Appendix of CodedVTR: Codebook-based Sparse Voxel Transformer with Geometric Guidance

1. Architecture Design

In this section, we present the detailed model architecture. We follow the MinkowskiNet's [1] scheme and adopt their ResNet-20 and ResNet-42 architecture, and replace their ResNet-like building block with our CodedVTR block. The CodedVTR block shares the same input and output channel size as the ResNet block.

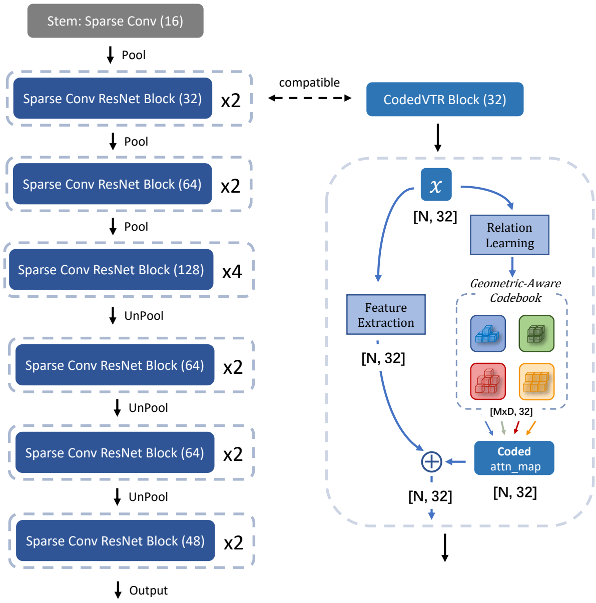

Figure 1. **The architecture of CodedVTR.** The CodedVTR shares the backbone architecture like MinkowskiNet [1] while replacing the convolution block with CodedVTR block.

2. Design of Geometric Pattern

In this section, we give a detailed description of the design process of the geometric pattern. We apply clustering on the one-hot neighbor sparse masks for each dataset, stride, and dilation. We randomly sample 10 scenes for each dataset and acquire the neighbor sparse mask in the $3 \times 3 \times 3$ region with different strides and dilations. It could be represented with a 27-dimension one-hot sparse mask, where each element represents whether the neighbor location is occupied or not. We apply K-modes clustering [2] to generate M representative sparse patterns, and the clustering centroids are chosen as the geometric pattern used in geometry-aware attention. We take the dilation=1, stride=2, sparse patterns on semanticKITTI as an example. Fig. 2 visualizes the t-SNE of the clustered sparse patterns. The hyper-parameter M is chosen by investigating the "saturate point" of the clustering error, as illustrated in Fig. 2, we set M as 8.

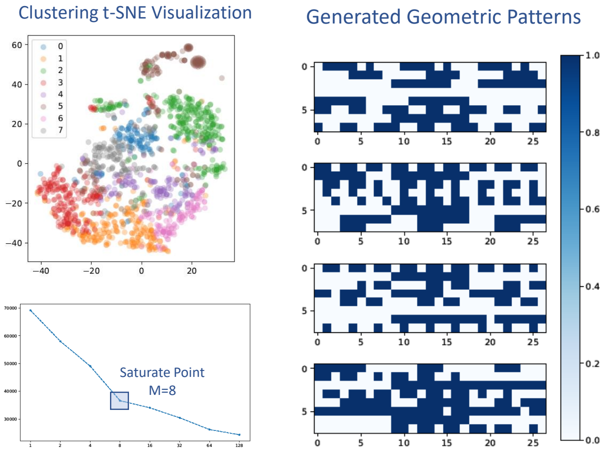

Figure 2. **The design process of geometric pattern.** The geometric patterns are generated through clustering on the neighbor sparse masks for each dataset, stride and dilation.

3. Comparison of Model Generalization

In this section, we present the analysis of model generalization. We tune the training hyperparameters (weight decay and learning rate) to align the training accuracy for VoTR (voxel transformer) and our CodedVTR under similar model capacity. As shown in Fig. 3, the CodedVTR has notably higher performance on the validation set, denoting that it has better generalization ability.

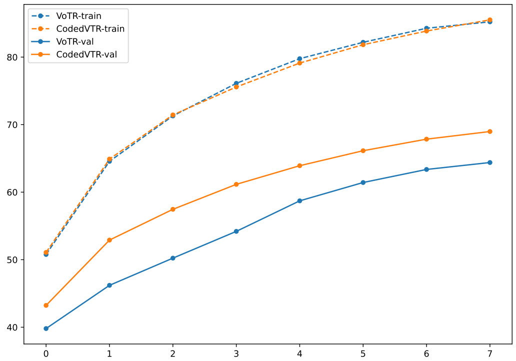

Figure 3. **Comparison of the generalization for voxel transformer and CodedVTR** With similar training accuracy, the CodedVTR has superior validation accuracy, proving that it has better generalization ability compared with the original voxel transformer.

4. Limitations and Future Work

This section discusses possible directions or approaches to improve our proposed CodedVTR. Firstly, when the codebook-based attention has “hard” choices, it becomes a “discretized self-attention” and has the potential for better efficiency. In the future, we might explore techniques to train a discretized version of our CodedVTR to achieve better inference-time efficiency. Secondly, using the clustering centroids as the geometric patterns of the codebook elements could be suboptimal. Developing more advanced learning-based methods to acquire them in the training process jointly is an interesting future exploration.

References

[1] Christopher Bongsoo Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019.

[2] Nelis J. de Vos. kmodes categorical clustering library. `https://github.com/nicodv/kmodes`, 2015–2021.