# Supplementary Materials: M3DETR: Multi-representation, Multi-scale, Mutual-relation 3D Object Detection with Transformers

Tianrui Guan[1*] Jun Wang[1*] Shiyi Lan[1†] Rohan Chandra[1] Zuxuan Wu [2]
Larry Davis [1] Dinesh Manocha[1]
[1]University of Maryland, College Park [2]Fudan University
{rayguan, sylan, rohan}@cs.umd.edu, {junwang,lsd}@umiacs.umd.edu,
zxwu@fudan.edu.cn, dmanocha@umd.edu

## 1. More Our Approach Details

We further discuss our approach in the following.

### 1.1. Voxel Representation in Multi-Representation Feature Embeddings

For the voxel-wise feature extraction from raw point clouds input, there are two steps, voxelization using voxelization layer and feature extraction using 3D sparse convolutions. We denote the size of each discretized voxel as $L \times W \times H \times C$, where $L, W, H$ indicate the length, width, and height of the voxel grid and $C$ represents the channel of the voxel features. We adopt the average of the point-wise features from all the points to represent the whole non-empty voxel feature. After voxelization, the input feature is propagated through a series of $3\times3\times3$ sparse cubes, including four consecutive blocks of 3D sparse convolution with downsampled sizes of $1\times$, $2\times$, $4\times$, $8\times$, using convolution operations of stride 2. Specifically, each sparse convolutional block includes a 3D convolution layer followed by a LayerNorm layer and a ReLU layer.

### 1.2. Multi-head Self-attention Basics

Building on the attention mechanism, Multi-head Self-attention (MHSA) with $N$ heads and the input matrix $\boldsymbol{X}$ is defined as follows:

$$\text{MHSA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concatenate}[\text{head}_1, ..., \text{head}_N]\boldsymbol{W}^O,$$

where $\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^q, \boldsymbol{K}\boldsymbol{W}_i^k, \boldsymbol{V}\boldsymbol{W}_i^v)$ and $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are linear projections of X. The projection matrices $\boldsymbol{W}_i^q \in \mathbb{R}^{d_h \times d_q}$, $\boldsymbol{W}_i^k \in \mathbb{R}^{d_h \times d_k}$, $\boldsymbol{W}_i^v \in \mathbb{R}^{d_h \times d_v}$, $\boldsymbol{W}^O \in \mathbb{R}^{d_v \times d_{out}}$ are learnable parameters in the network that correspond to each of the attention heads, where $d_k$, $d_v$ and $d_q$ are the hidden dimensions of each attention head for $\boldsymbol{K}$, $\boldsymbol{V}$, and $\boldsymbol{Q}$, respectively.

## 2. More Implementation Details

As mentioned in the Section 4.2, we give more details on our implementation for reproduction of our result. Our code will also be released later, including trained models that can match the performance that was included in the paper.

### 2.1. Backbone

The 3D voxel CNN branch consists of 4 blocks of 3D sparse convolutions with output feature channel dimensions of 16, 32, 64, 64. Those 4 different voxel representations of different scales, as well as point features from point cloud input, are used to refine keypoint features by PointNet [1] through set abstraction and voxel set abstraction [2]. The number of sampled keypoints $n$ is 2,048 for both Waymo and KITTI. In order to sample the keypoints effectively and accurately, we uses FPS on points within the range of top 1000 or 1500 initial proposals with a radius $r$ of 2.4.

**Waymo:** The voxel size is [0.1, 0.1, 0.15], and we focus on the input LiDAR point cloud range with [-75.2, 75.2], [-75.2, 75.2], and [-2, 4] meters in x, y, and z axis, respectively. The 2D ConvNets output size is $188 \times 188 \times 512$.

**KITTI:** The voxel size is [0.05, 0.05, 0.1], and we focus on the input LiDAR point cloud range with [0, 70.4], [-40, 40], and [-3, 1] meters in x, y, and z axis, respectively. The 2D ConvNets output size is $200 \times 176 \times 512$.

### 2.2. Detection Heads

The RPN anchor size for each object category is set by computing the average of the corresponding objects from the annotated training set. The RoI-grid pooling module samples $6 \times 6 \times 6$ grid points within each initial 3D proposal to form a refined features. The number of surrounding points used to extract the grid point's feature, M, is 16.

*Equal contribution.
†Corresponding author: <sylan@cs.umd.edu>.

During the training phase, 512 proposals are generated from RPN to R-CNN, where non-maximum suppression (NMS) with a threshold of 0.8 is applied to remove the overlapping proposals. In the validation phase, 100 proposals are fed into R-CNN, where the NMS threshold is 0.7.

## References


[1] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[2] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.