Spectral Informed Mamba for Robust Point Cloud Processing Supplementary Material

Ali Bahri * Moslem Yazdanpanah Mehrdad Noori Sahar Dastani
Milad Cheraghalikhani Gustavo Adolfo Vargas Hakim David Osowiechi
Farzad Beizaee Ismail Ben Ayed Christian Desrosiers

LIVIA, ÉTS Montréal, Canada
International Laboratory on Learning Systems (ILLS)

1. Computational Efficiency, Runtime, and Memory Usage

In this section, we conducted a comprehensive analysis regarding the computational efficiency, runtime, and memory usage of our Surface-Aware Spectral Traversing (SAST) approach. The focus of these experiments is to assess the overhead introduced by our SAST in comparison to the Point-Mamba backbone.

Our SAST strategy employs SciPy's sparse eigen-solver (function `eigs` implementing the implicitly restarted Arnoldi algorithm) to compute the first eigenvectors of the graph Laplacian. As highlighted in the main paper, this operation is not expensive since the size of this matrix depends on the number of patches which is much less than the original number of points (128 vs. 2048). Additionally, even when increasing the number of patches (tokens), the computational overhead of this step is limited due to three reasons: *i*) the Laplacian matrix is very sparse as we only consider the K nearest neighbors of each patch ($K \approx 20$), *ii*) we only compute the first k eigenvectors ($k \approx 5$ in our experiments), and *iii*) this computation is done **only once** for each point cloud in a **pre-processing step**.

Memory Usage: As shown in Fig. 1, the memory usage of our SAST strategy (black line) is significantly lower than the memory usage of the Point-Mamba backbone (red line). When increasing the number of patches along the x-axis, our strategy based on a sparse eigen-solver does not require substantially more memory compared to the backbone. The star in this figure shows the used number of tokens in downstream tasks.

Runtime: Our SAST strategy, which can be implemented in the data loader and ran in parallel on CPU, is also fast. As can be seen in Fig. 2, the runtime of SAST scales well when increasing the number of patches (tokens), and only a small amount of runtime is added in training or inference for the token length of 128 used in our main experiments (yellow star).

FLOPS: Fig. 3 presents the relationship between FLOPS and token length for both the Point-Mamba backbone and our SAST method. Compared to running the Point-Mamba back, our SAST demonstrates a more gradual increase in FLOPS due to the use of sparse computations and the low number of eigenvectors involved. Once again, this shows the limited overhead of incorporating SAST, even as token length increases.

2. Additional Ablation Study

The Effect of HLT on Classification: In this section, we investigate the effect of the Hierarchical Local Traversing (HLT) strategy on the classification task. The results for HLT on the ObjectNN dataset are shown in Tab. 2. As observed, the HLT strategy underperforms compared to SAST across all three settings (OBJ-BG, OBJ-ONLY, and PB-T50-RS), regardless of whether the model is trained from scratch or pretrained.

This performance gap highlights the limitations of the HLT strategy in tasks requiring global understanding, such as classification. Specifically, HLT processes high-level information from all eigenvectors simultaneously in a **single traversal order** (forward and backward), which is effective for segmentation tasks but may lead to insufficiently distinct feature representations for global classification. In contrast, SAST processes information from different eigenvectors in **separate traversals**, enabling better representation of high-level structures critical for classification tasks.

Number of Eigenvectors in HLT: Tab. 2 evaluates the impact of varying the number of eigenvectors in our proposed HLT strategy on part segmentation performance using the ShapeNetPart dataset, considering both *training from scratch* and *training from pretrained weights*.

*Correspondence to ali.bahri.1@ens.etsmtl.ca

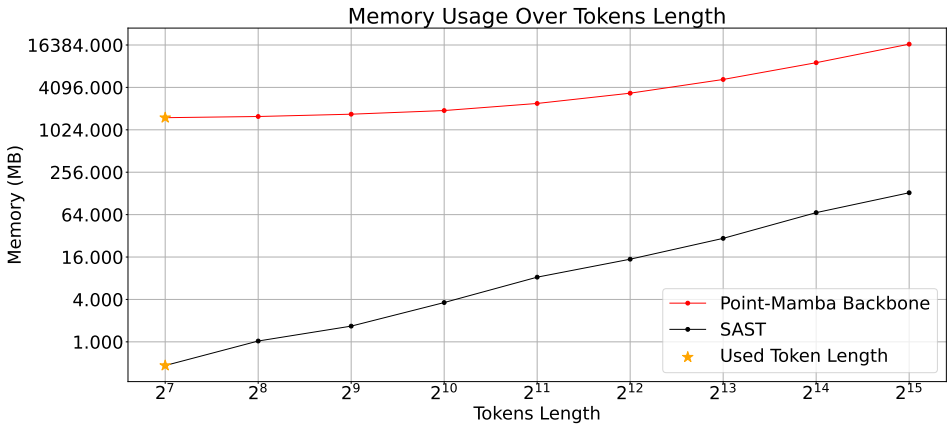

Figure 1. Memory Usage Over Tokens Length. Both axes are scaled by $\log_2$ for better visualization.

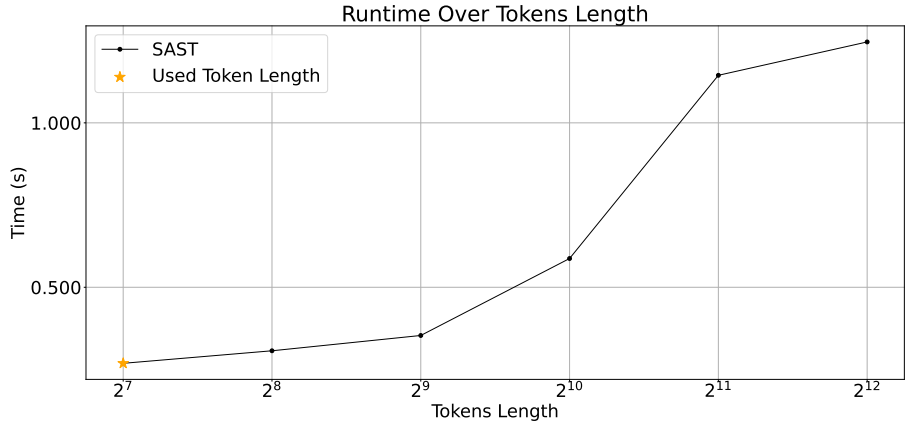

Figure 2. Runtime Over Tokens Length. Both axes are scaled by $\log_2$ for better visualization.

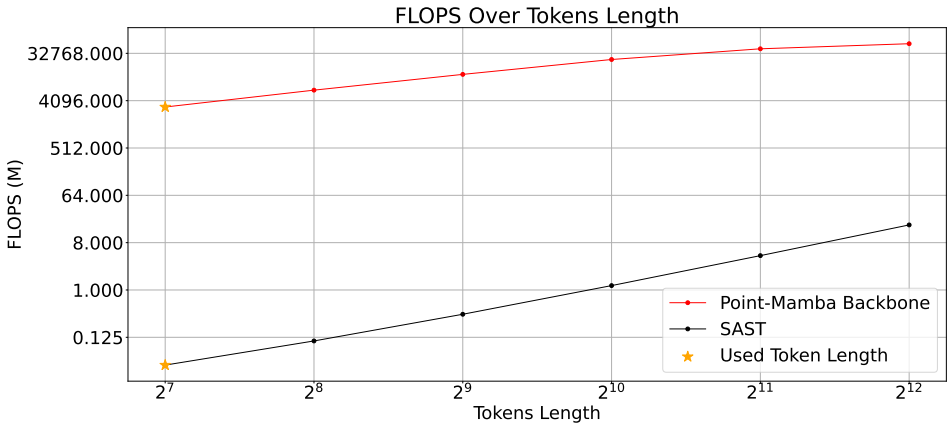

Figure 3. FLOPS Over Tokens Length. The horizontal axis is scaled by $\log_2$ for better visualization.

Table 1. Object classification on ScanObjectNN. Accuracy (%) is reported.

Methods	Backbone	OBJ-BG	OBJ-ONLY	PB-T50-RS
Training from scratch				
Ours (HLT)	Mamba	90.87	90.53	86.22
Ours (SAST)	Mamba	**92.25**	**91.39**	**87.30**
Training from pretrained				
Ours (HLT)	Mamba	92.94	91.42	87.52
Ours (SAST)	Mamba	**94.32**	**91.91**	**89.10**

In both scenarios, the segmentation accuracy, measured by the mean Intersection over Union (mIoU), demonstrates that the number of eigenvectors directly influences performance. The accuracy peaks at four eigenvectors, achieving 85.9% (scratch) and 86.1% (pretrained), as this setting provides an optimal balance for spatial encoding.

When the number of eigenvectors is low, the model fails to partition points accurately, resulting in unrelated points being grouped into the same segment. Conversely, when the number of eigenvectors is high, the performance decreases slightly due to redundancy and noise. Higher-order eigenvectors encode finer details or localized variations, which may not align with meaningful segmentation. This can lead to overfitting or confusion between closely related parts.

Table 2. Part segmentation on ShapeNetPart.

Methods	Param. (M)	Eigenvectors	mIoU
Training from scratch			
Ours (HLT)	12.3	3	85.6
Ours (HLT)	12.3	**4**	**85.9**
Ours (HLT)	12.3	5	85.9
Ours (HLT)	12.3	6	85.8
Training from pretrained			
Ours (HLT)	12.3	3	85.8
Ours (HLT)	12.3	**4**	**86.1**
Ours (HLT)	12.3	5	86.0
Ours (HLT)	12.3	6	85.8

3. Additional Visualization

Segmentation Results. Fig. 4 provides results for six object categories ("Airplane," "Bag," "Car," "Chair," "Motorbike," and "Guitar) obtained by our HLT method. In this figure, each point is color-coded based on its class label. The comparison between the ground truth (GT) and the predicted segmentation demonstrates the outstanding performance of our method, as well as its ability to capture fine-grained details.

Dataset Challenges and Ground Truth Anomalies in ShapeNetPart. The ShapeNetPart dataset is widely recognized as a challenging benchmark for 3D point cloud segmentation. Upon further investigation of the dataset, we observed certain inconsistencies and inaccuracies in the provided GT annotations. As illustrated in Fig. 5, our method exhibits a visually better segmentation than the ground truth.

Such discrepancies in the ground truth highlight potential limitations in the dataset itself, which poses a challenge for both training and evaluation. This phenomenon also provides an explanation for the observation that recent state-of-the-art methods (as listed in Table 2 of the main paper) achieve similar mIOU performance on this dataset, with only marginal improvements. The inherent noise and errors in the ground truth annotations make it difficult for methods to demonstrate significant gains in segmentation quality.

Despite these challenges, our method still achieves competitive performance while maintaining robust predictions that align closely with the underlying geometric features of the objects.

Reconstruction Results. Fig. 6 showcases the reconstruction capability of Masked Autoencoders (MAEs) on point cloud data using the ShapeNet dataset. The figure consists of three columns for each sample, illustrating the progression from the input point cloud to the final reconstructed result.

The first column, "Input Point Cloud", represents the original point cloud data, providing a complete view of the object before any masking or processing. This serves as a reference for evaluating the quality of the reconstruction. The second column, displays the same point cloud after a portion of the data has been masked out. The visible points indicate the sparse information available to the model during the reconstruction phase.The third column, "Reconstructed Point Cloud", demonstrates the MAE's ability to predict and restore the masked regions, resulting in a nearly complete reconstruction that aligns closely with the original structure.

These results underline the effectiveness of MAEs in capturing and reconstructing geometric details from incomplete data, making them well-suited for extracting meaningful features.

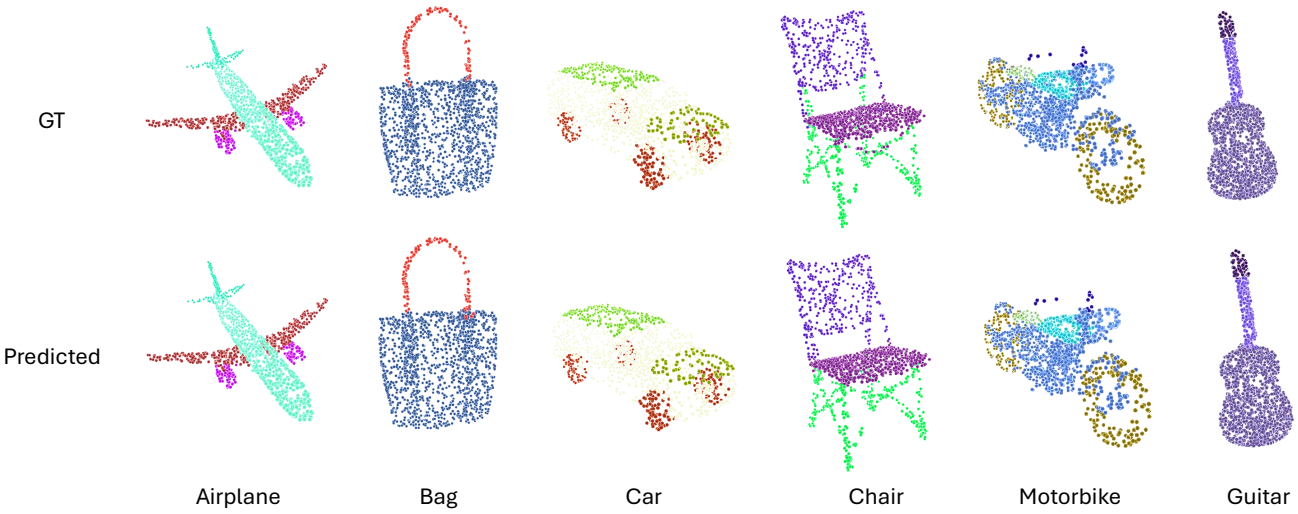

Figure 4. The qualitative results of part segmentation of our HLT method on ShapeNetPart dataset

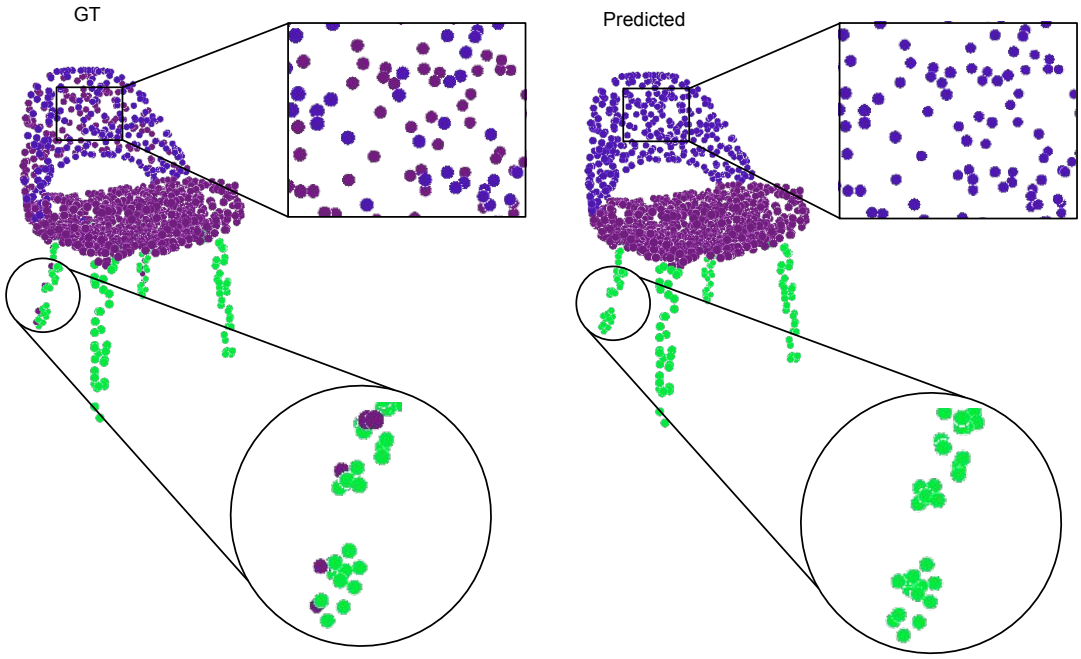

Figure 5. An example illustrating inconsistencies in the ground truth (GT) annotations. The predicted segmentation (right) is geometrically more accurate and consistent compared to the GT (left).

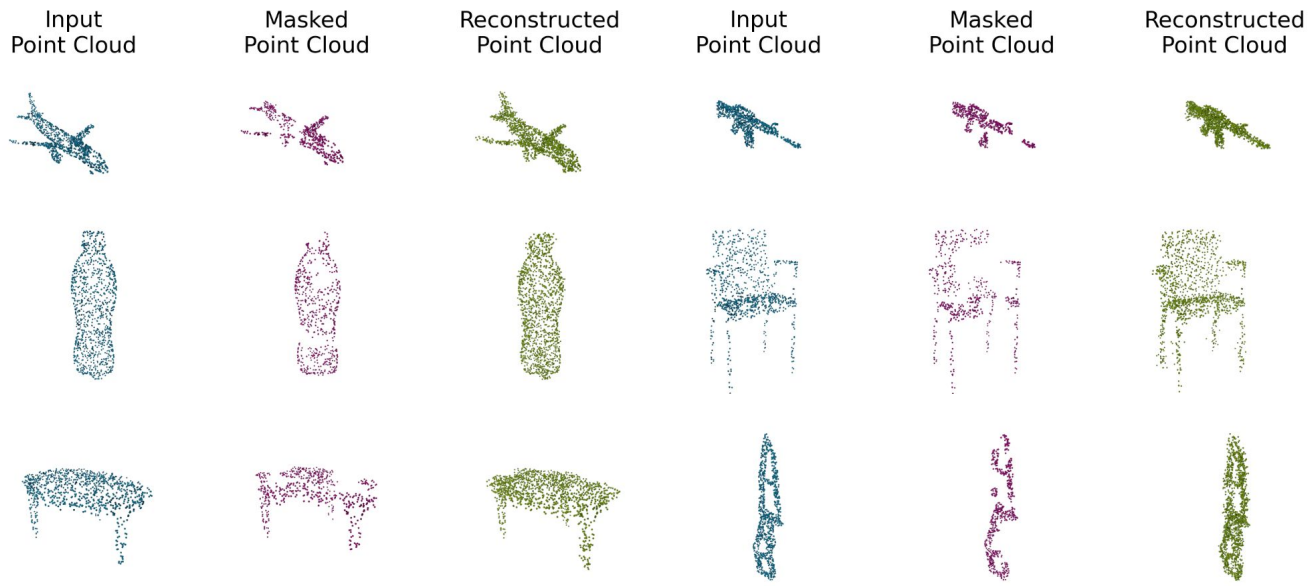

Figure 6. Reconstruction results on the ShapeNet dataset