Starting from Non-Parametric Networks for 3D Point Cloud Analysis
Supplementary Material

Renrui Zhang[1,5], Liuhui Wang[2,6], Yali Wang[4,5], Peng Gao[5], Hongsheng Li[1], Jianbo Shi[†3]

[1]CUHK MMLab [2]Peking University [3]University of Pennsylvania
[4]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[5]Shanghai Artificial Intelligence Laboratory [6]Heisenberg Robotics

{zhangrenrui, gaopeng}@pjlab.org.cn, jshi@seas.upenn.edu
wangliuhui0401@pku.edu.cn, hsli@ee.cuhk.edu.hk

1. Discussion

1.1. Why Do Trigonometric Functions Work?

We leverage the trigonometric function to conduct non-parametric raw-point embedding and geometry extraction. It can reveal the 3D spatial patterns benefited from the following three properties.

Capturing High-frequency 3D Structures. As discussed in Tancik et al. [45], transforming low-dimensional input by sinusoidal mapping helps MLPs to learn the high-frequency content during training. Similarly to our non-parametric encoding, Point-NN utilizes trigonometric functions to capture the high-frequency spatial structures of 3D point clouds, and then recognize them from these distinctive characteristics by the point-memory bank. In Figure 1, we visualize the low-frequency (Top) and high-frequency (Middle) geometry of the input point cloud, and compare them with the feature responses of Point-NN (Bottom). The high-frequency geometries denotes the spatial regions of edges, corners, and other fine-grained details, where the local 3D coordinates vary dramatically, while the low-frequency structure normally includes some flat and smooth object surfaces with gentle variations. As shown, aided by trigonometric functions, our Point-NN can concentrate well on these high-frequency 3D patterns.

Encoding Absolute and Relative Positions. Benefited from the nature of sinusoid, the trigonometric functions can not only represent the absolute position in the embedding space, but also implicitly encode the relative positional information between two 3D points. For two points, $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$, we first obtain their C-dimensional embeddings referring to Equation (5∼7) in the main paper, formulated as

$$\mathrm{PosE}(p_i) = \mathrm{Concat}(f_i^x,\ f_i^y,\ f_i^z), \tag{1}$$
$$\mathrm{PosE}(p_j) = \mathrm{Concat}(f_j^x,\ f_j^y,\ f_j^z), \tag{2}$$

where $\mathrm{PosE}(\cdot)$ denotes the positional encoding by trigonometric functions, and $f_{i/j}^x,\ f_{i/j}^y,\ f_{i/j}^z \in \mathbb{R}^{1\times\frac{C}{3}}$ denote the embeddings of three axes. Then, their spatial relative relation can be revealed by the dot production between the two embeddings, formulated as

$$f_i^x f_j^{xT} + f_i^y f_j^{yT} + f_i^z f_j^{zT} = \mathrm{PosE}(p_i)\,\mathrm{PosE}(p_j)^T.$$

Taking the x axis as an example,

$$\sum_{m=0}^{\frac{C}{6}-1} \mathrm{cosine}\big(\alpha(x_i - x_j)/\beta^{\frac{6m}{C}}\big) = f_i^x f_j^{xT}, \tag{3}$$

which indicates the relative x-axis distance of two points, in a similar way to the other two axes. Therefore, the trigonometric function is capable of encoding both absolute and relative 3D positional information for point cloud analysis.

Local Geometry Extraction. In Equation (9) of the main paper, we weigh each neighbor feature f_j within the local region by the relative positional embedding, $\mathrm{PosE}(\Delta p_j)$, formulated as

$$f_{cj}^w = \big(f_{cj} + \mathrm{PosE}(\Delta p_j)\big) \odot \mathrm{PosE}(\Delta p_j). \tag{4}$$

The weighing is conducted sequentially by element-wise addition and multiplication. Firstly, the addition is to complement f_{cj} with higher frequency information. Due to feature expansion, the output dimensions of $\mathrm{PosE}(\Delta p_j)$ of 4 stages are respectively $2C_I$, $4C_I$, $8C_I$, and $16C_I$. As

† Corresponding author

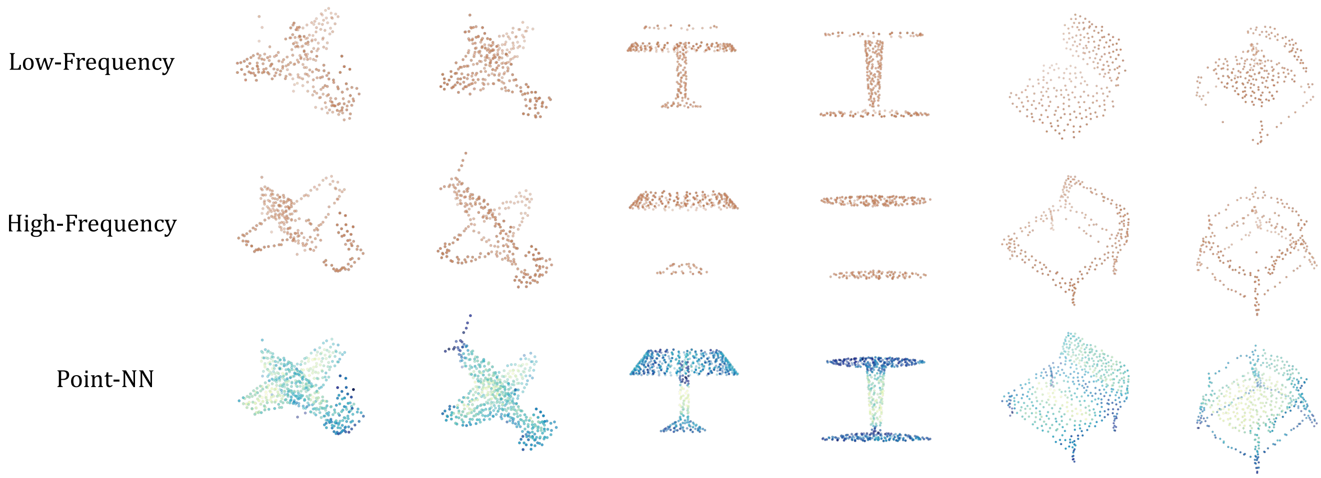

Figure 1. **Why Do Trigonometric Functions Work?** For an input point cloud, we visualize its low-frequency and high-frequency geometries referring to [57], and compare with the feature responses after the first network stage of Point-NN, where darker colors indicate higher responses. As shown, Point-NN can focus on the high-frequency 3D structures with sharp variations of the point cloud.

Benchmark	ScanObjectNN	ModelNet40	ShapeNetPart
Point-PN	87.1	93.8	86.6
+NN	+0.1	+0.2	+0.0

Table 1. **Can Point-NN Improve Point-PN by Plug-and-play?** We report the accuracy (%) on the PB-T50-RS split of ScanObjectNN [48], ModelNet40 [53], and ShapeNetPart [59].

Method	Pre-train in 2D	Pre-train in 3D	3D Data	Acc. (%)
PointCLIP [62]	✓	-	-	20.2
CALIP [16]	✓	-	-	21.5
CLIP2Point [22]	✓	✓	✓	49.4
ULIP [58]	✓	✓	✓	60.4
PointCLIP V2 [73]	✓	-	-	64.2
Point-NN	-	-	✓	**81.8**

Table 2. **Comparison of Training-free Methods in 3D.** We report their performance without training on ModelNet40 [53].

the embedding frequency depends on feature dimension referring to Equation (6) of the main paper, the embeddings at higher stages obtain higher frequencies. Taking the first stage as an example, $\mathrm{PosE}(\Delta p_j)$ is $2C_I$-dimensional, but f_{cj} is a concatenation of two C_I-dimensional vectors, which makes their embedding frequencies inconsistent. Therefore, we adopt addition to endow f_{cj} with the frequency corresponding to $2C_I$ dimension. Then, the second-step multiplication weighs the magnitude of each element in f_{cj} by its relative positional information. This determines the importance of different neighbor points in the subsequent pooling operations, and the final aggregated features of the local neighborhood. In this way, Point-NN can effectively embed local 3D patterns via $\mathrm{PosE}(\cdot)$ without any learnable operators.

1.2. Can Point-NN Improve Point-PN?

Point-NN can provide complementary geometric knowledge and serve as a plug-and-play module to boost existing learnable 3D models. Although Point-PN is also a learnable 3D network, the enhanced performance brought by Point-NN is marginal as reported in Table 1. By visualizing feature responses in Figure 2, we observe that the complementarity between Point-NN and Point-PN is much weaker than that between Point-NN and PointNet++ [36]. This is because the non-parametric framework of Point-PN is mostly inherited from Point-NN, also capturing high-frequency 3D geometries via trigonometric functions. Therefore, the learnable Point-PN extracts similar 3D patterns to Point-NN, which harms its plug-and-play capacity.

1.3. Training-free Methods in 3D

Our Point-NN conducts no training, but requires 3D training data to construct the point-memory bank. Inspired by the transfer learning in 2D and language [2, 10, 12, 27, 60, 63, 68, 72], some recent works [16, 22, 58, 62, 73] adapt the pre-trained models from other modalities, e.g., CLIP [39], to 3D domains in a zero-shot manner. Via the diverse pre-trained knowledge, they are also training-free and do not need any 3D training data. As compared in Table 2, different from other methods based on 2D or 3D pre-training, our method is a pure non-parametric network without any learnable parameters or pre-trained knowledge.

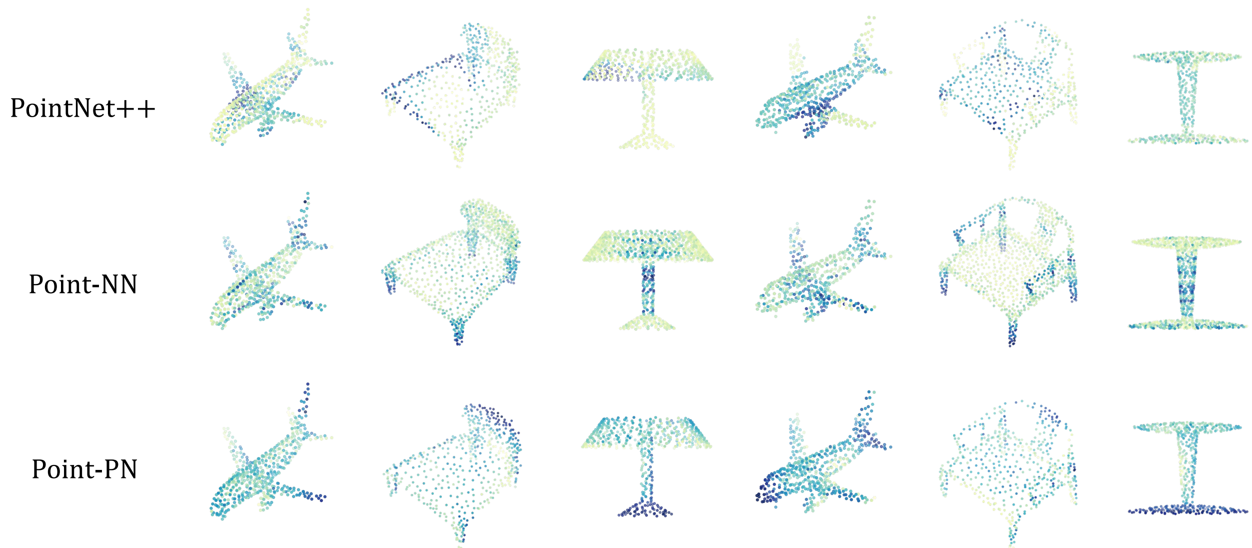

Figure 2. **Can Point-NN Improve Point-PN by Plug-and-play?** We visualize the feature responses after the first network stage for Point-NN, the trained PointNet++ [36] and Point-PN, where darker colors indicate higher responses. As shown, Point-PN captures similar 3D patterns to Point-NN, which harms their complementarity.

k	1	10	100	500	1000	5000	All
Top-k PoM	80.4	81.1	81.3	81.4	81.4	81.7	**81.8**
k-NN	80.7	79.5	67.0	45.7	36.4	8.5	-

Table 3. **Point-Memory Bank vs. k-NN.** 'Top-k PoM' denotes the point-memory bank with top-k similarities, and 'All' denotes 9,840 training samples. We utilize our non-parametric encoder to extract features and report the accuracy (%) on ModelNet40 [53].

Baseline	Method	Gain (%)	Param.	Time
PAConv	PnP-3D	+0.2 Acc.	+0.7 M	+14 h
	Point-NN	+0.2 Acc.	+0 M	+48 s
VoteNet	PnP-3D	+1.4 AP_{25}	+0.3 M	+10 h
	Point-NN	+1.2 AP_{25}	+0 M	+9.3 min

Table 4. **Point-NN vs. PnP-3D [38].** We adopt two baseline models for comparison, PAConv [56] and VoteNet [5], respectively on ModelNet40 [53] and SUN RGB-D [43] datasets.

1.4. Point-Memory Bank vs. k-NN?

Based on the already extracted point cloud features, our point-memory bank and k-NN algorithm both leverage the inter-sample feature similarity for classification without training, but are different from the following two aspects.

Soft Integration vs. Hard Assignment. As illustrated in Section (2.3) of the main paper, our point-memory bank regards the similarities S_{cos} between the test point cloud feature and the feature memory, F_{mem}, as weights, which are adopted for weighted summation of the one-hot label memory, T_{mem}. This can be viewed as a soft label integration. Instead, k-NN utilizes S_{cos} to search the k nearest neighbors from the training set, and directly outputs the category label with the maximum number of samples within the k neighbors. Hence, k-NN conducts a hard label assignment, which is less adaptive than the soft integration. Additionally, our point-memory bank can be accomplished simply by two matrix multiplications and requires no sorting, which is more efficient for hardware implementation.

All Samples vs. k Neighbors. Our point-memory bank integrates the entire label memory with different weights. This can take the semantics of all training samples into account for classification. In contrast, k-NN only involves the nearest k neighbors to the test sample, which discards the sufficient category knowledge from other training samples.

Performance Comparison. In Table 3, based on the point cloud features extracted by our non-parametric encoder, we implement the top-k version of point-memory bank for comparison with k-NN, which only aggregates the label memory of the training samples with top-k similarities. As the neighbor number k increases, k-NN's performance is severely harmed due to its hard label assignment, while our point-memory bank attains the highest accuracy by utilizing all 9,840 samples for classification, indicating their different characters.

1.5. Point-NN vs. PnP-3D?

One previous work, PnP-3D [38], proposes local-global 3D processing modules that are plugged into other 3D models for performance improvement. Different from Point-NN's plug-and-play, PnP-3D introduces additional learnable parameters and requires to re-train the baseline networks from scratch, which is time-consuming. In contrast, our Point-NN is non-parametric and enhances the baseline directly during inference. In Table 4, we compare Point-NN with PnP-3D respectively on PAConv [56] for shape classification and VoteNet [5] for 3D object detection. As shown, our method contributes to similar performance enhancement on the benchmarks, while brings no extra parameters or re-training. In the table, we report the additional time for Point-NN to construct the point-memory bank before plug-and-play, which are 48 seconds and 9.3 minutes for the two tasks.

2. Related Work

3D Point Cloud Analysis. As the main data form in 3D, point clouds have stimulated a range of challenging tasks, including shape classification [29, 31, 35–37, 55], scene segmentation [4, 25, 71], 3D object detection [5, 18, 21, 34, 42, 64], 3D vision-language learning [16, 52, 62, 73]. Existing solutions as backbone networks can be categorized into projection-based and point-based methods. To handle the irregularity and sparsity of point clouds, projection-based methods convert them into grid-like data, such as tangent planes [46], multi-view depth maps [13, 17, 44, 62, 73], and 3D voxels [30, 32, 41, 67]. By doing this, the efficient 2D networks [19] and 3D convolutions [30] can be adopted for robust point cloud understanding. However, the projection process inevitably causes geometric information loss and quantization error. Point-based methods directly extract 3D patterns upon the unstructured input points to alleviate this loss of information. The seminal PointNet [35] utilizes shared MLP layers to independently extract point features and aggregate the global representation via a max pooling. PointNet++ [36] further constructs a multi-stage hierarchy to encode local spatial geometries progressively. Since then, the follow-up methods introduce advanced yet complicated local operators [31, 56] and global transformers [1, 8, 9, 14, 15, 61, 66] for spatial geometry learning. In this paper, we follow the paradigm of more popular point-based methods, and propose a pure non-parametric network, Point-NN, with its two promising applications. For the first time, we verify the effectiveness of non-parametric components for 3D point cloud analysis.

Local Geometry Operators. Referring to the inductive bias of locality [19, 24], most existing 3D models adopt delicate 3D operators to iteratively aggregate neighborhood features. Following PointNet++ [36], a series of methods utilize shared MLP layers with learnable relation modules for local pattern encoding, e.g., fully-linked webs [69], structural relation network [7], and geometric affine module [31]. Some methods define irregular spatial kernels and introduce point-wise convolutions by relation mapping [28], Monte Carlo estimation [20, 51], and dynamic kernel assembling [56]. Inspired by graph networks, DGCNN [50] and others [26, 47] regard points as vertices and interact local geometry through edges. Transformers [25, 70] are also introduced in 3D for attention-based feature communication. CurveNet [54] proposes generating hypothetical curves for point grouping and feature aggregation. Unlike all previous methods with learnable operators, Point-NN adopts non-parametric trigonometric functions to reveal the spatial geometry within local regions, and Point-PN further appends simple linear layers on top with high performance-parameter trade-off.

Positional Encodings. Transformers [49] represent input signals as an orderless sequence and implicitly utilize positional encodings (PE) to inject positional information. Typically, trigonometric functions are adopted as the non-learnable PE [11] to encode both absolute and relative positions, each dimension corresponding to a sinusoid. For vision, PE can also be learnable during training [6] or online predicted by neural networks [3, 70]. Another work [40] indicates that deep networks can learn better high-frequency variation given a higher dimensional input. Tancik *et al*. [45] interpret it as Fourier transform to learn high-frequency functions in low dimensional problems. NeRF [33] utilizes trigonometric PE to enhance the MLPs for better neural scene representations, but in a different formulation from the Transformer's. In contrast, we extend the non-learnable trigonometric PE of Transformer for specialized raw-point embedding and local geometry extraction, other than serving as an accessory in previous learnable networks. By doing this, the non-parametric encoder of Point-NN can effectively capture low-level spatial patterns complementary to the already trained 3D models.

3. Implementation Details

Point-NN. The non-parametric encoder of Point-NN contains 4 stages. Each stage reduces the point number by half via FPS, and doubles the feature dimension during feature expansion. For shape classification, the initial feature dimension C_I is set to 72, and the final dimension C_G of global representation is 1,152. The neighbor number k of k-NN is 90 for all stages. We set the two hyperparameters α, β in $\mathrm{PosE}(\cdot)$ as 1000 and 100, respectively, referring to Equation (6) and (7) in the main paper. For part segmen-

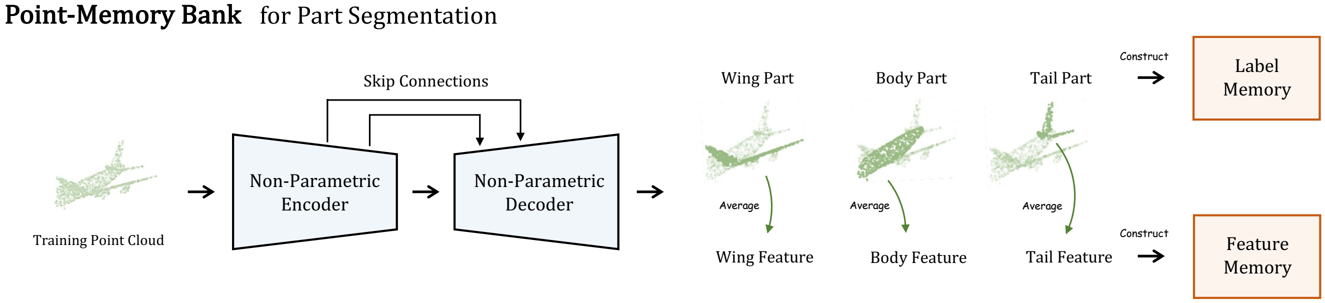

Figure 3. **Point-Memory Bank for Part Segmentation.** We first utilize the non-parametric encoder and decoder to extract the point-wise features of the point cloud in the training set. Then, we average the point features with the same part label to obtain the part-wise features, and construct them as the feature memory.

Grouping Method	Acc.
Ball Query	78.5
k-NN	**81.8**

Feature Expand	Acc.
w/o	70.0
w	**81.8**

Pooling	Acc.
Max	80.4
Ave	77.1
Both	**81.8**

Table 5. **Ablation Study of Non-Parametric Encoder.** We ablate the grouping method for local neighbors, feature expansion by concatenation, and pooling operation for feature aggregation. We report the classification accuracy (%) on ModelNet40 [53].

tation, we extend the non-parametric encoder into 5 stages for fully aggregating multi-scale 3D features. We set the initial feature dimension C_I as 144, and the neighbor number k as 128. We appended a non-parametric decoder with skip connections in each stage, which concatenate the propagated point features in the decoder with the corresponding ones from the encoder. As shown in Figure 3, we construct the segmentation point-memory bank by storing the part-wise features and labels from the training set, which largely saves the GPU memory. During inference, each point-wise feature of the test point cloud conducts similarity matching with the part-wise feature memory for segmentation.

Point-PN. For the parametric variant, we decrease C_I to 36 and the neighbor number k to 40 for lightweight parameters and efficient inference. We adopt the bottleneck architecture with a ratio 0.5 for the two cascaded linear layers after the *Geometry Extraction* step. The initial parametric raw-point embedding consists of only one linear layer, and the final classifier contains three linear layers as existing methods [31, 37]. Specially, for the second 2-layer linear layers, i.e., the '2' of '1+2', at the first stage of Point-PN, we stack them twice for better extracting elementary 3D patterns at shallow layers. For shape classification, we train Point-PN for 300 epochs with a batch size 32 on a single RTX 3090 GPU. On ModelNet40 [53], we adopt SGD

Magnitude α	1	10	50	**100**	200	500
Acc. (%)	68.3	77.9	81.1	**81.8**	81.4	81.0

Table 6. **Magnitude α in Trigonometric Functions.** We report the classification accuracy of Point-NN on ModelNet40 [53].

Wavelength β	10	100	**500**	1000	2000	3000
Acc. (%)	51.3	80.4	**81.8**	74.5	73.1	72.9

Table 7. **Wavelength β in Trigonometric Functions.** We report the classification accuracy of Point-NN on ModelNet40 [53].

optimizer with a weight decay 0.0002, and cosine scheduler with an initial learning rate 0.1. On ScanObjectNN [48], we adopt AdamW optimizer [23] with a weight decay 0.05, and cosine scheduler with an initial learning rate 0.002. We follow the data augmentation in PointMLP [31] and Point-NeXt [37] respectively for ModeNet40 and ScenObjectNN datasets. For part segmentation, we simply utilize the same learnbable decoder and training settings as CurveNet [54] for a fair comparison.

Plug-and-play. For part segmentation and 3D object detection, concurrently running an extra Point-NN to enhance existing models would be expensive in both time and memory. Thus, referring to SN-Adapter [65], we directly adopt the encoders of already trained models to extract point cloud features, and only apply our point-memory bank on top for plug-and-play. In this way, we can also achieve performance improvement by leveraging the complementary knowledge between similarity matching and traditional learnable classification heads.

Ratio (%)	1	5	10	20	40	80	100
Acc. (%)	39.5	64.2	70.3	75.0	77.9	80.8	**81.8**
Mem. (G)	3.84	3.87	3.93	4.05	4.26	4.82	5.21

Table 8. **Point-Memory Bank with Different Sizes.** We randomly sample different ratios of ModelNet40 [53] to construct the point-memory bank and report the classification accuracy with GPU memory consumption.

4. Additional Ablation Study

Non-Parametric Encoder. In Table 5, we further investigate other designs at every stage of Point-NN's non-parametric encoder. As shown, k-NN performs better than ball query [36] for grouping the neighbors of each center point since the ball query would fail to aggregate valid geometry in some sparse regions with only a few neighboring points. Expanding the feature dimension by concatenating the center and neighboring points can improve the performance by +5.3%. This is because each point obtains larger receptive fields as the network stage goes deeper and requires higher-dimensional vectors to encode more spatial semantics. For the pooling operation after geometry extraction, we observe applying both max and average pooling achieves the highest accuracy, which can summarize the local patterns from two different aspects.

Hyperparameters in Trigonometric Functions. In Table 6 and 7, we show the influence of two hyperparameters in trigonometric functions of Point-NN. We fix one of them to be the best-performing value (α as 100, β as 500), and vary the other one for ablation. The combination of the magnitude α and wavelength β control the frequency of the channel-wise sinusoid, and thus determine the raw point encoding for different classification accuracy.

Point-Memory Bank with Different Sizes. As default, we construct the feature memory by the entire training-set point clouds. In Table 8, we report how Point-NN performs when partial training samples are utilized for the point-memory bank. As shown, Point-NN can attain 60.1% classification accuracy with only 10% of the training data, and further achieves 70.1% with 40% data, which is comparable to the performance of 100% ratio but consumes less GPU memory. This indicates Point-NN is not sensitive to the memory bank size and can perform favorably with partial training-set data.

References

[1] Anthony Chen, Kevin Zhang, Renrui Zhang, Zihan Wang, Yuheng Lu, Yandong Guo, and Shanghang Zhang. Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. *arXiv preprint arXiv:2303.08129*, 2023. 4

[2] Yuxiao Chen, Jianbo Yuan, Yu Tian, Shijie Geng, Xinyu Li, Ding Zhou, Dimitris N. Metaxas, and Hongxia Yang. Revisiting multimodal representation in contrastive learning: from patch and token embeddings to finite discrete tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[3] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 4

[4] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018. 4

[5] Zhipeng Ding, Xu Han, and Marc Niethammer. Votenet: A deep learning label fusion method for multi-atlas segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 202–210. Springer, 2019. 3, 4

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[7] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019. 4

[8] Kexue Fu, Peng Gao, ShaoLei Liu, Renrui Zhang, Yu Qiao, and Manning Wang. Pos-bert: Point cloud one-stage bert pre-training. *arXiv preprint arXiv:2204.00989*, 2022. 4

[9] Kexue Fu, Peng Gao, Renrui Zhang, Hongsheng Li, Yu Qiao, and Manning Wang. Distillation with contrast is all you need for self-supervised point cloud representation learning. *arXiv preprint arXiv:2202.04241*, 2022. 4

[10] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2

[11] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017. 4

[12] Shijie Geng, Jianbo Yuan, Yu Tian, Yuxiao Chen, and Yongfeng Zhang. HiCLIP: Contrastive language-image pre-training with hierarchy-aware attention. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[13] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a sim-

ple and effective baseline. *arXiv preprint arXiv:2106.05304*, 2021. 4

[14] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 4

[15] Ziyu Guo, Xianzhi Li, and Pheng Ann Heng. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. *arXiv preprint arXiv:2302.14007*, 2023. 4

[16] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. Calip: Zero-shot enhancement of clip with parameter-free attention. *arXiv preprint arXiv:2209.14169*, 2022. 2, 4

[17] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2021. 4

[18] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11873–11882, 2020. 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[20] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 4

[21] Peixiang Huang, Li Liu, Renrui Zhang, Song Zhang, Xinli Xu, Baichao Wang, and Guoyi Liu. Tig-bev: Multi-view bev 3d object detection via target inner-geometry learning. *arXiv preprint arXiv:2212.13979*, 2022. 4

[22] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. *arXiv preprint arXiv:2210.01055*, 2022. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 4

[25] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 4

[26] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 4

[27] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 388–404. Springer, 2022. 2

[28] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 4

[29] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 326–342. Springer, 2020. 4

[30] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *arXiv preprint arXiv:1907.03739*, 2019. 4

[31] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 4, 5

[32] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8500–8508, 2019. 4

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 4

[34] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2906–2917, October 2021. 4

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 4

[36] 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, 3, 4, 6

[37] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv preprint arXiv:2206.04670*, 2022. 4, 5

[38] Shi Qiu, Saeed Anwar, and Nick Barnes. Pnp-3d: A plug-and-play for 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1312–1319, 2021. 3, 4

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[40] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 4

[41] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 4

[42] 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. 4

[43] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 3

[44] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 4

[45] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 1, 4

[46] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018. 4

[47] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. Rgcnn: Regularized graph cnn for point cloud segmentation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 746–754, 2018. 4

[48] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. 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*, pages 1588–1597, 2019. 2, 5

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[50] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 4

[51] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 4

[52] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual and language learning. *arXiv preprint arXiv:2209.14941*, 2022. 4

[53] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2, 3, 5, 6

[54] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. *arXiv preprint arXiv:2105.01288*, 2021. 4, 5

[55] Jianwen Xie, Yifei Xu, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14976–14985, 2021. 4

[56] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021. 3, 4

[57] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3056–3064, 2021. 2

[58] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. *arXiv preprint arXiv:2212.05171*, 2022. 2

[59] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 2

[60] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2

[61] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv preprint arXiv:2205.14401*, 2022. 4

[62] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 2, 4

[63] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Hongsheng Li, Yu Qiao, and Peng Gao. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. *arXiv preprint arXiv:2303.02151*, 2023. 2

[64] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-aware transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022. 4

[65] Renrui Zhang, Liuhui Wang, Ziyu Guo, and Jianbo Shi. Nearest neighbors meet deep neural networks for point cloud analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1246–1255, 2023. 5

[66] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. *arXiv preprint arXiv:2212.06785*, 2022. 4

[67] Renrui Zhang, Ziyao Zeng, Ziyu Guo, Xinben Gao, Kexue Fu, and Jianbo Shi. Dspoint: Dual-scale point cloud recognition with high-frequency fusion. *arXiv preprint arXiv:2111.10332*, 2021. 4

[68] Renrui Zhang, Ziyao Zeng, Ziyu Guo, and Yafeng Li. Can language understand depth? In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6868–6874, 2022. 2

[69] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5565–5573, 2019. 4

[70] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 4

[71] Zongyue Zhao, Min Liu, and Karthik Ramani. Dar-net: Dynamic aggregation network for semantic scene segmentation. *arXiv preprint arXiv:1907.12022*, 2019. 4

[72] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2

[73] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, and Peng Gao. Pointclip v2: Adapting clip for powerful 3d open-world learning. *arXiv preprint arXiv:2211.11682*, 2022. 2, 4