CodedVTR: Codebook-based
Sparse Voxel Transformer with Geometric Guidance

Tianchen Zhao[1], Niansong Zhang[1], Xuefei Ning[1], He Wang[2], Li Yi[13*], Yu Wang[1]

[1]Tsinghua University   [2]Peking University   [3]Shanghai Qizhi Institude

{suozhang1998,foxdoraame,ericyi0124}@gmail.com   niansong.zhang@outlook.com

hewang@pku.edu.cn   yu-wang@mail.tsinghua.edu.cn

Abstract

*Transformers have gained much attention by outperforming convolutional neural networks in many 2D vision tasks. However, they are known to have generalization problems and rely on massive-scale pre-training and sophisticated training techniques. When applying to 3D tasks, the irregular data structure and limited data scale add to the difficulty of transformer's application. We propose CodedVTR (**Code**book-based **V**oxel **TR**ansformer), which improves data efficiency and generalization ability for 3D sparse voxel transformers. On the one hand, we propose the codebook-based attention that projects an attention space into its subspace represented by the combination of "prototypes" in a learnable codebook. It regularizes attention learning and improves generalization. On the other hand, we propose geometry-aware self-attention that utilizes geometric information (geometric pattern, density) to guide attention learning. CodedVTR could be embedded into existing sparse convolution-based methods, and bring consistent performance improvements for indoor and outdoor 3D semantic segmentation tasks.*

1. Introduction

Recent advances in deep learning have significantly pushed forward representation learning on 3D point clouds. 3D deep learning models enable autonomous driving and robotic systems to perceive raw 3D sensor data. In this process, the 3D semantic segmentation task plays a crucial role in real-world scene understanding. It aims to classify each point into pre-defined semantic categories (e.g., car, pedestrian, table, floor, etc.), which provides point-wise perception information of the whole 3D scene. For 3D semantic segmentation on large-scale scenes, point-based methods [20, 22, 30] split the scene into cubical chunks and ap-
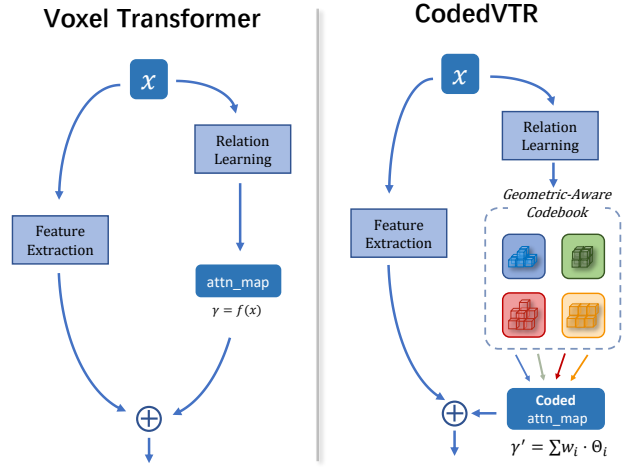

Figure 1. *CodedVTR* is a novel transformer-based building block for voxel-based 3D scene understanding tasks. We propose the **codebook-based attention** to alleviate the transformer's generalization problem, which is exacerbated in 3D tasks. Taking 3D point cloud's unique properties into consideration, we also propose the **geometry-aware attention** to leverage the geometric information into attention learning,

ply point-wise operations to each chunk. Differently, voxel-based approaches [3, 10, 21, 33] directly voxelize the whole scene and apply 3D sparse convolution to voxels, which usually yields superior efficiency and performance. In this paper, we follow the voxel-based scheme and focus on designing sparse voxel-based architecture for the 3D semantic segmentation task.

Transformers [24] have received massive attention and achieved state-of-the-art performance in many vision tasks such as image classification and semantic segmentation [15]. It discards the convolution-like inductive bias and adopts more general-purpose self-attention operations. Although less inductive bias gives transformers potentially

---

better representative ability, it also poses challenges to its generalization capability [23, 31]. Current transformers rely on large-scale data pre-training (ImageNet22K), strong data augmentation, and sophisticated hyper-parameter tuning to outperform convolution neural networks (CNNs). As the ViT [8] poses: when directly trained on the ImageNet, transformers "yield modest accuracies of a few percentage points below ResNets of comparable size". Prior studies [4, 13] prove that transformers have better expressive power compared with CNNs, and conclude that their poor performance on smaller datasets mainly comes from the poor generalization ability.

This problem is further aggravated on 3D tasks. 3D point cloud data has unique properties such as sparse and irregular structure, varying density, implicit geometric features in data locality [12, 14, 16]. For example, 95% of the voxels that locate farther than 5m from the scene center are empty in semanticKITTI [33]. Besides, different voxels have distinct local geometry patterns, e.g., the voxels on the desk and floor mainly have horizontal neighbors. Due to the large variance of density and geometric pattern in 3D data, capturing and adapting to these varying situations bring challenges for transformer's generalization. On the other hand, since high-quality labeled data are hard to acquire for 3D tasks, the dataset sizes are usually restricted, which also poses challenges to the generalization of transformer.

The above-mentioned phenomena stress the vitality of overcoming the generalization problem for transformers, especially for 3D tasks. We seek to address the generalization problem from the perspective of architecture design. We propose the **codebook-based attention** that regularizes the attention space to improve generalization. Specifically, we project self-attention maps into a subspace represented by the combination of several codebook elements (See Fig. 1). In this way, the dimension of attention learning is reduced from the whole attention space to several "prototypes" within the codebook. The projection constrains the attention learning to the subspace and could be viewed as a form of regularization for better generalization.

The above codebook plays a vital role in our design. Instead of restricting the spatial support of the codebook elements to be regular cubical regions, we in addition propose **geometry-aware attention** to incorporate inductive bias regarding the spatial patterns of 3D voxel data. Unlike pixels that have a regular and dense layout, the voxels have distinct geometric patterns and densities[1]. Existing sparse voxel CNNs [3] and transformer [17] use a fixed receptive field (spatial support) for all voxels. They solely rely on the learning process to extract geometric features without exploiting the voxel's geometric pattern. In contrast, we carefully design geometric regions of different shapes and

ranges by collecting and clustering the local sparse patterns in the input voxel data. Taking advantage of the codebook design, we assign these geometric regions to each codebook element as its attention spatial support. In this way, we could encourage attention learning to be adaptive to the sparse pattern of voxels.

The contributions of this work could be summarized into three aspects:

1) We address the transformer's generalization problem for the 3D domain, and propose CodedVTR, a transformer-based 3D backbone. The codebook-based self-attention projects the attention space into its subspace, serving as a regularization for better generalization.

2) We propose geometry-aware attention that exploits the sparse patterns of 3D voxel. It incorporates the inductive bias of geometric properties to guide the attention learning.

3) Our CodedVTR block can be effortlessly embedded into existing sparse convolution-based backbones. Replacing the sparse convolution with our CodedVTR brings consistent performance improvement on both indoor and outdoor 3D semantic segmentation datasets.

2. Related Works

2.1. Voxel-based 3D Semantic Segmentation

Deep learning for 3D scene understanding has gained much popularity in recent years. 3D semantic segmentation task is a representative task, in which a semantic label is assigned to every point in a given 3D point cloud. Earlier point-based approaches like [20, 22] split the scene into cubical chunks and process points in each chunk. These approaches struggle to yield satisfying performance on large-scale scenes like ScanNet [5] or SemanticKITTI [1]. Alternatively, voxel-based methods like MinkowskiNets [3] and SparseConvNets [10] apply the 3D voxelization to convert the irregular point cloud into regular sparse voxels to apply 3D convolution. They could directly process the whole scene and yield state-of-the-art performance on large-scale scene datasets. However, due to uniform-voxelization and restricted receptive field, sparse convolution struggles to handle the varying density and long-term relation modeling. In this paper, we seek to improve the voxel-based feature extractor with the more flexible transformer architecture, and propose geometry-aware self-attention to address the above-mentioned problems.

2.2. Transformer and Self-attention

Transformers have achieved great success in both NLP [24] and vision [8] tasks. Performance and theoretical analysis show that the self-attention operation has better expressive ability than convolution for its weaker inductive bias. Cordonnier et al. [4] proves that with sufficient number of heads, the self-attention could approximate the con-

---

[1]We summarize the geometric pattern and density of voxels as their "geometric information".

volution. It shows that the self-attention has better expressive ability compared with convolution, thus having the potential of being a better feature extractor. Moreover, transformers have more flexible receptive field compared with convolution and could leverage richer context information.

Some prior studies attempt to introduce the transformer to 3D point clouds. For point-based methods, PointTR [28] reformulates the point cloud completion task into a set-to-set translation problem and designs transformer to solve it. Point Transformer [30] and Pointformer [19] design self-attention operator for point cloud and improve point-based methods [20] on a variety of tasks. As for voxel-based methods, VoTR [17] designs a voxel-based transformer backbone for the 3D detection task. However, it solely adapts the self-attention in 2D vision to sparse voxels without addressing the generalization problem and 3D data's unique properties. Their reported performance and our reproduced results show that it suffers from the generalization problem. In CodedVTR, we propose **codebook-based self-attention** to alleviate such generalization problem and **geometry-aware self-attention** to leverage the geometric information for attention learning.

2.3. Generalization Issue of Transformer

The better representative ability of Transformer comes from its weaker inductive bias. However, it also makes transformers suffer from generalization problems and perform poorly on smaller-sized datasets [13, 23]. For instance, ViTs could not outperform ResNets when directly trained on CIFAR-10 or even ImageNet-1K. Large-scale pre-training or sophisticated augmentation and hyper-parameter tuning are required for alleviating the generalization problem. Many research attempts to alleviate this problem with less resource-consuming approaches. SAM [2] proposes a sharpness-aware optimizer to prevent the optimization from being trapped in sharp local minimal. Geng et al. [9] regularizes the optimization with matrix decomposition. Another line of studies [6, 26, 27] augment the model's inductive bias and improve the generalization via fusing the transformer with CNN.

3. Methods

In this section, we present Codebook-based Voxel Transformer (CodedVTR), a transformer-based 3D backbone with codebook-based and geometry-aware self-attention. Our design of the codebook-based and geometry-aware self-attention can be effortlessly embedded into existing 3D backbones to replace the vanilla sparse convolution. It alleviates the transformer's well-known generalization problem and works as a general building block with better performance and parameter efficiency than 3D sparse convolution.

This section is organized as follows. In Sec. 3.1 we introduce how to adapt the self-attention design to sparse voxels and formulate the basic "voxel transformer" model. Then, we present our design of the codebook-based self-attention that alleviates the generalization issue of transformer in Sec. 3.2. Finally, in Sec. 3.3 we introduce the geometry-aware attention mechanism that utilizes the geometric information to guide the attention learning, which also fits well with our codebook design.

3.1. Voxel Transformer

For self-attention in the sparse voxel domain, we define that the point cloud is voxelized into N sparse voxels with coordinates $p \in \mathbb{R}^{N \times 3}$ and features $x \in \mathbb{R}^{N \times C}$. To calculate the output for a voxel x_i, we conduct the self-attention operation in the local region $\varphi(x_i)$. The neighborhood $\varphi(x_i)$ represents the voxels located at nearby locations with some coordinate offsets (e.g., $(+1, -1, +1), ..., (0, -1, +1)$) from the centroid x_i. As discussed in prior studies [29, 30], self-attention could be decomposed into two parts: the "relation learning" part $\boldsymbol{f}$ and the "feature extraction" part $\boldsymbol{g}$. The "relation learning" part models the relative relations between the input voxel and its neighbors and generates the attention map $\boldsymbol{f}(\varphi(x_i))$. The attention map is then multiplied with the feature part and aggregated to produce the output. The calculation of the corresponding output y_i of the input voxel x_i goes as follows:

$$y_i = \sum_{x_j \in \varphi(x_i)} f_{ij}(\varphi(x_i)) \odot g(x_j) \tag{1}$$

where $f_{ij}(\varphi(x_i)) \in \mathbb{R}^H$ denotes the attention weight from voxel x_j to x_i (H is the head number), and $g(x_j)$ is the value at voxel x_j. For notation simplicity, we use a vector instead of a matrix to represent the attention map $\boldsymbol{f}(\varphi(x_i)) =$ Concat($\{f_{ij}(\varphi(x_i))\}_{x_j \in \varphi(x_i)}$) $\in \mathbb{R}^{N(\varphi)H}$, and write it as $\boldsymbol{f}(x)$ in the following texts, where $N(\varphi)$ stands for the neighbor size.

3.2. Codebook-based Self-attention

The attention map $\boldsymbol{f}(x)$ for a certain voxel x represents its relation with its $N(\varphi)$ neighbors. And we propose to constrain this attention to a learnable subspace through projection. Specifically, the projected attention is computed as the combination of multiple elements in a codebook. These codebook elements can be viewed as the "prototypes" of the attention weights and are jointly learned during the training process.

Fig. 2 gives out the graphical illustration of the codebook-based self-attention. To be more specific, the attention subspace is represented using the codebook elements $\Theta = \{\theta_1, \cdots, \theta_K\} \in \mathbb{R}^{K \times N(\varphi)H}$, where K denotes the codebook size and H denotes the head number. And the original attention $\boldsymbol{f}(x) \in \mathbb{R}^{N(\varphi)H}$ is projected onto this
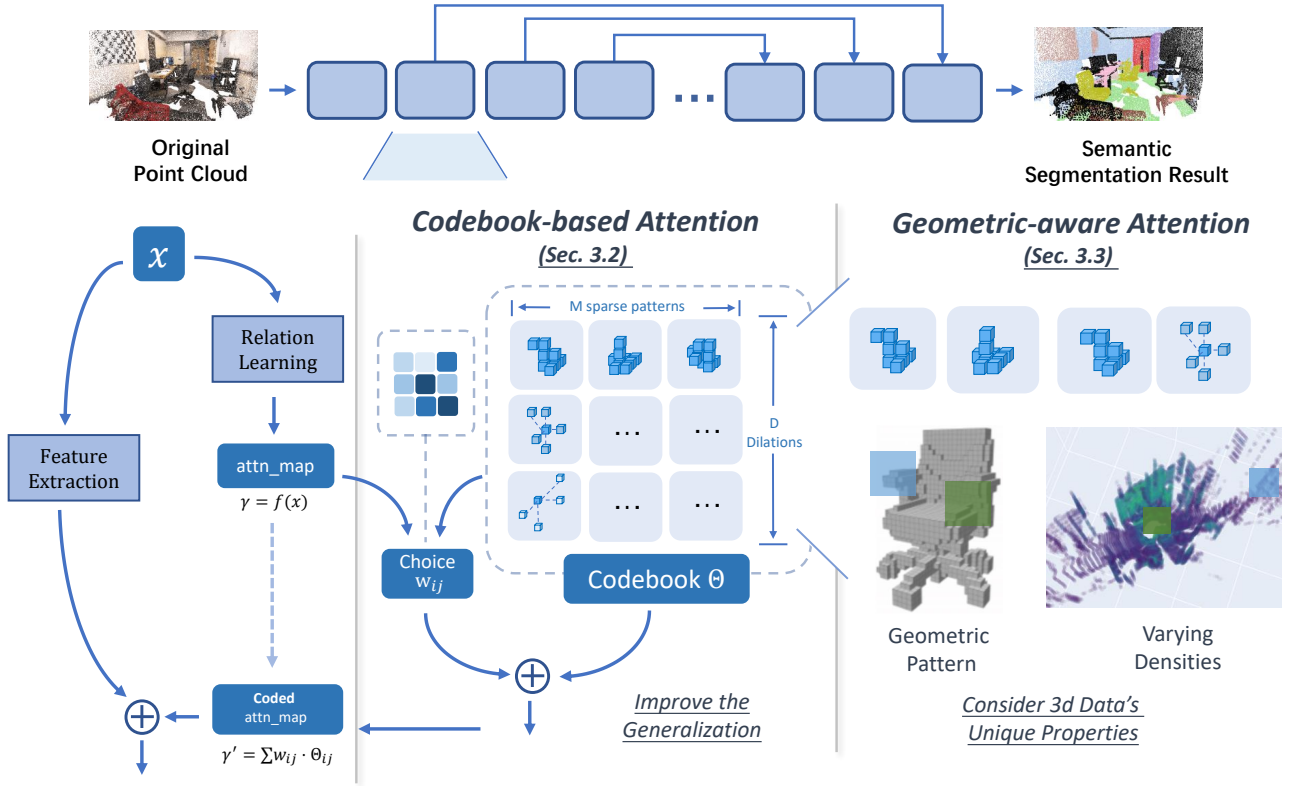
Figure 2. **Illustration of the CodedVTR block.** The codebook-based attention conduct optimization in attention weight subspace represented by the codebook. The geometry-aware attention explotits the unique properties of 3D point cloud and assign various geometric regions as the attention spatial supports of codebook elements.

subspace to become $\boldsymbol{f}_p(x) \in \mathbb{R}^{N(\varphi)H}$, where the subscript p denotes "projection". The projection goes as follows:

$$\boldsymbol{w} = \sigma(\{\psi(\theta_i, \boldsymbol{f}(x)\}_{i=1,\cdots,K}) \in \mathbb{R}^K$$
$$\boldsymbol{f}_p(x) = \Theta^T \boldsymbol{w} \tag{2}$$

where ψ calculates the similarity between the original attention $\boldsymbol{f}(x)$ and the codebook elements $\{\theta_i\}_{i=1,\cdots,K}$. σ denotes the softmax operation, and $\boldsymbol{w}$ is a "soft" choice of the codebook elements.

Our experimental results verify the effectiveness of the codebook-based self-attention. And its performance gain can be understood from two aspects. On the one hand, the subspace projection of attention can be viewed as a type of regularization that improves the generalization ability of the transformer. On the other hand, the codebook-based self-attention can be viewed as an intermediate design that bridges the architecture gap between convolution and vanilla self-attention. The two extreme cases of the codebook-based self-attention represent convolution and vanilla self-attention, respectively. When the codebook

has only one element ($K = 1$), it works like a convolution block. Whereas when K is sufficiently large, the subspace spanned by the codebook elements can easily have full rank. In this case, the codebook-based self-attention only conducts a mapping into a full-rank subspace without dimension reduction, and is similar to the vanilla self-attention. In another word, our design has a hyper-parameter (the codebook size K) that can flexibly adjust the trade-off between the generalization difficulty and representation capacity.

3.3. Geometry-aware Self-attention

Unlike the 2D pixels that have a dense and regular layout, the 3D voxels are sparse and irregular. Our geometry-aware attention design takes these two unique properties of the 3D sparse voxel into consideration, i.e., the sparse geometric pattern and the varying density.

Specifically, We design various geometric regions with different shapes and ranges (as shown in the Fig. 2). Then we assign them to each codebook element as its attention spatial support. These geometric regions are made up by combining M distinct shapes and D different dilations
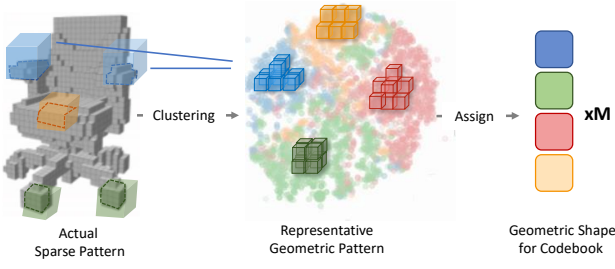
Figure 3. **Illustration of the design of geometric regions.** We conduct clustering on sparse patterns, and generate representative geometric pattern. The clustering is conducted respectively for different dataset/stride/dilation. These geometric regions are then adopted as the attention spatial support for each codebook element.

($M \times D$ types of attention spatial supports in total), corresponding to the varying geometric patterns and densities of 3D data respectively. And our model can learn to adapt the weights of these codebook elements with varying attention spatial supports.

M distinct shapes As illustrated in Fig. 3, we acquire M representative sparse patterns by applying the K-mode [7, 11] clustering algorithm on sparse patterns. The clustering generates M centroids, and we choose them as the shapes of geometric regions. This process could be interpreted as "decomposing" the full cubic region to the combination of a few representative sparse patterns. Instead of using a single cubic kernel for all voxels with different geometric patterns, we decouple the parameter learning of different geometric patterns and make the attention adaptive to the sparse pattern of voxels. Notice modern semantic segmentation networks usually consist of spatial pooling to quickly increase the receptive field, resulting varying spatial resolutions. We conduct clustering at each spatial resolution to obtain the respective geometric patterns.

D different dilations We adopt the above-mentioned clustering on D dilations and obtain $M \times D$ regions of different shapes and ranges. The flexible ranges are critical for handling the varying density issue. If the attention feature and relation learning is constrained to a fixed local region, the voxels in low density regions may have few neighboring voxels and fail to aggregate features. After introducing different dilations, these voxels have the chance to choose the geometric region from a longer range for proper neighborhood aggregation.

Computing the geometry-aware self-attention Aside from assigning various geometric regions and letting the

attention discover geometry-aware features on its own, we introduce explicit "geometric guidance" for attention learning, which can be described with the following equation:

$$
\begin{aligned}
\xi_{ij} &= \cap(\varphi(x), \varphi(\theta_{ij}))/N(\varphi(\theta_{ij})) \\
\boldsymbol{w}' &= \sigma(\{\sum_j^D \xi_{ij}/T\}_{i=1,...,M}) \\
&\otimes \sigma(\{\sum_i^M \xi_{ij}/T\}_{j=1,...,D}) \in \mathbb{R}^{M \times D} \\
\boldsymbol{w_f} &= \boldsymbol{w} \odot \boldsymbol{w}'
\end{aligned}
\tag{3}
$$

where θ_{ij} denotes the codebook element with the i-th geometric shape and the j-th dilation. To be more specific, we use a variable ξ to describe the relative "matching degree" between the geometric region $\varphi(\theta_{ij})$ and the sparse neighborhood $\varphi(x)$ of the voxel x. "$\cap$" calculates the intersection of these two regions, and the result is normalized by the geometric region size $N(\varphi(\theta_{ij}))$ to produce ξ. Then we apply the softmax with temperature T on ξ for both dimensions and generate $\boldsymbol{w}' \in \mathbb{R}^{M \times D}$ as "geometry-aware choice". The temperature T controls the "steepness" of the $\boldsymbol{w}'$ distribution, which stands for the "strength" of the explicit geometric guidance. Finally, we multiply $\boldsymbol{w} \in \mathbb{R}^{M \times D}$ ($\boldsymbol{w} \in \mathbb{R}^K$ is generated in Eqn. 2 and then reshaped into $\mathbb{R}^{M \times D}$) with $\boldsymbol{w}' \in \mathbb{R}^{M \times D}$ to generate the final "choice" $\boldsymbol{w_f}$. The "geometric guidance" explicitly enforces the attention to lean towards the geometric region that matches the sparse pattern of the voxel. For instance, the voxels on the floor should relate more closely with the "plane" shaped region, and voxels with low density tend to choose the region with larger dilation. It explicitly models the geometric pattern of voxels in space which CNN and transformer neglect, and leverage the geometric information to guide self-attention learning.

4. Experiments

In this section, we evaluate our CodedVTR on the popular indoor and outdoor 3D semantic segmentation datasets: ScanNet [5] and the SemanticKITTI [1]. We give detailed descriptions of the implementation details about our CodedVTR and present its performance on these datasets. For each dataset, we first briefly introduce the dataset and evaluation metrics. Then we compare the performance of our CodedVTR with the sparse voxel CNN and transformer and demonstrate its superior performance. We also show that CodedVTR could be embedded into existing sparse convolution-based methods and further improve their performance from the aspect of architectural design.

Dataset	Method (Model)		Param	mIOU
ScanNet	Convolution	Minkowski-M [18]	7M	67.3%
		Minkowski-L [3]	11M	72.4%
	Transformer	PointTransformer [30]†	6M	58.6%
		VoTR (Mink-M) [17]†	7M	62.5%
		VoTR (Mink-L) [17]†	11M	66.1%
		CodedVTR (Mink-M)	7M	**68.8%**
		CodedVTR (Mink-L)	11M	**73.0%**
SemanticKITTI	Convolution	Minkowski-M [21]	7M	58.9%
		Minkowsk-L [21]	11M	61.1%
		SPVCNN [21]	8M	60.7%
	Transformer	VoTR (Mink-M) [17] †	7M	56.5%
		VoTR (Mink-L) [17]†	11M	58.2%
		CodedVTR (Mink-M)	7M	**60.4%**
		CodedVTR (Mink-L)	11M	**63.2%**
		CodedVTR (SPVCNN)	8M	**61.8%**
Nuscenes	Convolution	Minkowski-M [21]	7M	66.5%
		Minkowsk-L [21]	11M	69.4%
	Transformer	CodedVTR (Mink-M)	7M	**69.9%**
		CodedVTR (Mink-L)	11M	**72.5%**

Table 1. 3D Semantic Segmentation peformance on ScanNet and SemanticKITTI val set. "†" denotes the result is reproduced due to absence in the original paper. "M/L" denotes the model has similar depth and width like "MinkowskiResNet20/42".

4.1. Implemention Details

For compatibility with current backbones, we adopt the U-Net-like block layout like the well-known Minkowsk-iNet [3] (as the upper part of the Fig. 2) and design a CodedVTR block to replace ResNet-like sparse convolution block. We also adapt the VoTR [17] to our architecture as the voxel transformer baseline. In Table. 1, the "M" or "L" suffixes mark the medium-sized or large-sized models, which have similar depth and width like MinkowskiResNet-20 and MinkowskiResNet-42. "CodedVTR-X" denotes embedding CodedVTR into other convolution-based methods.

We supplement some details about our CodedVTR block as follows. We take the feature within the local neighborhood φ and generate the attention at the corresponding location. More specifically, the $\boldsymbol{f}$ in $\boldsymbol{f}(\varphi(x_i))$ consists of a linear layer and a lightweight channel-wise convolution for local aggregation within $\varphi(x_i)$ like previous attention-based methods [25]. The local aggregation already models the spatial information thus the positional encoding could be discarded [26]. The design of the geometric regions is conducted by applying k-modes [7] clustering on the sparse patterns of voxels from 10 randomly sampled scenes within $3 \times 3 \times 3$ cubic neighborhood for each stride. The hyper-parameter M is chosen through investigating the saturate point of the clustering cost function and set as 8. The above-mentioned clustering is applied on three dilations respectively, resulting in 8×3 geometric regions for the codebook.

4.2. Performance on the Indoor Segmentation Dataset

We conduct experiments on indoor 3D semantic segmentation dataset, ScanNet [5], which contains various 3D indoor scenes annotated with 20 semantic classes. We follow the common setting and split 1201, 312 for train and validation respectively. We follow the training protocol of MinkowskiNet [3] for fair comparison. As shown in Table. 1, replacing the ResNet-like sparse convolution block with our proposed CodedVTR block could bring consistent performance improvement. The middle-sized CodedVTR-M model achieves 1.5% higher mIoU (68.8%/67.3%) with $2/3$ param size (6.1M/9.2M), the large-sized CodedVTR-L outperforms the convolution-based counterpart (+0.6%) with half the param size (25.7M/40.2M). Besides, due to the relatively small dataset size, the voxel transformer (VoTR) suffers from severe generalization problem and fail to achieve comparable performance with the CNN (-6%). It shows the effectiveness of our CodedVTR on improving the 3D transformer's generalization ability.

4.3. Performance on the Outdoor Segmentation Dataset

For the outdoor scene semantic segmentation experiment, we choose to use a larger dataset, SemanticKITTI [1], which consists of LIDAR scans of outdoor scenes. Among them 19130 scenes are for training, and 4071 are for validation. There are 19 semantic classes to evaluate. We
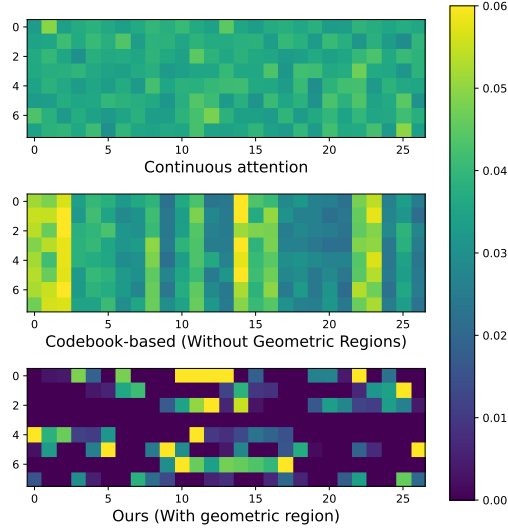
Figure 4. **Illustration of the attention maps of deeper layers for different self-attention design.** For both the "continuous attention" and "codebook-based attention without geometric regions", the attention maps collapse in latter layers (i.e., become uniform and fail to provide valuable information).

Method	Details	mIoU
VoxelTR†	-	56.5%
SAM [2]†	Optimizer	56.6%
ConvViT [6]†	Conv TR fusion	57.4%
CodedVTR	Codebook Only	58.1%
CodedVTR	Codebook + Geo-Region	58.9%
CodedVTR	Codebook + Geo-Guide	**60.4%**

Table 2. **Ablation studies of CodedVTR proposed techniques and comparison of other approaches for improving transformer's generalization ability on SemanticKITTI.** "Geo-region" denotes that we only assign different geometric regions for codebook elements. "Geo-guidance" represents applying the explicit encouragement.

also keep our training settings the same as the baseline method [21] and present the results in Table 1. We observe that CodedVTR consistently outperforms both the CNN and transformer models with similar sizes (by 1.5%/3.9% for middle-sized model and 2.1%/5.0% for large-sized model). Since SemanticKITTI is a relatively larger-scale dataset than ScanNet, the generalization problem of transformer is moderate. However, it still struggles to outperform convolution (-2%). Moreover, SPVCNN [21] notably outperforms (1.8%) Minkowski-M, and replacing the voxel branch in SPVCNN with our CodedVTR block could further boost its performance by 1.1%. It shows that our CodedVTR block could be easily embedded into the mainstream sparse convolution-based methods and further improve their performance from the aspect of architectural design.

5. Analysis and Discussions

5.1. Analysis of the Attention Learning

Comparison with methods for transformer's generalization improvement The main purpose of our CodedVTR is to alleviate the transformer's generalization problem in 3D domain. There exists many attempts for improving the transformer's generalization ability. We introduce them into the sparse voxel transformer and compare our CodedVTR with them. We compare two representative approaches from the aspects of the optimizer (SAM [2]) and architectural design ConViT [6] respectively. As shown in Table. 2, these

approaches do not effectively address the problem as CodedVTR. The reason for their unsatisfactory performance is that they are designed for general-purpose vision transformer. Therefore, they did not take 3D data's unique properties into consideration, thus failing to exploit the rich geometric information to assist the attention learning.

Visualization of the attention map Prior literature [32] points out that deeper transformers suffer from the "attention collapse" problem, in which the attention maps in latter layers tend to be uniform and fail to provide valuable information. We witness similar phenomenon for 3D voxel transformer, as shown in the upper heatmap of Fig. 4. Introducing the codebook-based self-attention alleviates this problem to some extent. However, in deeper layers, we still witness the similar "collapse" problem where codebook weights grow similar and lose expressivity. After we introduce the geometry-aware attention, the "collapse" problem disappears. We explain this phenomenon in two ways: 1) The geometric guidance helps the attention weight learning match the distinct geometric patterns. 2) The geometric region itself can be viewed as a form of diversity regularization on attention weights in terms of applying a hard mask on the attention weights.

5.2. Ablation studies of CodedVTR

Ablation studies of the proposed techniques We conduct ablation studies in Table. 2. The "Codebook only" denotes codebook-based self-attention without various geometric regions, it alleviates the generalization problem and matches the performance of convolution. However, as shown in Fig. 4, the codebook-based self-attention alone could not completely address the problem. The "Codebook+Geo-Region" denotes that we assign distinct geometric regions to each codebook element and solely rely on attention learning. Finally, the "Codebook+Geo-Guide" denotes our final scheme with all the techniques.

Codebook Design		mIoU
D	M	
1	1	57.1%
3	1	58.5%
4	1	55.3%
3 (RS)	8 (RS)	54.2%
3	8	**60.4**%
3	16	58.1%

Table 3. Ablation study: Comparison of performance of CodedVTR-A with different codebook designs on semKITTI. Noted that the geometric regions are applied in this experiment. The "D & M" stands for different dilations and sparse patterns.
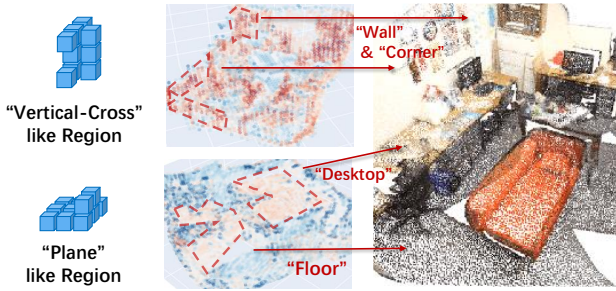
Figure 5. **Visualization of the geometry-aware self-attention's soft "choice" for certain geometric shape** (Red denotes larger value). The attention learns to use "plane" shaped region for table-top and floor for ScanNet scene.

Ablation studies of the codebook design As we state in Sec. 3.2, the design of the codebook is critical for our codebook-based self-attention to strike a balance between the generalization ability and expressivity, Therefore, We also conduct ablation studies of the codebook design as shown in Table. 3. When the codebook has only one element, it behaves like convolution as expected. When we properly enlarge the size of M and D, the model's expressivity and performance improves. However, too large codebook (e.g., D/M=3/16 or D/M=4/1) size burdens the optimization and causes performance degradation. Besides, we compare our learned attention with random sampling for the same codebook size. We witness significant performance differences (+6.2%), which demonstrates the effectiveness of our attention learning.

5.3. Visualization

CodedVTR learns to adapt to geometric patterns We visualize the "choice" value with geometric regions for a certain geometric pattern of the whole scene. As shown in Fig. 5, the "plane-shaped" geometric region relates more closely with the floor and the desktop region. Similarly, the
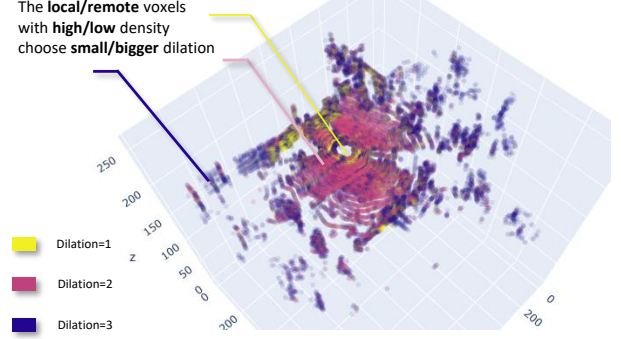
Figure 6. **Visualization of the geometry-aware self-attention's "choice" for different dilations.** (the color represents the best choice out of 3 dilations for certain voxel). The attention learns to use smaller dilation for local region (yellow) and larger ones for farther voxels (purple).

"vertical cross" shaped region adapts to the wall corners. It demonstrates that the geometric regions successfully encourage the attention to adapt to the voxel's sparse pattern.

CodedVTR learns to adapt to varying densities We visualize the best "choice" along the dilation dimension of the codebook. As shown in Fig. 6, CodedVTR learns to use the yellow ones for the local region with high density, and region with larger receptive field for remote, low-density points. In this way, CodedVTR handles the varying densities and ensures proper feature aggregation for all voxels.

6. Conclusions

This paper proposes CodedVTR, which seeks to address the generalization problem of transformer for 3D scene understanding. Specifically, we propose the "codebook-based" attention to project the attention space into a learnable subspace represented by the codebook. This technique can be seen as a a sort of regularization to improve the generalization of self-attention. Besides, we propose "geometry-aware attention" that assigns various geometric shapes to each codebook element and leverages the geometric feature to guide the attention learning. Extensive experiments on both indoor and outdoor 3D semantic segmentation datasets show that our CodedVTR could achieve better performance compared with both CNN and transformer.

Acknowledgement

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9296–9306, 2019. 2, 5, 6

[2] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pretraining or strong data augmentations. *ArXiv*, abs/2106.01548, 2021. 3, 7

[3] 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. 1, 2, 6

[4] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *ArXiv*, abs/1911.03584, 2020. 2

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 2, 5, 6

[6] Stéphane d'Ascoli, Hugo Touvron, Matthew L. Leavitt, Ari S. Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, 2021. 3, 7

[7] Nelis J. de Vos. kmodes categorical clustering library. https://github.com/nicodv/kmodes, 2015–2021. 5, 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 2

[9] Zhengyang Geng, Meng-Hao Guo, Hongxu Chen, Xia Li, Ke Wei, and Zhouchen Lin. Is attention better than matrix decomposition? *ArXiv*, abs/2109.04553, 2021. 3

[10] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 1, 2

[11] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998. 5

[12] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[13] Shanda Li, Xiangning Chen, Di He, and Cho-Jui Hsieh. Can vision transformers perform convolution? *ArXiv*, abs/2111.01353, 2021. 2, 3

[14] Xiang Li, Lingjing Wang, Mingyang Wang, Congcong Wen, and Yi Fang. Dance-net: Density-aware convolution networks with context encoding for airborne lidar point cloud classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:128–139, 2020. 2

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Ching-Feng Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ArXiv*, abs/2103.14030, 2021. 1

[16] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3d point cloud understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[17] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. *ArXiv*, abs/2109.02497, 2021. 2, 3, 6

[18] Alexey Nekrasov, Jonas Schult, Or Litany, B. Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. *ArXiv*, abs/2110.02210, 2021. 6

[19] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *CVPR*, 2021. 3

[20] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1, 2, 3

[21] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020. 1, 6, 7

[22] Hugues Thomas, C. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419, 2019. 1, 2

[23] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2, 3

[24] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017. 1, 2

[25] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 6

[26] Haiping Wu, Bin Xiao, Noel C. F. Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *ArXiv*, abs/2103.15808, 2021. 3, 6

[27] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross B. Girshick. Early convolutions help transformers see better. *ArXiv*, abs/2106.14881, 2021. 3

[28] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. *ArXiv*, abs/2108.08839, 2021. 3

[29] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10073–10082, 2020. 3

[30] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. *ArXiv*, abs/2012.09164, 2020. 1, 3, 6

[31] Yucheng Zhao, Guangting Wang, Chuanxin Tang, Chong Luo, Wenjun Zeng, and Zhengjun Zha. A battle of network structures: An empirical study of cnn, transformer, and mlp. *ArXiv*, abs/2108.13002, 2021. 2

[32] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *ArXiv*, abs/2103.11886, 2021. 7

[33] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. 1, 2