

Supplementary Materials

Anonymous Authors

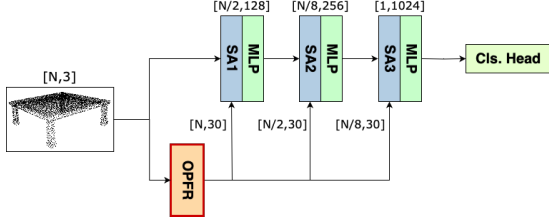


Figure 1: Detailed architecture of PointNet++ & OPFR.

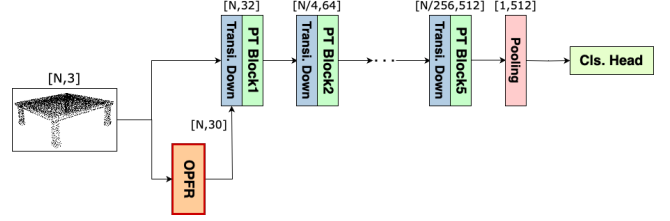


Figure 2: Detailed architecture of Point Transformer & OPFR.

1 OVERVIEW

Due to the space limit of main paper, the supplementary materials provide more details of experiments and contain visualizations, ablation study, and theoretical analysis to further understand our proposed *On-the-fly Point Feature Representation (OPFR)*.

In Sec. 2, we provide more implementation details, specific network architectures to integrate OPFR with various 3D backbones [2, 8], and explanation of efficiency metrics. In Sec. 3, we present more interesting visualization results, including the triangle sets produced by *Hierarchical Sampling* module, OPFR values on ScanObjectNN classification task, and qualitative results on S3DIS semantic segmentation task. In Sec. 4, we conduct additional ablation study for *Hierarchical Sampling* module when applied to RepSurf [4]. In Sec. 5, we provide the theoretical analysis of the proposed curvature proxy p_{ij} as mentioned in the main paper.

2 DETAILS IN EXPERIMENTS

Implementation details. For classification tasks, we apply point clouds augmentation (random scale, random shift, and random dropout) and point clouds normalization (normalize to $[-1, 1]$) when training on ModelNet40 [7], while we do not apply any augmentation techniques when training on ScanObjectNN [6]. Before feeding into the backbone, the input point clouds are downsampled to 1024 points with farthest point sampling algorithm. For semantic segmentation tasks, we conduct point clouds augmentation (point cloud scaling, color contrasting, color shifting, and color jittering) on S3DIS [1] dataset.

Architecture details of PointNet++ & OPFR. For classification, shown in Fig. 1, we add our OPFR representation $\mathbf{r}_i \in \mathbb{R}^{30}$ before each set abstraction (SA) layer for PointNet++ [2] backbone. The OPFR representation \mathbf{r}_i is concatenated with original input \mathbf{x}_i^k , followed by shared-MLP. Here, \mathbf{x}_i^k represents the i -th input before the k -th SA layer ($k = 1, 2, 3$). The dimensions of shared-MLP for SA1, SA2, SA3 are $[64, 64, 128]$, $[128, 128, 256]$, $[256, 512, 1024]$ respectively. Eventually, the resultant 1024-dim point cloud representation is fed into a classification head. The dimensions of classification head are $[1024, 512, 256, K]$, and K is the number of categories for our outputs. For semantic segmentation task, we simply replace the classification head with standard PointNet++ decoder and the segmentation head.

Architecture details of Point Transformer & OPFR. For classification, shown in Fig. 2, we add our OPFR representation $\mathbf{r}_i \in \mathbb{R}^{30}$ before 1-st Transition Down (Transi. Down) layer for Point Transformer [8] backbone. The OPFR representation \mathbf{r}_i is concatenated with original input point cloud \mathbf{x}_i^1 , followed by Point Transformer blocks. The dimensions of Point Transformer blocks are $[32, 64, 128, 256, 512]$ respectively. Eventually, the resultant 512-dim point cloud representation is fed into a classification head. The dimensions of classification head are $[512, 512, K]$, and K is the number of categories for our outputs. For semantic segmentation task, we simply replace the classification head with standard Point Transformer decoder and the segmentation head.

Explanation of efficiency metrics. We adopt number of total parameters (#Params) and floating point operations (FLOPs) to quantify the efficiency of the proposed OPFR. FLOPs count the number of floating point operations (addition and multiplication) required for a given input. We adopt FLOPs as our efficiency metric since it is hardware-agnostic, ensuring a fair comparison across different models. Furthermore, we fix the size of input point clouds for different models. Following PointNeXt [3], we employ 1024 input point clouds for classification, and 15000 input point clouds for semantic segmentation to calculate FLOPs fairly.

3 VISUALIZATION

Triangle sets of Hierarchical Sampling. In Fig. 3, we zoom in to visualize the triangle sets generated by *Hierarchical Sampling* and naive k nearest neighbors (k -NN) sampling. Column (a) depicts input point clouds from ModelNet40 [7], and the corresponding zoom-in regions using black dashed circles, column (b) and (c) depict triangle sets produced by k -NN sampling ($k = 8$) from isometric view and front view, and column (d) and (e) depict triangle sets generated by *Hierarchical Sampling* ($k_1 = 20, k_2 = 2, k_3 = 8$) from isometric view and front view. The **first two rows** demonstrate the ability of *Hierarchical Sampling* module to decouple paralleled structures. Specifically, in 2-nd row, we visualize the triangle set around an airplane wing using different sampling methods. With naive k -NN sampling, those points from upper and lower surfaces of the wing are mixed together, which leads to a noticeably messy distortion of the obtained triangle set. These points are the “noisy points”, as we mentioned in the introduction of main paper. Our

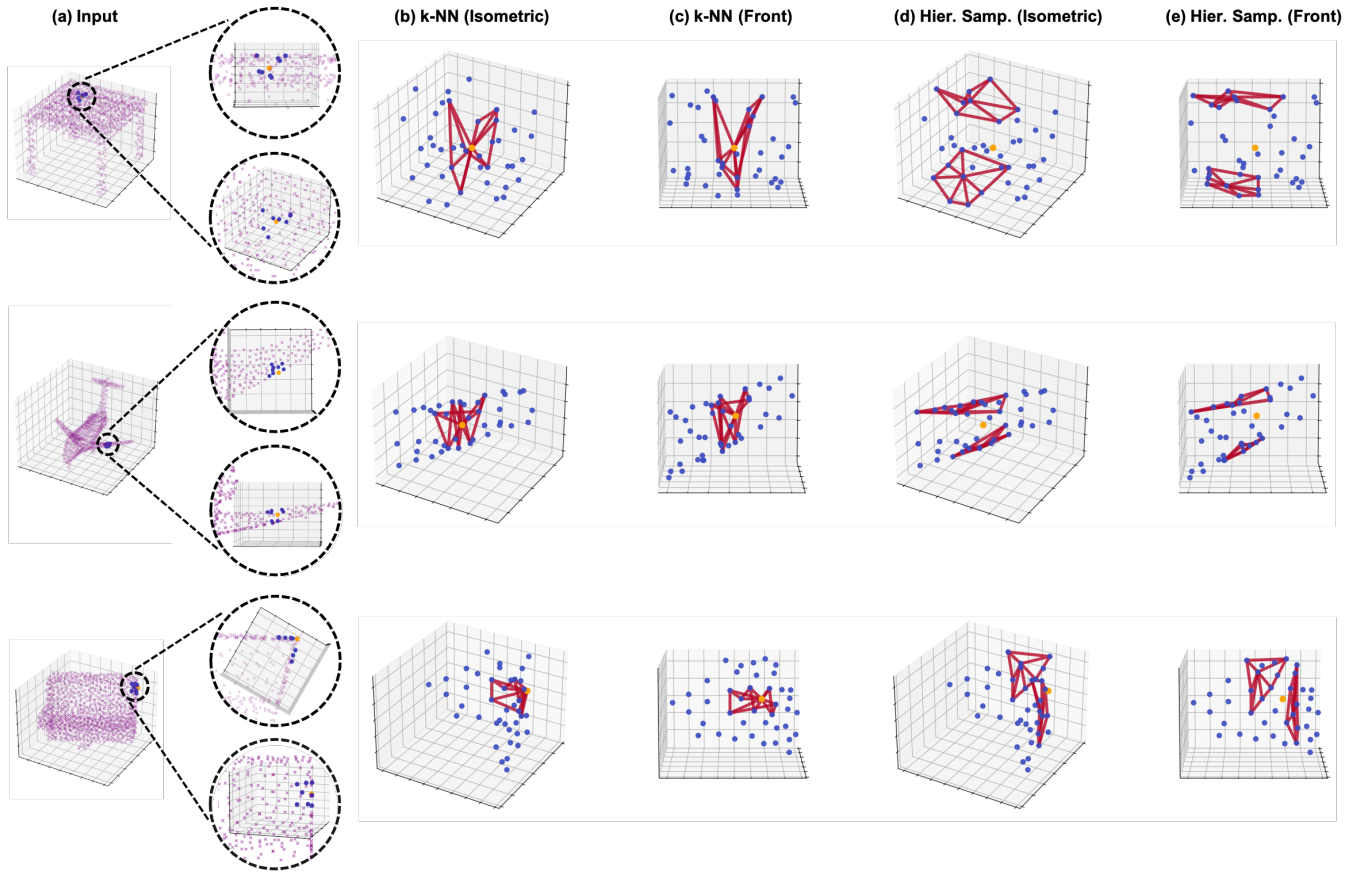


Figure 3: Visualization of triangle sets generated by naive k -NN sampling and *Hierarchical Sampling* from isometric view and front view. The three rows correspond to table, airplane, and piano. We zoom in to highlight the triangle sets in column (b), (c), (d), and (e), and the zoom-in regions are depicted using black dashed circles upon raw inputs from two different views. Orange represents our interested point, blue represents nearby neighbors, and red represents the obtained triangle set. The triangle sets generated by k -NN are messy, while those produced by *Hierarchical Sampling* are well-organized.

proposed method addresses this distortion issue by employing farthest point sampling to decouple the wing structure into two distinct parallel surfaces. In column (e), the wing structure is clearly discernible through the *Hierarchical Sampling*. Furthermore, the **last row** demonstrates the ability of our proposed method to identify the edged structure. Specifically, in 3-rd row, we visualize the triangle set around the edge of a piano using different sampling methods. With naive k -NN sampling, the corresponding triangle set contains points from side surface and back surface, resulting in the distortion of generated triangle set. These points are the “noisy points”. In contrast, as observed from column (e), through *Hierarchical Sampling*, it effectively approximates the edged structure by two disjoint surfaces which are located in the side surface and back surface. From these visualization examples, it clearly demonstrates the superiority of our presented *Hierarchical Sampling* strategy to mitigate the distortion issue that occurs in naive k -NN sampling, ensuring the robustness of obtained geometric features.

OPFR values on ScanObjectNN classification. In Fig. 4, we provide additional visualization examples of OPFR values on ScanObjectNN [6] benchmark from isometric view and corresponding three-view drawing (side view, top view, and front view). We show the OPFR values of 1-st channel for four representative objects: sofa, sink, bin, and toilet. Note that, to maintain the consistency with classification experiments in the main paper, we use the hardest variant (PB_T50_RS_variant) as well for visualization. Different from ModelNet40 [7], this hardest variant applies data augmentation for original point clouds, which injects noise to the visualization results. This poses additional challenges for OPFR. Shown in Fig. 4, the red hues represent smaller OPFR values, typically those curved regions (e.g., backrest of the sofa and basin of the sink), while the blue hues indicate larger OPFR values, particularly those flat regions (e.g., seat cushion of the sofa and vanity top of the sink). This color differentiation emphasizes that, the proposed OPFR representation is eligible to perceive the local geometric information numerically, even with injected noise. It clearly demonstrates the curvature-aware property of the proposed OPFR.

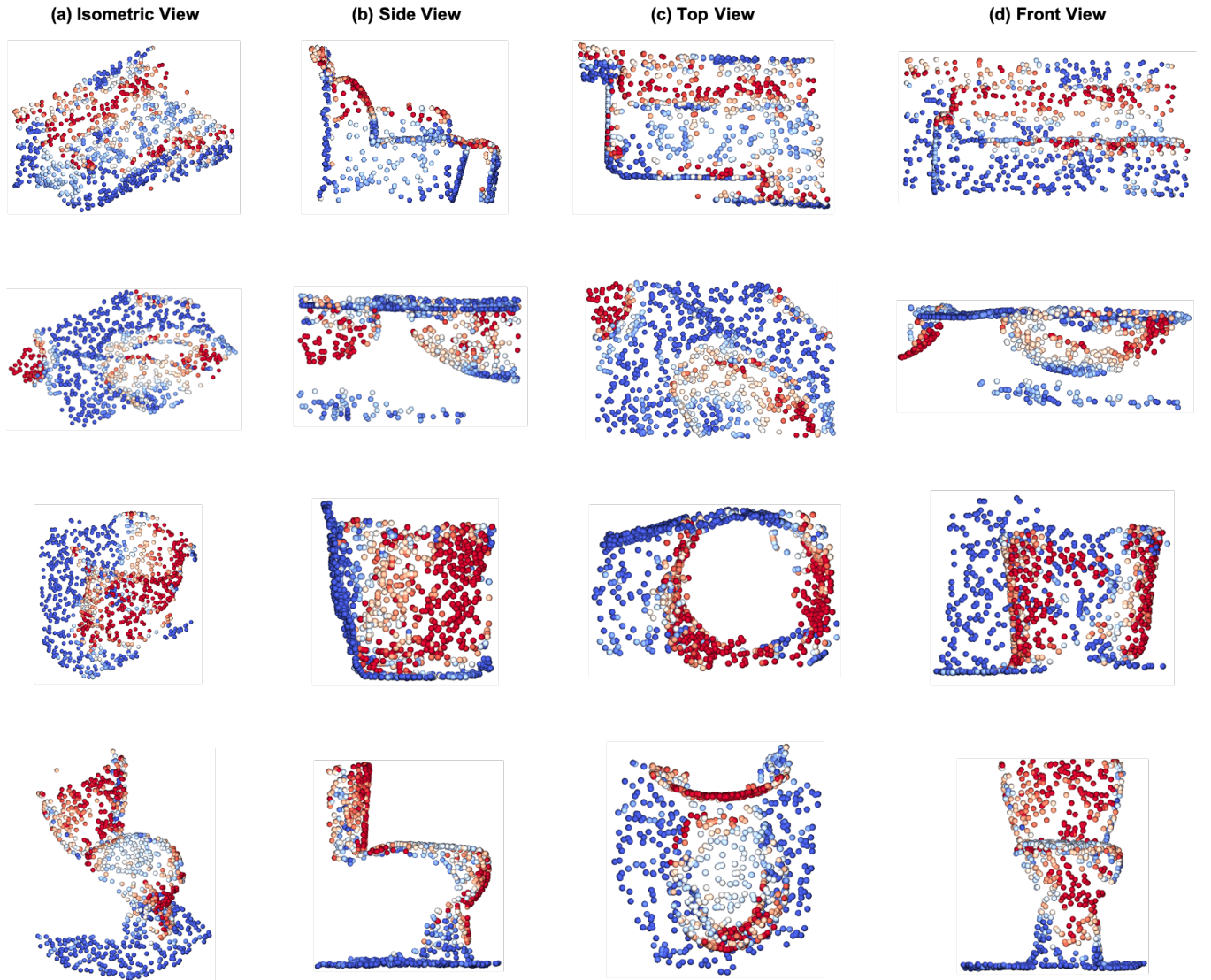


Figure 4: Visualization of OPFR values (1-st channel) on ScanObjectNN benchmark from isometric view and three-view drawing (side view, top view, and front view). The four rows correspond to sofa, sink, bin, and toilet, respectively. Red represents small OPFR values, typically those curved regions, and blue indicates large OPFR values, particularly those flat regions.

Qualitative results on S3DIS semantic segmentation. In Fig. 5, we provide qualitative results when integrating the OPFR with Point Transformer [8] backbone on S3DIS [1] benchmark for semantic segmentation. We visualize the segmentation results on various scenes, including two conference rooms (first two rows), two offices (middle two rows), and two hallways (last two rows). When equipped with OPFR, Point Transformer demonstrates its superior capability to generate predictions that are closer to the ground truth on all scenes. Shown in Fig. 5, it can better segment difficult classes, including columns (1-st, 2-nd, 3-rd, and 4-th rows), clutter (2-nd, 5-th, and 6-th rows), bookcase (3-rd, 4-th, and 5-th rows). Furthermore, our Point Transformer & OPFR can capture more precise segmentation boundaries between chair and table

(1-st, 2-nd and 3-rd rows), and distinguish between two analogous classes: chair and sofa (5-th row). Zoom in for more details.

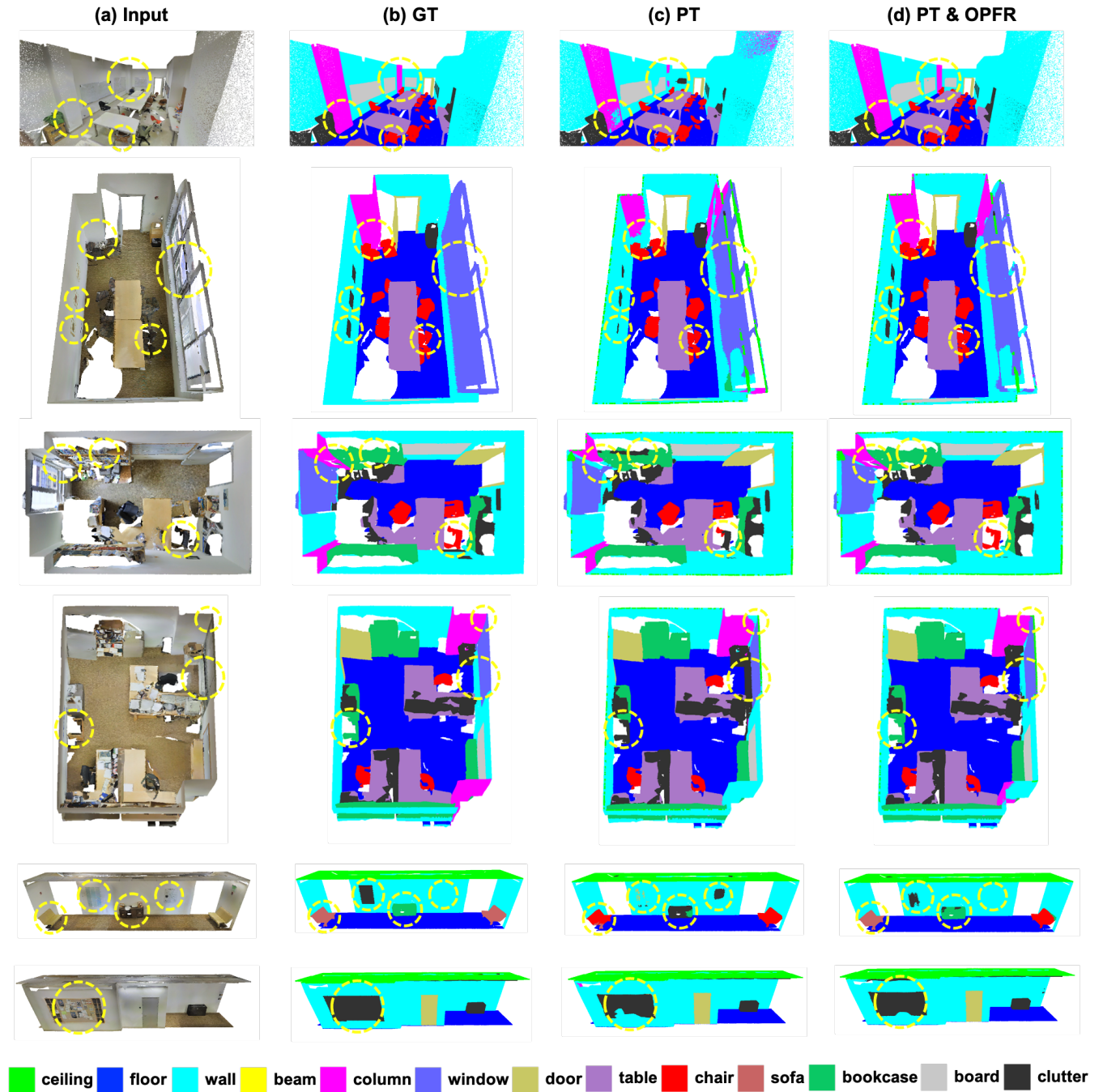


Figure 5: Qualitative comparisons of Ground Truth (GT), Point Transformer (PT), and Point Transformer & OPFR (PT & OPFR) on S3DIS semantic segmentation. We visualize the input point clouds using original RGB in column (a), and using color mappings of ground truth or predicted semantic class in column (b), (c), and (d). Differences between PT predictions and PT & OPFR predictions are highlighted with yellow dashed circles. Zoom in for more details.

Table 1: Ablation study on the *Hierarchical Sampling* module to combine with prior state-of-the-art representation, RepSurf. We conduct experiments on ScanObjectNN dataset.

Method	OA	mAcc
RepSurf [4]	84.22	81.79
(+) <i>Hierarchical Sampling</i> strategy	+0.62	+0.57
PointNet++ & OPFR[†] (ours)	84.51	82.90
(+) <i>Hierarchical Sampling</i> strategy	+1.17	+0.91

[†]: OPFR w/o *Hierarchical Sampling* module.

4 ABLATION STUDY

Compatibility of *Hierarchical Sampling* module. As mentioned in the ablation study of the effectiveness of different modules in the main paper, the *Hierarchical Sampling* technique can substitute k -NN sampling in other methods as well to mitigate the distortion issue of obtained triangle sets. In Tab. 1, we apply *Hierarchical Sampling* strategy to the previous state-of-the-art representation, RepSurf [4], and achieve an improvement of 0.62% overall accuracy (OA) and 0.57% mean accuracy (mAcc). Furthermore, *Hierarchical Sampling* strategy boosts the OPFR by 1.17% OA and 0.91% mAcc when integrated with PointNet++ [2] backbone. The empirical results demonstrate that, the *Hierarchical Sampling* module is compatible with different methods, and alleviates the distortion issue of k -NN triangle sets. Additionally, when integrated with the *Hierarchical Sampling* strategy, our proposed OPFR achieves more improvements (+0.55% OA and +0.34% mAcc) compared to RepSurf. We hypothesize that, this phenomenon is attributed to, curvature information is more sensitive to the quality of inherent triangle sets. Therefore, when equipped with *Hierarchical Sampling* strategy, we can guarantee the reliability of obtained geometric features.

5 THEORETICAL ANALYSIS

We provide the theoretical analysis of our proposed curvature proxy \mathbf{p}_{ij} . As claimed in the main paper, curvature proxy \mathbf{p}_{ij} can be viewed as one good approximation of curvature definition [5] from differential geometry in the direction of three reference frames $\{\hat{\mathbf{u}}_{ij}, \hat{\mathbf{v}}_{ij}, \hat{\mathbf{w}}_{ij}\}$. Without loss of generality, we give the detailed analysis for the reference frame $\hat{\mathbf{u}}_{ij}$. The analyses for another two reference frames $\hat{\mathbf{v}}_{ij}$ and $\hat{\mathbf{w}}_{ij}$ are similar.

Theoretical Analysis. In Fig. 6, we depict the schematic and necessary notation for the analysis. We follow the notation in the main paper: $\{(x_i, x_{ij})\}$ is the point pair, $\hat{\mathbf{u}}_{ij}$ is one of the approximated local reference frames, and \mathbf{x}'_{ij} is the relative position between the point pair. Additionally, l_{ij} represents the arc length between the point pair, and θ_{ij} represents the angle between relative position \mathbf{x}'_{ij} and reference frame $\hat{\mathbf{u}}_{ij}$. Given one continuous 3D surface whose cross-section is shown in Fig. 6, its curvature $\kappa_{\hat{\mathbf{u}}_{ij}}(x_i)$ at point x_i with respect to direction $\hat{\mathbf{u}}_{ij}$ can be defined as [5]:

$$\kappa_{\hat{\mathbf{u}}_{ij}}(x_i) := \lim_{\Delta l \rightarrow 0} \frac{\Delta \theta}{\Delta l}, \quad (1)$$

where $\Delta \theta$ and Δl are infinitesimal angle and arc length along the direction $\hat{\mathbf{u}}_{ij}$ around point x_i . In the following discussion, we omit x_i in curvature notation for simplicity. However, in practice, we lack

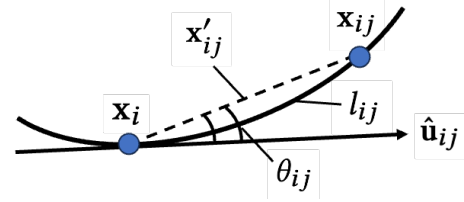


Figure 6: Schematic and necessary notation for theoretical analysis. This curve is obtained by the intersection of, underlying continuous 3D surface for point clouds and the plane spanned by vector \mathbf{x}'_{ij} and $\hat{\mathbf{u}}_{ij}$.

access to the underlying continuous 3D surface. This absence makes it impossible to determine the exact curvature $\kappa_{\hat{\mathbf{u}}_{ij}}$. In real-life scenarios, we work with point clouds, which are discrete samplings of continuous 3D surface. Then, a practical approach is to approximate the curvature $\kappa_{\hat{\mathbf{u}}_{ij}}$ based on nearest neighbor x_{ij} . As a result, the infinitesimal angle $\Delta \theta$ can be approximated by the angle θ_{ij} from nearest neighbor x_{ij} :

$$\Delta \theta \approx \theta_{ij} = \arccos(\hat{\mathbf{u}}_{ij} \odot \frac{\mathbf{x}'_{ij}}{\|\mathbf{x}'_{ij}\|}). \quad (2)$$

Similarly, the infinitesimal arc length Δl can be approximated by the arc length l_{ij} from the nearest neighbor x_{ij} :

$$\Delta l \approx l_{ij} \approx \|\mathbf{x}'_{ij}\|. \quad (3)$$

Based on Equ. 1, Equ. 2 and Equ. 3, we can summarize that,

$$\kappa_{\hat{\mathbf{u}}_{ij}} \approx \frac{1}{\|\mathbf{x}'_{ij}\|} \cdot \arccos(\hat{\mathbf{u}}_{ij} \odot \frac{\mathbf{x}'_{ij}}{\|\mathbf{x}'_{ij}\|}).$$

□

Therefore, we have achieved our claim in the main paper that, the proposed curvature proxy \mathbf{p}_{ij} is one approximation of curvature definition from differential geometry. This can be formally formulated as:

$$\mathbf{p}_{ij} \approx [\kappa_{\hat{\mathbf{u}}_{ij}}, \kappa_{\hat{\mathbf{v}}_{ij}}, \kappa_{\hat{\mathbf{w}}_{ij}}].$$

Lastly, we want to mention that, the accuracy of this approximation can be ensured via the usage of k nearest neighbors (k -NNs) sampling method, which guarantees $\Delta l \approx \|\mathbf{x}'_{ij}\| \approx 0$.

REFERENCES

- [1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1534–1543.
- [2] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).
- [3] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems* 35 (2022), 23192–23204.
- [4] Haoxi Ran, Jun Liu, and Chengjie Wang. 2022. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18942–18952.
- [5] Isabel M Serrano and Bogdan D Suceava. 2015. A medieval mystery: Nicole Oresme's concept of curvitas. *Notices of the AMS* 62, 9 (2015).

- [6] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. 2019. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- [7] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.
- [8] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 16259–16268.

639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696