then use it in our segmentation.
 108

109 Given an input 3D point cloud converted to a mesh graph, the pre-trained NeLo model predicts an edge weight w_{ij}
 110 for every edge (i, j) in the graph. We assemble these weights into a sparse adjacency matrix
 111

$$112 \quad A_{ij} = \frac{1}{2}(w_{ij} + w_{ji}),$$

113 and then form the degree matrix D with $D_{ii} = \sum_j A_{ij}$. Finally, the Laplacian matrix is computed as
 114

$$115 \quad L_{\text{NeLo}} = D - A,$$

116 which we can then use for downstream applications.
 117

119 **3.2 Spectral Clustering**

120 First segmentation method takes the spectral embedding of the Laplacian matrix and then obtains clusters directly
 121 from that. To segment the 3D point cloud into k parts, we use L_{NeLo} and compute its $k + 1$ smallest eigenpairs
 122 $\{(\{\lambda_0, \phi_0\}, (\lambda_1, \phi_1), \dots, (\lambda_k, \phi_k)\}$ via SciPy's eigsh. We discard the trivial constant eigenvector ϕ_0 , and embed each
 123 vertex i as
 124

$$125 \quad (\phi_1(i), \phi_2(i), \dots, \phi_k(i)) \in \mathbb{R}^k.$$

126 K-means is then applied in this k -dimensional space to obtain hard cluster labels for each point.
 127

130 **3.3 Mapper-Based Segmentation**

131 Second segmentation method leverages the Mapper algorithm in the following way. We use the first m nontrivial
 132 eigenvectors of L_{NeLo} as a filter function in the KeplerMapper pipeline. Let ψ_1, \dots, ψ_m be these eigenvectors; we stack
 133 them into an $N \times m$ "filter" matrix. With cover parameters $\{\text{n_cubes}, \text{perc_overlap}\}$ and the DBSCAN clustering
 134 algorithm, Mapper partitions the filter space into overlapping bins and clusters within each. The resulting nerve graph
 135 nodes induce an assignment of each original point to one or more Mapper clusters; we resolve overlaps by assigning
 136 each point to the node it appears in most frequently.
 137

140 **3.4 Manual Laplacian Baseline**

141 In order to compare and evaluate our results, we construct a standard Gaussian-kernel Laplacian on the same mesh
 142 vertices. We compute a full adjacency matrix
 143

$$144 \quad W_{ij} = \exp(-\|x_i - x_j\|^2 / (2\sigma^2)),$$

145 zeroed on the diagonal, form $D_{ii} = \sum_j W_{ij}$, and let $L_{\text{Gauss}} = D - W$. We then perform both spectral clustering and
 146 Mapper segmentation exactly as above but using the manually computed L_{Gauss} in place of L_{NeLo} .
 147

150 **4 Results**

151 We test out approach on objects from the ShapeNet [2] database. Our focus is on the qualitative/visual results as
 152 meaningful quantitative metrics are hard to come by without the ground truth segmentations which we do not have.
 153 Despite that, we can compare some metrics which measure the quality of clustering. The chosen metrics are : Silhouette,
 154

Davies-Bouldin index(DBI) and Calinski-Harabasz index(CHI). We showcase three informative examples, which help demonstrate the approaches strengths and weaknesses.

We set $k = 5$ clusters for spectral clustering. For Mapper, we used $m = 4$ filter functions for the plane and desk, and $m = 3$ for the chair, with $n_cubes = 10$, $perc_overlap = 0.3$, and DBSCAN parameters $\varepsilon = 0.05$ and $min_samples = 10$.

The quantitative results are presented in table 1 and the qualitative results are shown in figures 1, 2 and 3. We first notice that the number of clusters produced with the Mapper method is very large, even if it might not seem like it in the figures. This happens because Mapper clusters each filter hyperspace separately instead of clustering the whole filter space at once. This is not always bad, as it allows for finding finer segmentations as demonstrated in the plane example, where it distinguishes between the planes motors and hull and even identifies different parts of the rear wing in the case of NeLo Laplacian. We also notice that the metrics for Mapper methods are significantly worse than the KMeans method. This is a consequence of the phenomena described above. The huge number of clusters significantly worsens the metrics, although the visual results are not so bad.

When we compare between NeLo and Manual Mapper methods we see that the NeLo method is capable of producing a finer segmentation than Manual Mapper. We see this in the chair example where it differentiates between the backrest and the seat. Lastly we notice that the method does not work well on objects that have no distinct features that would be distinguishable with Laplacian operator. This can be seen in the desk example where Mapper methods finds clusters with no meaning. When comparing quantitative measures, we see that they perform relatively similarly, with one performing better than the other in some metrics and vice versa.

Method	Chair (Silhouette / DBI / CHI / #Clusters)	Plane (Silhouette / DBI / CHI / #Clusters)	Desk (Silhouette / DBI / CHI / #Clusters)
Spectral+KMeans	0.689 / 0.559 / 5363.99 / 5	0.641 / 0.461 / 3179.109 / 5	0.390 / 0.893 / 3963.550 / 5
Mapper (NeLo)	0.193 / 0.784 / 15945.96 / 94	-0.249 / 4.577 / 118.624 / 79	-0.295 / 1.373 / 34.891 / 396
Mapper (Gauss)	0.033 / 0.800 / 8212.56 / 78	-0.202 / 3.746 / 152.137 / 51	-0.306 / 1.832 / 47.114 / 352

Table 1. Clustering-quality metrics (Silhouette / DBI / CHI / #Clusters) for our three examples

5 Conclusion

In this work, we presented a novel approach to 3D point cloud segmentation by combining Neural Laplacian Operators (NeLo) with the Mapper algorithm. By leveraging NeLo's learned spectral embeddings as filter functions, we transformed Mapper into an end-to-end framework that eliminates the need for manual filter selection. Our experiments on ShapeNet objects revealed that this method can produce finer segmentations compared to traditional spectral clustering and manual Laplacian baselines, particularly for objects with distinct geometric features. However, the approach struggles with objects lacking such features, as seen in the desk example. Additionally, the Mapper method's tendency to produce numerous clusters impacts quantitative metrics, though visual results remain meaningful. The code and experiments are available at: <https://github.com/BlazBulic/DP-Project-NeLo-Mapper>. Future work could explore adaptive parameter tuning for Mapper or integrating a method to combat the explosion in number of clusters such as smart merging of clusters after the original clustering.



Fig. 1. Different segmentation methods for chair

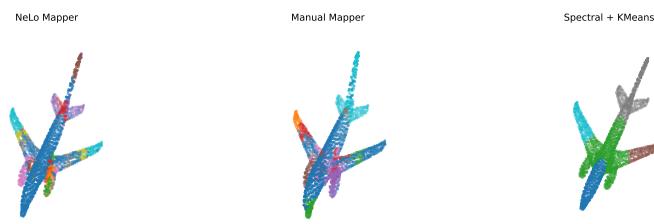


Fig. 2. Different segmentation methods for plane

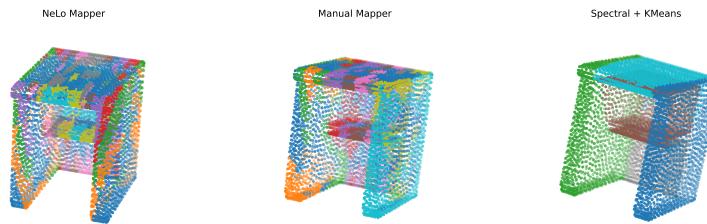


Fig. 3. Different segmentation methods for desk

References

- [1] Mario Botsch, Léif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. 2010. *Polygon Mesh Processing*. AK Peters/CRC Press.
- [2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository*. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University – Princeton University – Toyota Technological Institute at Chicago.

- 261 [3] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. 2020. Fourier Neural Operator for Parametric
262 Partial Differential Equations. *arXiv preprint arXiv:2010.08895* (2020). <https://arxiv.org/abs/2010.08895>
- 263 [4] Bo Pang, Zhongtian Zheng, Yilong Li, Guoping Wang, and Peng-Shuai Wang. 2024. Neural Laplacian Operator for 3D Point Clouds. *ACM Transactions
264 on Graphics* 43, 6 (2024), 1–14. doi:10.1145/3687901
- 265 [5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the
266 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 77–85. doi:10.1109/CVPR.2017.16
- 267 [6] Yujing Qian, Meng-Hao Xu, Zhiyuan Zhang, Enze Xie, Wenhui Wang, Yulu Wei, Zhenguo Li, Chen Change Loy, and Bo Dai. 2022. PointNeXt:
268 Revisiting PointNet++ with Improved Training and Scaling Strategies. In *Advances in Neural Information Processing Systems*.
- 269 [7] Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. 2007. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object
270 Recognition. In *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 91–100. doi:10.2312/SPBG/SPBG07/091-100
- 271
- 272
- 273
- 274
- 275
- 276
- 277
- 278
- 279
- 280
- 281
- 282
- 283
- 284
- 285
- 286
- 287
- 288
- 289
- 290
- 291
- 292
- 293
- 294
- 295
- 296
- 297
- 298
- 299
- 300
- 301
- 302
- 303
- 304
- 305
- 306
- 307
- 308
- 309
- 310
- 311
- 312