Supplementary Materials for
"VI-Net: Boosting Category-level 6D Object Pose Estimation via Learning Decoupled Rotations on the Spherical Representations"

A. Network Specifies

For the estimation of translation and size, we employ the PointNet++ [1] with mulit-scale grouping to extract hierarchical features for making point-wise predictions, which are averaged as the final results; the network specifies are given in Fig. 1(a). Please refer to [1] for more details.

For the estimation of rotation, we desgin VI-Net with three main modules, including Spherical FPN, V-Branch and I-Branch, whose network specifies are all given in Fig. 1(b).

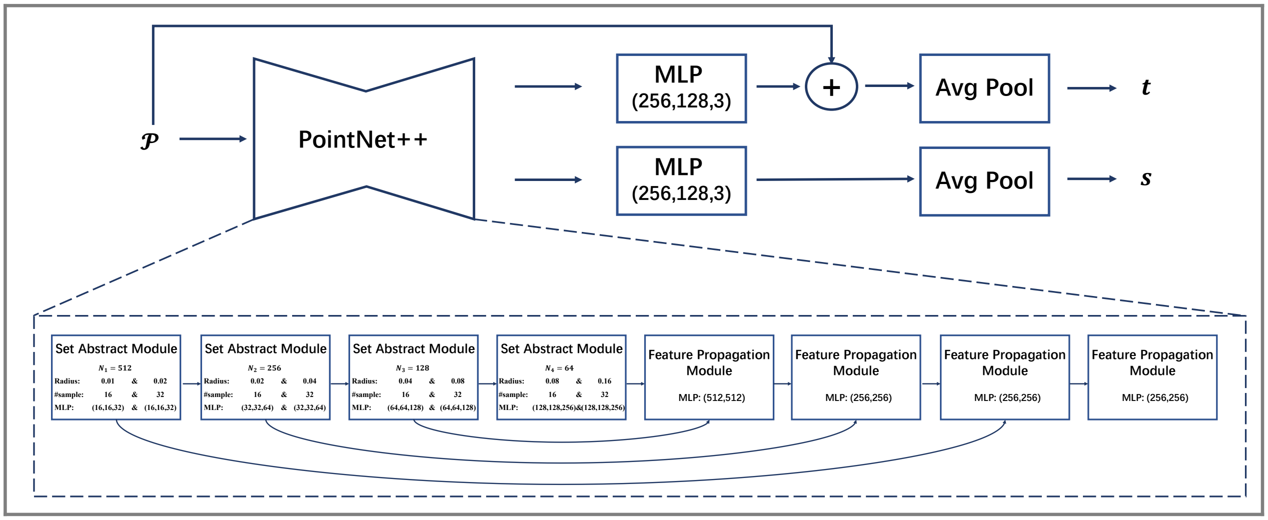
B. Implementation of SPA-SConv

To conduct continuous convolutions on the sphere, we introduce the design of Spatial Spherical Convolution (SPA-SConv) in Sec. 4.3, and also include the whole process of SPA-SConv in Algorithm 1.

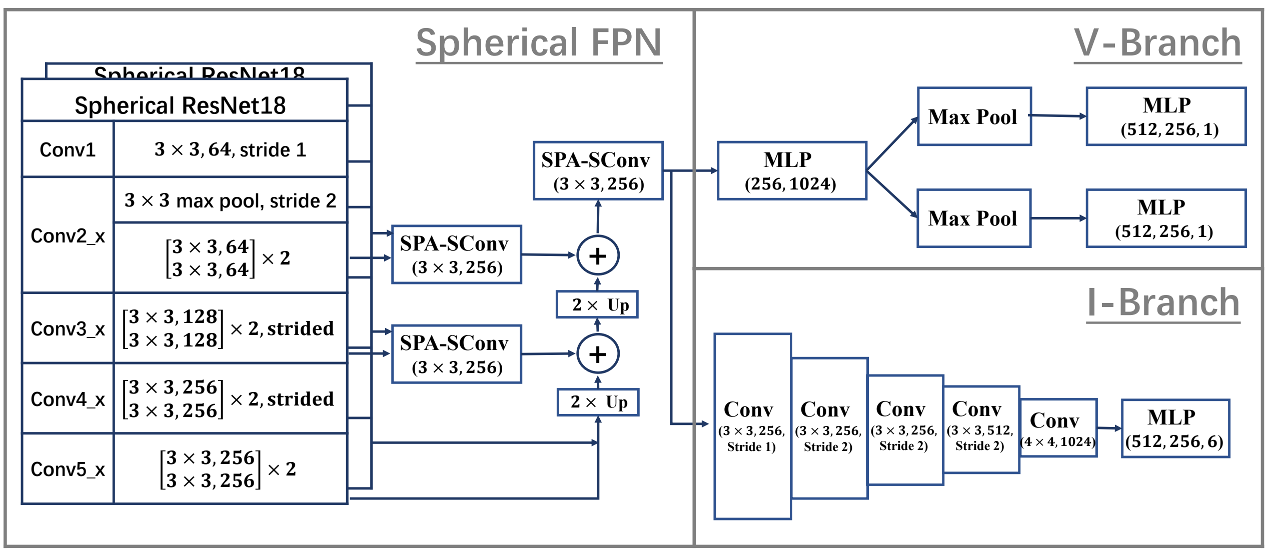
Algorithm 1 Spatial Spherical Convolution.

Input: spherical feature $\mathcal{S}_l \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$, kernel size K, stride s, and output feature channel C_{l+1}, with batch size B, input feature channel C_l, resolution $H_l \times W_l$.
Output: spherical feature $\mathcal{S}_{l+1} \in \mathbb{R}^{B\times C_{l+1}\times H_{l+1}\times W_{l+1}}$, with resolution $H_{l+1} \times W_{l+1}$.

$P = K//2$
Initialize $\mathcal{S}_l^{pad} \in \mathbb{R}^{B\times C_l\times (H_l+2P)\times(W_l+2P)}$
for $h = 1$ to H_l **do**
  for $w = 1$ to W_l **do**
    $\mathcal{S}_l^{pad}[:,:,h+P,w+P] = \mathcal{S}_l[:,:,h,w]$
for $p = 1$ to P **do**
  for $w = 1$ to W_l **do**
    if $w \leq W_l/2$ **then**
      $\mathcal{S}_l^{pad}[:,:,p,w+P] = \mathcal{S}_l^{pad}[:,:,2P-p+1,w+W_l/2+P]$
      $\mathcal{S}_l^{pad}[:,:,H_l+P+p,w+P] = \mathcal{S}_l^{pad}[:,:,H_l+P-p+1,w+W_l/2+P]$
    else
      $\mathcal{S}_l^{pad}[:,:,p,w+P] = \mathcal{S}_l^{pad}[:,:,2P-p+1,w-W_l/2+P]$
      $\mathcal{S}_l^{pad}[:,:,H_l+P+p,w+P] = \mathcal{S}_l^{pad}[:,:,H_l+P-p+1,w-W_l/2+P]$
for $p = 1$ to P **do**
  for $h = 1$ to $H_l + 2P$ **do**
    $\mathcal{S}_l^{pad}[:,:,h,p] = \mathcal{S}_l^{pad}[:,:,h,W_l+p]$
    $\mathcal{S}_l^{pad}[:,:,h,W_l+P+p] = \mathcal{S}_l^{pad}[:,:,h,P+p]$
Initialize the 2D convolution `Conv` with the kernel weight $\kappa_l \in \mathbb{R}^{C_{l+1}\times C_l\times K\times K}$ and stride s
$\mathcal{S}_{l+1,1} = \texttt{Conv}\ (\mathcal{S}_l^{pad}; \kappa_l)$
$\mathcal{S}_{l+1,2} = \texttt{Conv}\ (\mathcal{S}_l^{pad}; \texttt{Flip}(\kappa_l))$ % `Flip` denotes horizontal flip
$\mathcal{S}_{l+1} = \texttt{Max}\ (\mathcal{S}_{l+1,1}, \mathcal{S}_{l+1,2})$ % `Max` denotes element-wise max-pooling
return $\mathcal{S}_{l+1}$



(a)



(b)

Figure 1. Network Specifies of (a) Pointnet++ for translation / size estimation and (b) our proposed VI-Net for rotation estimation.

C. Proof: SPA-SConv is Viewpoint-equivariant

We define the SPAtial Spherical Convolutions (SPA-SConv) as follows:

$$\begin{aligned} \mathcal{S}_{l+1} =& f(\mathcal{S}_l; \kappa_l) \\ =& \texttt{Max}(\texttt{Conv}(\mathcal{S}_l^{pad}; \kappa_l), \texttt{Conv}(\mathcal{S}_l^{pad}; \texttt{Flip}(\kappa_l))), \end{aligned} \tag{1}$$

where

$$\texttt{Conv}(\mathcal{S}_l(h,w); \kappa_l) = \sum_i \sum_j \kappa_l(i,j)\mathcal{S}_l(h+i, w+j), \tag{2}$$

$$\texttt{Conv}(\mathcal{S}_l(h,w); \texttt{Flip}(\kappa_l)) = \sum_i \sum_j \kappa_l(i,j)\mathcal{S}_l(h-i, w+j), \tag{3}$$

with $i \in \{K//2-K, ..., 0, ..., K//2\}$ and $j \in \{K//2-K, ..., 0, ..., K//2\}$. κ_l denoting the weight of the convolution and K is the kernel size. Conv, Flip, and Max denotes the 2D convolutional operation, horizontal flip, and element-wise max-pooling, respectively. Given the viewpoint rotation $\boldsymbol{R}_{vp} = \boldsymbol{R}_Z(\varphi)\boldsymbol{R}_Y(\theta)$, we claim that SPA-SConv is viewpoint-equivariant, that is,

$$\mathcal{T}\mathcal{S}_{l+1} = \mathcal{T}f(\mathcal{S}_l;\kappa_l) = f(\mathcal{T}\mathcal{S}_l;\kappa_l), \tag{4}$$

where $\mathcal{T}$ denotes the transformation w.r.t $\boldsymbol{R}_{vp}$ on the spherical features. In the following, we will prove the property of viewpoint-equivariance of SPA-SConv. For simplicity, $\mathcal{S}_{l+1}$ and $\mathcal{S}_l$ are assumed to share the same spatial sizes $H_l \times W_l$.

Firstly, considering $\boldsymbol{R}_{vp} = \boldsymbol{R}_Z(\varphi)$ with the feature transformation $\mathcal{T}_\varphi$, we have

$$\mathcal{T}_\varphi\mathcal{S}_l(h,w) = \begin{cases} \mathcal{S}_l(h, w-\Delta w+1), & \text{if} \quad \texttt{C1}: \Delta w \le w \\ \mathcal{S}_l(h, w-\Delta w+1+W_l), & \text{if} \quad \texttt{C2}: \Delta w > w \end{cases}, \tag{5}$$

where $\Delta w = \lfloor \varphi / W_l \cdot 2\pi \rfloor$, such that

$$\begin{aligned}
&\mathcal{T}_\varphi\mathcal{S}_{l+1}(h,w) \\
=&\begin{cases} S_{l+1}(h, w-\Delta w+1), & \text{if} \quad \texttt{C1} \\ S_{l+1}(h, w-\Delta w+1+W_l), & \text{if} \quad \texttt{C2} \end{cases} \\
=&\begin{cases} f(\mathcal{S}_l(h, w-\Delta w+1);\kappa_l), & \text{if} \quad \texttt{C1} \\ f(\mathcal{S}_l(h, w-\Delta w+1+W_l);\kappa_l), & \text{if} \quad \texttt{C2} \end{cases} \\
=&\begin{cases} \texttt{Max}(\texttt{Conv}(\mathcal{S}_l(h, w-\Delta w+1);\kappa_l), \texttt{Conv}(\mathcal{S}_l(h, w-\Delta w+1);\texttt{Flip}(\kappa_l))), & \text{if} \quad \texttt{C1} \\ \texttt{Max}(\texttt{Conv}(\mathcal{S}_l(h, w-\Delta w+1+W_l);\kappa_l), \texttt{Conv}(\mathcal{S}_l(h, w-\Delta w+1+W_l);\texttt{Flip}(\kappa_l))), & \text{if} \quad \texttt{C2} \end{cases} \\
=&\begin{cases} \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h+i, w-\Delta w+1+j), \sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h-i, w-\Delta w+1+j)), & \text{if} \quad \texttt{C1} \\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h+i, w-\Delta w+1+W_l+j), \sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h-i, w-\Delta w+1+W_l+j)), & \text{if} \quad \texttt{C2} \end{cases} \\
=&\texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\varphi\mathcal{S}_l(h+i, w+j), \sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\varphi\mathcal{S}_l(h-i, w+j)) \\
=&\texttt{Max}(\texttt{Conv}(\mathcal{T}_\varphi\mathcal{S}_l(h,w);\kappa_l), \texttt{Conv}(\mathcal{T}_\varphi\mathcal{S}_l(h,w);\texttt{Flip}(\kappa_l))) \\
=&f(\mathcal{T}_\varphi\mathcal{S}_l(h,w);\kappa_l).
\end{aligned} \tag{6}$$

Next, considering $\boldsymbol{R}_{vp} = \boldsymbol{R}_Y(\theta)$ with the feature transformation $\mathcal{T}_\theta$, we have

$$\mathcal{T}_\theta\mathcal{S}_l(h,w) = \begin{cases} \mathcal{S}_l(h-\Delta h+1, w), & \text{if} \quad \texttt{C3}: w \le W_l//2, \Delta h \le h \\ \mathcal{S}_l(\Delta h - h, w+W_l//2), & \text{if} \quad \texttt{C4}: w \le W_l//2, \Delta h > h \\ \mathcal{S}_l(h+\Delta h, w), & \text{if} \quad \texttt{C5}: w > W_l//2, \Delta h \le H_l - h \\ \mathcal{S}_l(2H_l-(h+\Delta h)+1, w-W_l//2), & \text{if} \quad \texttt{C6}: w > W_l//2, \Delta h > H_l - h \end{cases}, \tag{7}$$

where $\Delta h = \lfloor \theta / H_l \cdot \pi \rfloor$, such that

$$
\begin{aligned}
&\mathcal{T}_\theta \mathcal{S}_{l+1}(h,w)\\
=&\begin{cases} S_{l+1}(h-\Delta h+1,w), & \text{if } \texttt{C3}\\ S_{l+1}(\Delta h-h,w+W_l//2), & \text{if } \texttt{C4}\\ S_{l+1}(h+\Delta h,w), & \text{if } \texttt{C5}\\ S_{l+1}(2H_l-(h+\Delta h)+1,w-W_l//2), & \text{if } \texttt{C6}\end{cases}\\
=&\begin{cases} \texttt{Max}(\texttt{Conv}(\mathcal{S}_l(h-\Delta h+1,w);\kappa_l),\texttt{Conv}(\mathcal{S}_l(h-\Delta h+1,w);\texttt{Flip}(\kappa_l))), & \text{if } \texttt{C3}\\ \texttt{Max}(\texttt{Conv}(\mathcal{S}_l(\Delta h-h,w+W_l//2);\kappa_l),\texttt{Conv}(\mathcal{S}_l(\Delta h-h,w+W_l//2);\texttt{Flip}(\kappa_l))), & \text{if } \texttt{C4}\\ \texttt{Max}(\texttt{Conv}(\mathcal{S}_l(h+\Delta h,w);\kappa_l),\texttt{Conv}(\mathcal{S}_l(h+\Delta h,w);\texttt{Flip}(\kappa_l))), & \text{if } \texttt{C5}\\ \texttt{Max}(\texttt{Conv}(\mathcal{S}_l(2H_l-(h+\Delta h)+1,w-W_l//2);\kappa_l),\texttt{Conv}(\mathcal{S}_l(2H_l-(h+\Delta h)+1,w-W_l//2);\texttt{Flip}(\kappa_l))), & \text{if } \texttt{C6}\end{cases}\\
=&\begin{cases} \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h-\Delta h+1+i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h-\Delta h+1-i,w+j)), & \text{if } \texttt{C3}\\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(\Delta h-h+i,w+W_l//2+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(\Delta h-h-i,w+W_l//2+j)), & \text{if } \texttt{C4}\\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h+\Delta h+i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(h+\Delta h-i,w+j)), & \text{if } \texttt{C5}\\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(2H_l-(h+\Delta h)+1+i,w-W_l//2+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{S}_l(2H_l-(h+\Delta h)+1-i,w-W_l//2+j)), & \text{if } \texttt{C6}\end{cases}\\
=&\begin{cases} \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h+i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h-i,w+j)), & \text{if } \texttt{C3}\\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h-i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h+i,w+j)), & \text{if } \texttt{C4}\\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h+i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h-i,w+j)), & \text{if } \texttt{C5}\\ \texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h-i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h+i,w+j)), & \text{if } \texttt{C6}\end{cases}\\
=&\texttt{Max}(\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h+i,w+j),\sum_i\sum_j \kappa_l(i,j)\mathcal{T}_\theta\mathcal{S}_l(h-i,w+j))\\
=&\texttt{Max}(\texttt{Conv}(\mathcal{T}_\theta\mathcal{S}_l(h,w);\kappa_l),\texttt{Conv}(\mathcal{T}_\theta\mathcal{S}_l(h,w);\texttt{Flip}(\kappa_l)))\\
=&f(\mathcal{T}_\theta\mathcal{S}_l(h,w);\kappa_l).
\end{aligned}
\tag{8}
$$

Finally, for the general case of $\mathcal{T}=\mathcal{T}_\varphi\mathcal{T}_\theta$, we prove that

$$
\mathcal{T}\mathcal{S}_{l+1}=\mathcal{T}_\varphi\mathcal{T}_\theta\mathcal{S}_{l+1}=\mathcal{T}_\varphi(\mathcal{T}_\theta\mathcal{S}_{l+1})=\mathcal{T}_\varphi f(\mathcal{T}_\theta\mathcal{S}_l;\kappa_l)=f(\mathcal{T}_\varphi\mathcal{T}_\theta\mathcal{S}_l;\kappa_l)=f(\mathcal{T}\mathcal{S}_l;\kappa_l). \tag{9}
$$

References

[1] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.