1. Network Specifics

In this section, we present architectural specifics of our proposed Semantic Superpoint Tree Network (SSTNet).

1.1. Backbone and Learning Branches

Fig. 2-(a) illustrates the architecture of our backbone, where we employ a U-Net [4] style network with a depth of 5. Fig. 2-(b) illustrates the branch specifics of semantic scoring and offset prediction.

1.2. The Classifier for Tree Traversal and Splitting

Fig. 3 presents the multi-layer perceptron (MLP) of the binary classifier $\phi$ that is used for generation of object proposals, where we also show how the classifier is used when traversing the tree.

1.3. CliqueNet

An illustration on how a tree branch can be converted as a graph clique and the thus constructed CliqueNet.
Figure 2: Module specifics of the backbone, semantic scoring, and offset prediction used in SSTNet. $N$ is the number of input points, and numbers in each block denote those of output channels.

Figure 3: An illustration on the node-splitting classifier $\phi$ and how it is used when traversing the semantic superpoint tree.

Figure 4: An illustration on how a tree branch can be converted as a graph clique and also the construction of CliqueNet $\psi$. Fig. 4 illustrates how a tree branch can be converted as a graph clique and also the construction of CliqueNet $\psi$. Given an input feature $F_C^i$, the $i^{th}$ layer of the CliqueNet (i.e., the $i^{th}$ CliqueLayer) performs the following computation

$$ReLU(\overline{D}_C^{-1/2} \overline{A}_C \overline{D}_C^{-1/2} F_C^i W^i_{\psi})$$

(1)

where the adjacency matrix $A_C$ is shown in Fig. 4-(a), $A_C = A_C + I$, and $\overline{D}_C$ is the diagonal degree matrix of $A_C$. CliqueNet specifics are given in Fig. 4-(b).

2. Training of the Proposal Evaluation Module

We follow [3] and use a ScoreNet (denoted as $\omega$) to evaluate the proposals refined by CliqueNet. For such a proposal $B_t^i$, we get the corresponding point-wise features $\tilde{F}_{B_t^i} = [\tilde{f}_1, \ldots, \tilde{f}_{N_t}] \in \mathbb{R}^{n \times N_t}$, and use the following...
loss to train the ScoreNet

\[ L_{\text{evaluation}} = \frac{1}{|R|} \sum_{t \in R} \text{BCE}(\omega(\tilde{F}^t_{B_t}), v^*_t), \]  

(2)

where the value \( v^*_t \) of supervision used in binary cross-entropy loss (BCE) is determined by the Intersection over Union (IoU) between the proposal \( B_t \) and its best matched ground-truth instance; we denote the IoU value as \( \text{IoU}_{B_t} \). Given \( \text{IoU}_{B_t} \), \( v^*_t \) is determined as

\[ v^*_t = \begin{cases} 
0 & \text{if } \text{IoU}_{B_t} < \theta_l \\
1 & \text{if } \text{IoU}_{B_t} > \theta_h \\
\frac{1}{\theta_h - \theta_l}(\text{IoU}_{B_t} - \theta_l) & \text{otherwise}
\end{cases} \]  

(3)

where we set the hyperparameters \( \theta_l = 0.25 \) and \( \theta_h = 0.75 \).

3. Results Comparison Visualization

In this section, we show more comprehensive comparisons with 3D-MPA[2], SSEN[5] and PointGroup[3] on ScanNet\( (V2)[1] \). As shown in Fig. 5, our results can better maintain the boundaries and the integrity of the segmentation results.

References


of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9031–9040, 2020. 3

