CodedVTR: **Code**book-based Sparse **V**oxel **T**Ransformer with Geometric Guidance

Tianchen Zhao, Niansong Zhang, Xuefei Ning, He Wang, Li Yi*, Yu Wang

*Corresponding Author*
Background

- **Transformers outperform CNN** and achieve SOTA in many vision tasks
- Transformer’s superiority:
  - Less inductive bias -> Better representation power

Performance of ImageNet1K Classification

---

Background

- **Transformer’s Problem:** harder to **optimize** and **generalize**
  - Rely on large-scale pretraining, **overfits** when directly trained on smaller dataset
  - Slow convergence & sensitive to hyperparameters (LR, initialization, data-aug...)

- **Solution:** introduce **domain-specific inductive bias**

  “When directly trained on the ImageNet, ViT yields modest accuracies of a few points below ResNets of comparable size” [1]

ViT[1] requires large-scale pretraining to outperform ResNets

Swin Transformer[2] uses **hierarchical window-based local** aggregation

---


Background

- Introducing transformer into 3D Domain: The **generalization issue** is aggravated
  - 3D Data (Sparse Voxel) has unique properties (sparse & irregular)
  - Relatively limited data scale

Voxel’s unique properties: sparse and irregular

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method (Model)</th>
<th>Params</th>
<th>mIOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScanNet</td>
<td>Convolution</td>
<td>Minkowski-M</td>
<td>7M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minkowski-L</td>
<td>11M</td>
</tr>
<tr>
<td>Transformer</td>
<td>PointTransformer</td>
<td>6M</td>
<td>58.6%</td>
</tr>
<tr>
<td></td>
<td>VoTR (Mink-M)</td>
<td>7M</td>
<td>62.5%</td>
</tr>
<tr>
<td></td>
<td>VoTR (Mink-L)</td>
<td>11M</td>
<td>66.1%</td>
</tr>
<tr>
<td>SemanticKITTI</td>
<td>Convolution</td>
<td>Minkowski-M</td>
<td>7M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minkowski-L</td>
<td>11M</td>
</tr>
<tr>
<td>Transformer</td>
<td>VoTR (Mink-M) †</td>
<td>7M</td>
<td>56.5%</td>
</tr>
<tr>
<td></td>
<td>VoTR (Mink-L)</td>
<td>11M</td>
<td>58.2%</td>
</tr>
</tbody>
</table>

Simply employ transformer fails to outperform CNN
**Contribution**

- **Key:** alleviate the generalization issue
- **CodedVTR:** introduce **geometric-aware codebook**
  - Codebook-based attention
  - Geometric-aware attention
Motivation

- **Comparison of conv and transformer** (with local self-attention)
  - The attention weight generation:

\[
\gamma = f(x)
\]

\[
y = \sum_i \gamma_i \cdot g(x_i)
\]

**Self-attention**

**Convolution**

Learn the whole continuous space \( f(x) \) of dimension: \( N(\Phi) \)

Learn some weights \( w_i \) on anchor points
Methodology

- **Codebook-based attention**: encode the attn-weight with codebook elements

\[
    f(x) \sim f_d(x) = \sum w_i \theta_i
\]

- Codebook elements \( \theta_i \) could be viewed as:
  - attention weight "Prototypes"
  - a set of basis that span a subspace

- **Project** the attention learning in the subspace

- **Regularization** – helps generalization

Learn the subspace \( f_d(x) \) of dim: \( N(\Theta) < N(\Phi) \)
Methodology

- Codebook-based attention is an **intermediate state** of self-attention and convolution.

\[
\gamma = f(x)
\]

\[
\Theta = \infty
\]

**Self-Attention**

\[
\gamma = f_d(x) = \sum w_i \theta_i
\]

**Codebook-based Self-Attention**

\[
\gamma = w_i
\]

**Convolution**
Methodology

- Codebook Design

![Diagram showing Codebook-based Attention](https://example.com/diagram.png)
Experimental Results

- Results of Codebook Design

<table>
<thead>
<tr>
<th>$N(\Phi)$</th>
<th>mIOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57.1%</td>
</tr>
<tr>
<td>4</td>
<td>58.5%</td>
</tr>
<tr>
<td>24</td>
<td>60.4%</td>
</tr>
<tr>
<td>48</td>
<td>58.1%</td>
</tr>
</tbody>
</table>

CodedVTR helps generalization
Methodology

- 3D data’s unique properties -> Geometric-aware attention

- Sparsity & Varying geometric shape

- Varying densities (outdoor)
Methodology

- Geometric-aware attention
  - Geo-shape: assign different geometric shapes for codebook elements
  - Geo-guide: encourage attention to match the input’s sparse pattern

Same convolution weight

Different input voxel sparse pattern

Different geometric shapes codebook-elements
Methodology

- How to determine geometric shape?
  - Adopt **k-means clustering** to get 8 representative sparse pattern in 3 dilations
Methodology

- How to adopt geometric guide?
  - Regularize the attention with “mismatch code”

Encourage the attention
To lean to matching geometric shape
Methodology

- How to adopt geometric guide?
  - Regularize the attention with “mismatch code”

Adaptively choosing the **Receptive Field** for different densities
Experimental Results

Naïve self-attention
(Uniform Attention Map)

Codebook-based self-attention
(Similar attention map)

Geometric-aware self-attention
(Meaningful attention map)
# Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method (Model)</th>
<th>Params</th>
<th>mIOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScanNet</td>
<td>Convolution</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minkowski-M</td>
<td>7M</td>
<td>67.3%</td>
</tr>
<tr>
<td></td>
<td>Minkowski-L</td>
<td>11M</td>
<td>72.4%</td>
</tr>
<tr>
<td></td>
<td>Transformer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CodedVTR (Mink-M)</td>
<td>7M</td>
<td>68.8% (+1.5%)</td>
</tr>
<tr>
<td></td>
<td>CodedVTR (Mink-L)</td>
<td>11M</td>
<td>73.0% (+0.6%)</td>
</tr>
<tr>
<td>SemanticKITTl</td>
<td>Convolution</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minkowski-M</td>
<td>7M</td>
<td>58.9%</td>
</tr>
<tr>
<td></td>
<td>Minkowski-L</td>
<td>11M</td>
<td>61.1%</td>
</tr>
<tr>
<td></td>
<td>SPVCNN</td>
<td>8M</td>
<td>60.7%</td>
</tr>
<tr>
<td></td>
<td>Transformer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CodedVTR (Mink-M)</td>
<td>7M</td>
<td>60.4% (+0.5)</td>
</tr>
<tr>
<td></td>
<td>CodedVTR (Mink-L)</td>
<td>11M</td>
<td>63.2% (+2.1%)</td>
</tr>
<tr>
<td></td>
<td>CodedVTR (SPVCNN)</td>
<td>8M</td>
<td>61.8% (+1.1%)</td>
</tr>
<tr>
<td>Nuscenes</td>
<td>Convolution</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minkowski-M</td>
<td>7M</td>
<td>66.5%</td>
</tr>
<tr>
<td></td>
<td>Minkowski-L</td>
<td>7M</td>
<td>69.4%</td>
</tr>
<tr>
<td></td>
<td>Transformer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CodedVTR (Mink-M)</td>
<td>7M</td>
<td>69.9% (+3.4%)</td>
</tr>
<tr>
<td></td>
<td>CodedVTR (Mink-L)</td>
<td>11M</td>
<td>72.5% (+3.1%)</td>
</tr>
</tbody>
</table>