Eurographics Symposium on Geometry Processing 2023
P. Memari and J. Solomon
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 5

Cross-Shape Attention for Part Segmentation of 3D Point Clouds – Supplementary Material –

Appendix A: Backbone architecture details

MinkHRNetCSN architecture details. In Table 1 we describe the overall Cross-Shape Network architecture for $K = 1$ key shapes per query shape, based on the HRNet [WSC*21] backbone ("MinkHRNetCSN-K1"). For $K = 2,3$ and SSA variants, we use the same architecture. First, for an input query-key pair of shapes $\mathcal{S}_m \in \mathcal{R}^{P_m \times 3}$ and $\mathcal{S}_n \in \mathcal{R}^{P_n \times 3}$, point-wise features $\boldsymbol{X}_m$ and $\boldsymbol{X}_n$ are extracted using the "Mink-HRNet" backbone (Layers 2 and 3). For each set of point features their self-shape attention representations are calculated via the *Cross-Shape Attention* layer (Layers 4 and 5). The query shape point self-shape attention representations are then aggregated into a global feature using mean-pooling and undergo two separate linear transformations (*Linear-Q* and *Linear-K* in Layers 6 and 7, respectively). Leveraging these, the self-shape similarity is computed, using the *scaled dot product* (Layer 9). For the key shape point self-shape attention representations, we use only the *Linear-K* transformation on the key shape's global feature (Layer 8), and calculate the query-key similarity (Layer 10). The compatibility for the cross-shape attention is computed as the softmax transformation of these two similarity measures (Layer 11). The cross-shape point representations of the query shape, propagating point features from the key shape, are extracted by our CSA module in Layer 12. The self-shape (Layer 4) and cross-shape (Layer 12) point representations are combined together, weighted by the pairwise compatibility, resulting in the cross-shape attention representations $\boldsymbol{X}'_m$ (Layer 13). Finally, part label probabilities are extracted per point, through a $1 \times 1 \times 1$ convolution and a softmax transformation (Layer 14), based on the concatenation of the query shape's backbone representations $\boldsymbol{X}_m$ and cross-shape attention representations $\boldsymbol{X}'_m$.

The architecture of our backbone network, "Mink-HRNet", is described in Table 2. Based on an input shape, our backbone first extracts point representations through two consecutive convolutions (Layers 2-5). Then, three multi-resolution branches are deployed. The first branch, called *High-ResNetBlock* (Layers 6, 8 and 15), operates on the input shape's resolution, while the other two, *Mid-ResNetBlock* (Layers 9 and 16) and *Low-ResNetBlock* (Layer 17), downsample the shape's resolution by a factor of 2 and 4, respectively. In addition, feature representations are exchanged between these branches, through downsampling and upsampling modules (Layers 7, 10-14). The point representations of the two low-resolution branches are upsampled to the original resolution (Layers 18-20) and by concatenating them with point features of the high-resolution branch, point representations are extracted for the input shape, through a full-connected layer (Layers 21 and 22).

The *Cross-Shape Attention* (CSA), *Downsampling* and *Upsampling* layers, along with *Residual Basic Block* are described in more detail in Table 3.

	Cross-shape network architecture	$\leftarrow$ CSN$\big($*query* $\mathcal{S}_m$, *key* $\mathcal{S}_n$, *#classes* $K\big)$
Index	Layer	Out
1	*Input*: $\mathcal{S}_m, \mathcal{S}_n$	$P_m \times 3, P_n \times 3$
2	*Mink-HRNet*$\big(\mathcal{S}_m$, 3, 256$\big)$	$P_m \times 256$ - query point repr.
3	*Mink-HRNet*$\big(\mathcal{S}_n$, 3, 256$\big)$	$P_n \times 256$ - key point repr.
4	*CSA*$\big($Out(2), Out(2), 256, 4$\big)$	$P_m \times 256$ - query SSA repr.
5	*CSA*$\big($Out(3), Out(3), 256, 4$\big)$	$P_n \times 256$ - key SSA repr.
6	*Linear-Q*$\big($*avg-pool*$\big($Out(4)$\big)$, 256, 256$\big)$	1×256 - query global repr.
7	*Linear-K*$\big($*avg-pool*$\big($Out(4)$\big)$, 256, 256$\big)$	1×256 - query global repr.
8	*Linear-K*$\big($*avg-pool*$\big($Out(5)$\big)$, 256, 256$\big)$	1×256 - key global repr.
9	*ScaledDotProduct*$\big($Out(6), Out(7)$\big)$	1×1 - query-query similarity
10	*ScaledDotProduct*$\big($Out(6), Out(8)$\big)$	1×1 - query-key similarity
11	*Softmax*$\big($Out(9), Out(10)$\big)$	2×1 - compatibility
12	*CSA*$\big($Out(2), Out(3), 256, 4$\big)$	$P_m \times 256$ - query CSA repr.
13	Out(4) $*$ *compatbility*[0] + Out(12) $*$ *compatbility*[1]	$P_m \times 256$ - cross-shape attention
14	*Softmax*$\big($*Conv*$\big($*ConCat*$\big($Out(2), Out(13)$\big)$, 512, $K\big)\big)$	$P_m \times K$ - per-point part label probabilities

Table 1: *Cross-shape network architecture for* $K = 1$ *key shapes per query shape.*

	Mink-HRNet backbone	$\leftarrow$ Mink-HRNet$\big(\textit{shape repr. } \boldsymbol{X}_m, \textit{in_feat } D_{in}, \textit{out_feat } D_{out}\big)$
Index	Layer	Out
1	$\mathit{Input}\colon \boldsymbol{X}_m$	$P_m \times D_{in}$
2	$\mathit{Conv}\big(\boldsymbol{X}_m, D_{in}, 32\big)$	$P_m \times 32$
3	$\mathit{ReLU}\big(\mathit{BatchNorm}\big(\text{Out}(2)\big)\big)$	$P_m \times 32$
4	$\mathit{Conv}\big(\text{Out}(3), 32, 64\big)$	$P_m \times 64$
5	$\mathit{ReLU}\big(\mathit{BatchNorm}\big(\text{Out}(4)\big)\big)$	$P_m \times 64$
6	$\mathit{High\text{-}ResNetBlock}\big(3\times\ \mathit{BasicBlock}\big(\text{Out}(5), 64\big)\big)$	$P_m \times 64$
7	$\mathit{Downsampling}\big(\text{Out}(6), 64, 128\big)$	$P_m/2 \times 128$
8	$\mathit{High\text{-}ResNetBlock}\big(3\times\ \mathit{BasicBlock}\big(\text{Out}(6), 64\big)\big)$	$P_m \times 64$
9	$\mathit{Mid\text{-}ResNetBlock}\big(3\times\ \mathit{BasicBlock}\big(\mathit{ReLU}\big(\text{Out}(7)\big), 128\big)\big)$	$P_m/2 \times 128$
10	$\mathit{Downsampling}\big(\text{Out}(8), 64, 128\big)$	$P_m/2 \times 128$
11	$\mathit{ReLU}\big(\mathit{Downsampling}\big(\text{Out}(8), 64, 128\big)\big)$	$P_m/2 \times 128$
12	$\mathit{Downsampling}\big(\text{Out}(11), 128, 256\big)$	$P_m/4 \times 256$
13	$\mathit{Upsampling}\big(\text{Out}(9), 128, 64\big)$	$P_m \times 64$
14	$\mathit{Downsampling}\big(\text{Out}(9), 128, 256\big)\big)$	$P_m/4 \times 256$
15	$\mathit{High\text{-}ResNetBlock}\big(3\times\ \mathit{BasicBlock}\big(\mathit{ReLU}\big(\text{Out}(8)\text{+}\text{Out}(13)\big), 64\big)\big)$	$P_m \times 64$
16	$\mathit{Mid\text{-}ResNetBlock}\big(3\times\ \mathit{BasicBlock}\big(\mathit{ReLU}\big(\text{Out}(9) + \text{Out}(10)\big), 128\big)\big)$	$P_m/2 \times 128$
17	$\mathit{Low\text{-}ResNetBlock}\big(3\times\ \mathit{BasicBlock}\big(\mathit{ReLU}\big(\text{Out}(12) + \text{Out}(14)\big), 256\big)\big)$	$P_m/4 \times 256$
18	$\mathit{ReLU}\big(\mathit{Upsampling}\big(\text{Out}(16), 128, 128\big)\big)$	$P_m \times 128$
19	$\mathit{ReLU}\big(\mathit{Upsampling}\big(\text{Out}(17), 256, 256\big)\big)$	$P_m/2 \times 256$
20	$\mathit{ReLU}\big(\mathit{Upsampling}\big(\text{Out}(19), 256, 256\big)\big)$	$P_m \times 256$
21	$\mathit{Conv}\big(\mathit{ConCat}\big(\text{Out}(3), \text{Out}(15), \text{Out}(18), \text{Out}(20)\big), 480, D_{out}\big)$	$P_m \times D_{out}$
22	$\mathit{ReLU}\big(\mathit{BatchNorm}\big(\text{Out}(21)\big)\big)$	$P_m \times D_{out}$

Table 2: *Mink-HRNet backbone architecture. High, mid and low-resolution ResNet blocks consist of 3 consecutive residual basic blocks, each. Point representations are exchanged between multi-resolution branches via downsampling and upsampling layers (see Table 3 for a more detailed description of their architecture). The convolution kernel of Layer 2 is of size* $5 \times 5 \times 5$*, in order to increase its receptive field, while for Layer 4 is of size* $3 \times 3 \times 3$*. For Layer 21 we used a kernel of* $1 \times 1 \times 1$*, since this acts as a fully-connected layer.*

MID-FC-CSN architecture details. Similar to the MinkHRNetCSN, the MID-FC-CSN variant also follows a comparable architecture (see Table 4). To extract point features $\boldsymbol{X}_m$ and $\boldsymbol{X}_n$ for the input query-key pair of shapes, the “MID-Net” backbone is utilized (Layers 2 and 3). This backbone also adopts a three-stage HRNet architecture, which is built on an octree-based CNN framework [WLG*17]. ResNet blocks with a bottleneck structure [HZRS16] are used in all multi-resolution branches, and feature sharing is achieved using downsample and upsample exchange blocks, implemented by max-pooling and tri-linear up-sampling, respectively. The CSA module (Layers 4 and 12) is employed to construct the self-shape and cross-shape attention features for the query shape. These are then weighted by the learned pairwise compatibility (Layers 4-11) and aggregated to generate the final cross-shape attention representations $\boldsymbol{X}'_m$ (Layer 13). Part label probabilities are extracted per point using a fully-connected layer and a softmax transformation based on the cross-shape attention representations (Layer 14).

Appendix B: Key shape retrieval measure comparison

As an additional ablation, we evaluated the performance of our “MinkHRNetCSN-K1” variant for two key shape retrieval measures (see Section 3.2 in the main text). The first relies on the point-wise representations between a query and a key shape and retrieves key shapes that are on average more similar to their query counterparts (Eq. 14, main text). The second measure, takes into account only the global representations of a query-key pair of shapes (Eq. 10, main text). In Table 5 we report the performance for both measures, in terms of Part IoU and Shape IoU. Our default variant, “MinkHRNetCSN-K1 (Eq. 14)”, achieves better performance according to Part IoU (+2.5%), and it outperforms the other variant (“MinkHRNetCSN-K1, Eq. 10)” in 16 out 17 object categories. This is a strong indication that the key shape retrieval measure based in Eq. 14 is more effective in retrieving key shapes for cross-shape attention.

References

[HZRS16] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. “Deep Residual Learning for Image Recognition”. *Proc. CVPR*. 2016 2.

[WLG*17] Wang, Peng-Shuai, Liu, Yang, Guo, Yu-Xiao, et al. “O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis”. *TOG* 36.4 (2017) 2.

[WSC*21] Wang, Jingdong, Sun, Ke, Cheng, Tianheng, et al. “Deep High-Resolution Representation Learning for Visual Recognition”. *TPAMI* 43.10 (2021) 1.

	Cross-Shape Attention Layer	$\leftarrow$ CSA$\big(query\ \boldsymbol{X}_m,\ key\ \boldsymbol{X}_n,\ \#feats\ D,\ \#heads\ H\big)$
Index	Layer	Out
1	$Input$: $\boldsymbol{X}_m$,$\boldsymbol{X}_n$	$P_m \times D, P_n \times D$
2	$H\times$ $Linear\text{-}Q\big(\boldsymbol{X}_m, D, \lfloor D/H \rfloor\big)$	$P_m \times H \times D'$
3	$H\times$ $Linear\text{-}K\big(\boldsymbol{X}_n, D, \lfloor D/H \rfloor\big)$	$P_n \times H \times D'$
4	$H\times$ $Linear\text{-}V\big(\boldsymbol{X}_n, D, \lfloor D/H \rfloor\big)$	$P_n \times H \times D'$
5	$Attention\big($Out(2), Out(3)$\big)$	$H \times P_m \times P_n$
6	$MatMul\big($Out(5), Out(4)$\big)$	$P_m \times H \times D'$
7	$Linear\big(ConCat\big($Out(6)$\big)$, D, $D\big)$	$P_m \times D$
8	$LayerNorm\big(\boldsymbol{X}_m+$ Out(7)$\big)$	$P_m \times D$

	Downsampling Layer	$\leftarrow$ Downsampling$\big(shape\ repr.\ \boldsymbol{X}_m,\ in_feat\ D_{in},\ out_feat\ D_{out}\big)$
1	$Input$: $\boldsymbol{X}_m$	$P_m \times D_{in}$
2	$Conv\big(\boldsymbol{X}_m, D_{in}, D_{out}, stride = 2\big)$	$P_m/2 \times D_{out}$
3	$BatchNorm\big($Out(2)$\big)$	$P_m/2 \times D_{out}$

	Upsampling Layer	$\leftarrow$ Upsampling$\big(shape\ repr.\ \boldsymbol{X}_m,\ in_feat\ D_{in},\ out_feat\ D_{out}\big)$
1	$Input$: $\boldsymbol{X}_m$	$P_m \times D_{in}$
2	$TrConv\big(\boldsymbol{X}_m, D_{in}, D_{out}, stride = 2\big)$	$2 * P_m \times D_{out}$
3	$BatchNorm\big($Out(2)$\big)$	$2 * P_m \times D_{out}$

	Residual Basic Block	$\leftarrow$ BasicBlock$\big(shape\ repr.\ \boldsymbol{X}_m,\ \#feats\ D\big)$
1	$Input$: $\boldsymbol{X}_m$	$P_m \times D$
2	$Conv\big(\boldsymbol{X}_m, D, D\big)$	$P_m \times D$
3	$ReLU\big(BatchNorm\big($Out(2)$\big)\big)$	$P_m \times D$
4	$Conv\big($Out(3), D, $D\big)$	$P_m \times D$
5	$ReLU\big(\boldsymbol{X}_m + BatchNorm\big($Out(4)$\big)\big)$	$P_m \times D$

Table 3: *Cross-Shape Network basic layers. All convolution kernels are of size* $3 \times 3 \times 3$.

	MID-FC-Cross-shape network architecture	$\leftarrow$ MID-FC-CSN$\big(query\ \mathcal{S}_m,\ key\ \mathcal{S}_n,\ \#classes\ K\big)$
Index	Layer	Out
1	$Input$: $\mathcal{S}_m$,$\mathcal{S}_n$	$P_m \times 3, P_n \times 3$
2	$MID\text{-}Net\big(\mathcal{S}_m$, 3, 256$\big)$	$P_m \times 256$ - query point repr.
3	$MID\text{-}Net\big(\mathcal{S}_n$, 3, 256$\big)$	$P_n \times 256$ - key point repr.
4	$CSA\big($Out(2), Out(2), 256, 8$\big)$	$P_m \times 256$ - query SSA repr.
5	$CSA\big($Out(3), Out(3), 256, 8$\big)$	$P_n \times 256$ - key SSA repr.
6	$Linear\text{-}Q\big(avg\text{-}pool\big($Out(4)$\big)$, 256,256$\big)$	1×256 - query global repr.
7	$Linear\text{-}K\big(avg\text{-}pool\big($Out(4)$\big)$, 256,256$\big)$	1×256 - query global repr.
8	$Linear\text{-}K\big(avg\text{-}pool\big($Out(5)$\big)$, 256,256$\big)$	1×256 - key global repr.
9	$ScaledDotProduct\big($Out(6), Out(7)$\big)$	1×1 - query-query similarity
10	$ScaledDotProduct\big($Out(6), Out(8)$\big)$	1×1 - query-key similarity
11	$Softmax\big($Out(9), Out(10)$\big)$	2×1 - compatibility
12	$CSA\big($Out(2), Out(3), 256, 8$\big)$	$P_m \times 256$ - query CSA repr.
13	Out(4) $*$ $compatbility$[0] + Out(12) $*$ $compatbility$[1]	$P_m \times 256$ - cross-shape attention
14	$Softmax\big(FC\big($Out(13), 256, $K\big)\big)$	$P_m \times K$ - per-point part label probabilities

Table 4: *MID-FC-CSN architecture for* $K = 1$ *key shapes per query shape.*

Category	Bed	Bott	Chai	Cloc	Dish	Disp	Door	Ear	Fauc	Knif	Lamp	Micr	Frid	Stor	Tabl	Tras	Vase	**avg.**	**#cat.**
Part IoU																			
MinkHRNetCSN-K1 (Eq. 14)	**42.1**	**54.0**	**42.5**	**42.9**	**58.2**	**83.2**	**43.5**	**51.5**	**59.4**	**47.8**	**27.9**	**57.4**	43.7	**46.2**	**36.8**	**51.5**	**60.0**	**49.9**	**16**
MinkHRNetCSN-K1 (Eq. 10)	38.7	47.0	41.9	40.8	55.7	82.3	41.3	50.5	57.9	37.3	24.7	56.2	**44.1**	45.9	32.3	51.4	58.8	47.4	1

Table 5: *Comparison of shape retrieval measures based on point-wise (Eq. 14, main text) and global (Eq. 10, main text) representations of a query-key pair of shapes, in terms of Part IoU and Shape IoU.*