DepthEnhanced PointNet++ (DEP-Net): An Optimization of PointNet++ for Incomplete Point Clouds Using Projected Depth Maps

Anonymous Wirter,XJTU,Computer Science

***Abstract*—Since the introduction of PointNet [1], many current research methodologies have shifted their focus towards directly processing point cloud data using deep learning techniques. However, these studies predominantly concentrate on the analysis of complete point clouds for 3D affordance analysis, and most existing models experience a decline in performance when dealing with incomplete point cloud inputs. The research data from 3D AffordanceNet [2] indicates that due to the loss of geometric information in partial point clouds relative to complete ones, the performance of classic networks such as PointNet++ [3], DGCNN [4], and U-Net [5] decreases by 2.3% , 4.2%, and 4.4%, respectively, compared to their performance with complete point clouds.**

Inspired by point cloud completion networks [6] [7] [8] [9], we initially designed a self-view fusion network that utilizes multi-view depth image information to observe the incomplete self-shape and generate a compact global shape. By acquiring complete view features through the completion network, these are then inputted into the PointNet++ network to further perform downstream semantic segmentation tasks.

***Index Terms*—DEP-Net, 3D Affordance analysis, incomplete point cloud, AffordanceNet, PointNet++, multi-view depth image, semantic segmentation, Depth Enhanced**

I. Introduction

THE concept of affordance focuses on the interaction between humans and their environment. The capacity to understand how humans interact with objects through visual cues, known as visual affordance, is essential for research in vision-guided robotics. Studies in this area cover tasks such as the classification, segmentation, and inference of visual affordance.

Given a 3D point cloud object without known affordance estimations, the task of point cloud affordance estimation aims to predict the types of affordances the object supports and to estimate the per-point probability distribution of its affordances.

Although a complete point cloud input can provide more detailed geometric information for affordance estimation, in real-world scenarios, point clouds obtained from the real world are often imperfect and noisy due to occlusions among objects, limited precision of scanning devices, etc. We usually only have access to partial views of 3D shapes, represented as partial point clouds. Hence, an important task of interest is the affordance estimation of partial

Many past studies on affordance analysis have focused on the 2D and 2.5D domains [10] [11] [12] [13], achieving commendable performance in visual tasks such as classification and semantic segmentation. Since the introduction of PointNet, many contemporary research methods have increasingly focused on directly processing point cloud data using deep learning techniques. However, these studies on 3D model affordance analysis are mostly concerned with complete point clouds. Data from the 3D AffordanceNet study [2] indicates that due to the loss of geometric information in partial point clouds relative to complete ones, the performance of classic networks such as PointNet++ [3], DGCNN [4], and U-Net [5] each experience a decline in performance compared to their use with complete point clouds.

Inspired by the point cloud completion network [6] [7] [8] [9], we can optimize the understanding of the global shape of point clouds and the recovery of details by supplementing information for feature fusion and point cloud repair.Generally, when humans observe objects that are difficult to distinguish, they adopt multi-view observations. Thus, it can be considered to use multi-view depth maps and original point cloud data for feature fusion to complete the missing geometric features.

In summary, this paper, inspired by related work in the point cloud completion field, proposes the use of depth maps to augment the original point cloud data. The depth maps are directly generated by projecting the point cloud itself from controllable viewpoints during the data preprocessing stage. This module aims to observe the partial input from different angles and learn effective descriptors to produce a globally coherent and complete feature representation.

Our contributions are as follows:

- We propose a novel approach that utilizes depth maps generated from multiple viewpoints to enhance the understanding of incomplete point clouds. This method improves the global shape representation, which is critical for downstream tasks such as semantic segmentation.
- We introduce the View Enhanced Depth Map (VEDM) module, which effectively fuses features from 2D depth maps with 3D point cloud features, leading to a more robust and complete representation of the point cloud data.
- We demonstrate through our experiments on the 3D AffordanceNet dataset that our DEP-Net model outperforms the classic PointNet++ in terms of mean Average Precision (mAP), Average Intersection Over Union (aIOU), and Mean Squared Error (MSE) on various affordance categories.
- Our work provides insights into the fusion of 2D and 3D data for point cloud analysis, opening up new avenues for

research in the field of 3D computer vision and robotics.

II. Related Work

In 2017, the pioneering PointNet [1], a deep learning-based technolmkmmlogy, was introduced, and since then, many current research methods have increasingly focused on directly processing point cloud data using deep learning techniques [5] [14] [15] [16]. Researchers start from raw point cloud data, directly processing it, which not only makes better use of the information from each point but also reduces noise and computational errors. However, the performance loss caused by the loss of geometric information due to missing parts of the point cloud for incomplete point cloud inputs is almost inevitable.

A. PointNet & PointNet++

PointNet: It directly processes point cloud data, using shared multilayer perceptrons (MLPs) to extract features from each point in the input point cloud data. Global features are obtained through max pooling.PointNet was designed following two main principles:

Permutation Invariance: Since point cloud data is unordered, when an NxD point cloud data is arbitrarily shuffled in the N dimension, it still represents the same object and does not affect the overall representation of the object. This is known as the permutation invariance of point clouds. For the characteristic that point clouds have permutation invariance, the designed network must be a symmetric function, that is:

$$f(x_1, x_2, ..., x_n) = f(x_{\pi_1}, x_{\pi_2}, ..., x_{\pi_n}), X_i \in R^D \quad (1a)$$

Common symmetric functions include SUM and MAX:

$$\begin{cases} f(x_1, x_2, ..., x_n) = & max\{x_1, x_2, ..., x_n\} \\ f(x_1, x_2, ..., x_n) = & x_1 + x_2 + ... + x_n \end{cases} \quad (2a)$$

PointNet uses the max function as the max pooling layer to aggregate the local features extracted from all points.

Rotation Invariance: The rotation invariance of point clouds means that when the point cloud data is subjected to certain rigid changes (rotation or translation), the coordinates (x, y, z) of all points change, but it still represents the same object. PointNet introduces a T-Net network to learn the rotation of point clouds, and then inputs the calibrated point clouds into PointNet to complete classification or segmentation tasks.

The structure of PointNet is very concise. It achieved state-of-the-art performance using simple MLP + max pooling.

PointNet only considers global features, directly and brutally max pooling all the points into a global feature, thus losing the local information of each point, and the connection between local points is not learned by the network. To solve this problem, PointNet++ was further proposed. PointNet++ introduces a hierarchical neural network that recursively applies PointNet to the nested partitions of the input point set. By utilizing metric space distances, PointNet++ can learn local features with continually increasing context scales.

B. 3DCTN

The 3D Convolution-Transformer Network (3DCTN) [17] integrates convolution into the Transformer to effectively learn local and global features of point cloud classification, and achieves competitive results using state-of-the-art classification methods. The network consists of two main modules: multi-scale local feature aggregation and global feature learning, both operating on the downsampled point set and implemented through graph convolution and Transformer, respectively.

Local Feature Aggregating Block: The local feature extraction module of 3DCTN inherits the concept of PointNet++, using Farthest Point Sampling (FPS) to obtain a subset of point clouds, referred to as the sampled point set. At the same time, to ensure the diversity of the receptive field of the sampling points, multi-scale neighborhoods for each sampling point are constructed through ball query grouping. For each neighborhood of the sampling points, the innovation of 3DCTN lies in proposing a context fusion method to encode and combine the coordinate and feature information of the neighborhood, followed by edge convolution to aggregate local features.

Global Feature Learning Block (GFL): 3DCTN uses the aggregated features $Y = \{y_i\}_{i=1}^{i=S}$ as input, where S is the number of points, and the Global Feature Learning (GFL) module has two main components: self-attention mechanism and position encoding. There are no embedding inputs (words) in the GFL block, as Y in the LFA block can be considered as the embedded input for the GFL block.

III. DEP-Net

The input to DEP-Net is composed of three parts: an incomplete low-resolution point cloud $P_{in} \subseteq \mathbb{R}^{N\times 3}$, N_V camera positions $VP \subseteq \mathbb{R}^{N_V\times 3}$ (three orthogonal views), and N_V depth maps $D \subseteq \mathbb{R}^{N_V\times 1\times H\times W}$. Given these inputs, our objective is to use the depth maps to ameliorate the loss of geometric information in the incomplete point cloud and obtain an enhanced point cloud feature representation $P_2 \subseteq \mathbb{R}^{N_2\times 3}$. The overall architecture, depicted in Figure 1, includes two main components: the VEDM module and the traditional PointNet++ network.

A. View Enhance by Depth Maps (VEDM)

The View Enhanced Depth Map (VEDM) module utilizes a series of self-generated depth maps to create an improved global shape representation. It captures partial inputs from various viewpoints and employs effective descriptors to synthesize a shape that is both globally consistent and complete. To begin, we employ a point-based 3D network to derive global features, denoted as F_p, from the input point cloud P_{in}. Concurrently, we extract a set of view features, represented as F_d, from N_V different depth maps through a CNN-based 2D network. We incorporate established backbone networks for these tasks. Specifically, we utilize PointNet++ with a three-layer abstraction approach to progressively process P_{in}, while the 2D features from the depth maps are captured using the ResNet-18 architecture.

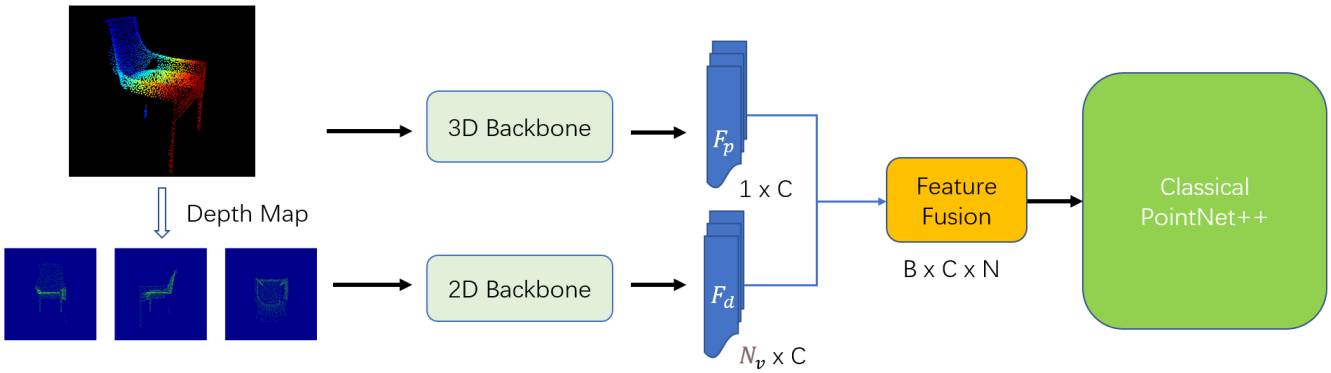

Fig. 1. structure of the DEP-NET

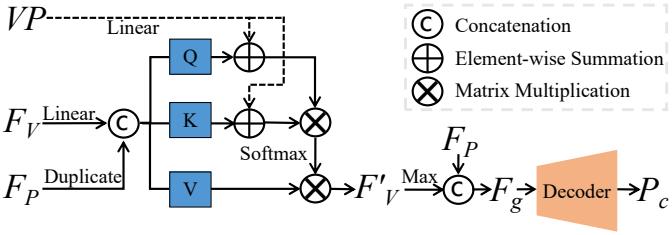

Fig. 2. Illustration of the feature fusion module in the SVDFormer.

In our approach to effectively integrating the cross-modal features described earlier, we took inspiration from the SVDFormer's research insights. The domain difference between 2D and 3D data representations leads to suboptimal results when simply concatenating these features. To overcome this, we applied the feature fusion strategy from SVDFormer to combine the global features F_p from the point cloud and the view features F_d from the depth maps, resulting in a comprehensive global shape descriptor F_g. This descriptor is then processed by a decoder to produce the global features P_c. The decoder employs 1D Conv-Transpose layers to convert F_g into individual point features and uses self-attention layers to estimate the 3D coordinates of each point. Building on techniques from prior works, we merge the generated P_c with the input P_{in} and resample this combined data to produce an initial coarse shape P_0.

As depicted in the figure, the feature fusion process begins with the transformation of F_V into query, key, and value components through linear layers, under the influence of the global shape features F_P. The next step is to enhance the distinctiveness of the view features by calculating attention weights. These weights are based on the queries and keys, and are adjusted according to the projected viewpoints VP. We linearly transform VP to a latent space and utilize them as positional cues to facilitate the feature fusion. By conducting element-wise multiplication, each element in F'_d amalgamates relational data from different views, steered by F_P. The culmination of this process is the derivation of the output shape descriptor F_g from F'_d, which is obtained through max pooling.

B. *PointNet++*

For the affordance analysis network, we employed the classic PointNet++ network for downstream semantic segmentation tasks. PointNet++ introduced a hierarchical neural network that recursively applies PointNet to nested subdivisions of the input point set. By leveraging metric space distances, PointNet++ is able to learn local features with incrementally increasing contextual scales. The network structure is illustrated in 3

Each set abstraction layer group in the network primarily consists of three parts: Sampling layer, Grouping layer, and PointNet layer.

- Sample layer: Farthest Point Sampling (FPS) is used to sample the input points, which, compared to random sampling, can cover the entire sampling space more effectively.
- Grouping layer: Utilizes the sampled centroid points to divide the point set into N local regions using the Ball query method;
- PointNet layer: Performs convolution and max pooling on each local region provided by the grouping layer, with the resulting features serving as local features for the respective centroid point.

After feature extraction is completed by the PointNet++ network, for each affordance type, we pass the features to multiple classifiers and use the sigmoid function to obtain posterior scores. Classifiers are individually set up for each affordance category, while the backbone network is shared.

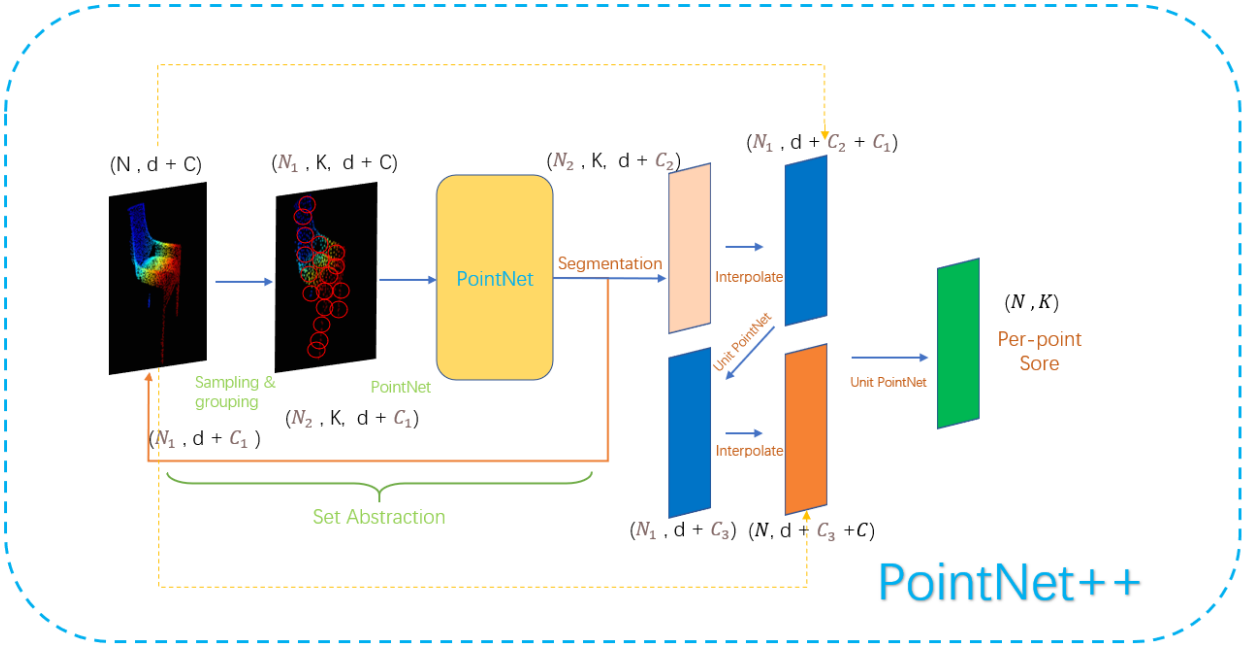

Fig. 3. structure of the pointnet++

We use the Cross-Entropy Loss function L_{CE} to train the network.

IV. Experiment

A. Dataset

In this paper, we utilize the 3D AffordanceNet dataset [2], which is an extension based on PartNet [18] and incorporates fine-grained part hierarchy information of 3D shapes from large 3D CAD model datasets such as ShapeNet [19] and 3D Warehouse. 3D AffordanceNet, as introduced by Deng, et al. [2], is a functional affordance dataset based on 3D point cloud data, consisting of 56,307 well-defined affordance annotations for 22,949 point cloud shapes, covering 18 affordance categories and 23 semantic object categories. It is also the first large-scale dataset with well-defined probabilistic distribution annotations for affordances. Based on this dataset, we evaluate the affordance understanding capabilities of the PointNet++ network enhanced with depth maps and the classic PointNet++ network.

B. Training Details

We train our network on a point cloud dataset with shape sizes of 1024 points. For the original dataset, we obtain a subset of size $N * 1024 * 3$ using Farthest Point Sampling (FPS) during the data preprocessing process. We set the initial learning rate to 0.001 and use the Adam optimizer to optimize parameters, reducing the learning rate by half every 20 epochs. We train the network for 100 epochs with a batch size of 128. The weight decay for the Adam optimizer is set to 1e-8. We use the cosine annealing algorithm to adjust the learning rate, which can be described by the following equation:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{T_{cur}}{T_{max}}\pi)) \quad (3)$$

where η_t is the adjusted learning rate, η_{min} is the minimum learning rate, η_{max} is set as the initial learning rate, and T_{cur} is the number of the current epoch. We set the batch size to 128 and train the network for 100 epochs.

C. Results

TABLE I
Affordance Estimation Results

Metric	P_mAP	P_AUC	P_aIOU	P_MSE	D_mAP	D_AUC	D_aIOU	D_MSE
Avg	45.7	85.2	16.9	0.062	46.4	85.1	17.5	0.056
Grasp	43.2	81.2	14.4	0.003	39.0	79.5	13.0	0.003
Lift	80.6	96.2	45.6	0.0001	79.4	90.1	42.2	0.00005
Contain	41.9	83.3	13.2	0.005	44.2	83.6	13.1	0.005
Open	48.5	87.9	21.6	0.003	51.3	89.8	23.2	0.003
Lay	52.6	86.7	25.2	0.0006	43.3	83.8	19.9	0.0004
Sit	69.8	95.0	31.0	0.004	71.9	95.7	31.4	0.004
Support	45.5	86.5	11.2	0.013	46.9	88.3	14.9	0.011
Wrap.	20.0	71.3	3.6	0.002	18.3	66.6	4.4	0.003
Pour	47.0	88.4	17.8	0.002	48.6	87.8	18.7	0.002
Display	52.5	85.2	19.3	0.002	56.2	86.8	28.1	0.002
Push	24.1	84.9	5.7	0.0002	23.2	83.2	4.4	0.0002
Pull	36.5	86.1	11.5	0.0001	51.1	84.7	16.2	0.00006
Listen	42.1	84.2	13.4	0.0007	49.4	86.9	13.9	0.0003
Wear	15.3	64.1	2.4	0.0004	16.9	68.3	2.6	0.0003
Press	30.7	84.7	12.4	0.0006	34.8	86.6	10.6	0.0004
Move	37.8	79.6	6.2	0.025	37.0	79.3	10.2	0.022
Cut	43.3	90.2	13.5	0.0003	35.8	89.9	10.5	0.0003
Stab	92.6	98.8	37.8	0.0001	88.1	99.3	37.0	0.00009

We compared the performance of the DEP-Net, which incorporates depth map feature fusion, with the classic PointNet++ on the 3D AffordanceNet dataset. The results, as shown in figure IV-C, demonstrate that DEP-Net outperforms PointNet++ on several metrics, including mAP, a_IOU, and MSE,

with improvements of 0.7%, 0.6%, and a 9.7% decrease in MSE, respectively. Except for the MSE score, all others are displayed as percentages, with higher scores indicating better performance. The algorithms P and D represent PointNet++ [3] and DEP-Net (Ours), respectively. The terms Contain, Open, Sit, Support, Pour, Display, Pull, Listen, Wear, Press, Grasp, Lift, Lay, Wrap, Push, Move, Cut, and Stab represent different affordance categories. In the analysis of specific affordance categories, DEP-Net outperforms PointNet++ in predicting accuracies for affordances such as Contain, Open, Sit, Support, Pour, Display, Pull, Listen, Wear, and Press. However, for affordance categories like Grasp, Lift, Lay, Wrap, Push, Move, Cut, and Stab, our DEP-Net experienced some performance decline. Interestingly, the shapes that saw performance improvements tend to have flatter and generally larger surface areas compared to those where performance decreased, suggesting that DEP-Net may be less sensitive to minute edge details and features in convex regions.

V. Problems & Plans

A. Problems

For the overall network, the structure is not sufficiently streamlined. We believe there is redundancy between the feature extraction of 3D point clouds in the depth map feature extraction and the feature extraction of point clouds in the downstream semantic segmentation task. It might be possible to merge the two PointNet set abstraction feature extraction processes to optimize network efficiency.

B. Plans

With the significant advancements in the fields of natural language processing (NLP) and computer vision, Transformer models have demonstrated superior capabilities in learning global features. Consequently, they have been deployed in a variety of point cloud processing applications [20] [21] [22], including object classification, semantic scene segmentation, and object part segmentation. At the heart of the Transformer model lies the self-attention mechanism, which initially computes the similarity between pairs of embedded tokens, and then employs these similarities to create a weighted sum of all tokens, yielding a new set of outputs. This process enables each output token to be connected with every input token, which underpins the Transformer's adeptness at capturing global features. Therefore, replacing conventional convolutional operations in the network with Transformer models may enhance feature representation. This is the objective we aim to pursue in our future work. We anticipate that by leveraging the Transformer architecture, we can refine the network's ability to perceive detailed features within point clouds, thereby improving performance in predicting affordances.

VI. Conclusion

In this work, we introduced DEP-Net, an optimized version of PointNet++ designed to address the challenges of semantic segmentation in incomplete point clouds by leveraging projected depth maps. Our approach demonstrates that the integration of depth maps as additional features can enhance the performance of point cloud processing networks, particularly in the context of affordance analysis. The VEDM module within DEP-Net effectively captures the global shape and recovers missing geometric information, resulting in improved segmentation accuracy.

The experimental results on the 3D AffordanceNet dataset confirmed the efficacy of our method, showing that DEP-Net outperforms the classic PointNet++ in various affordance categories. This advancement suggests that the consideration of different views and the fusion of 2D and 3D data can lead to a more robust understanding of 3D shapes, which is crucial for applications in robotics and computer vision.

However, our approach also revealed certain limitations, such as potential redundancies in feature extraction and a lack of sensitivity to minute details in some affordance categories. Our future work will focus on streamlining the network structure to eliminate redundancies and exploring the integration of Transformer models to improve the network's capability to capture detailed features. By incorporating these advancements, we aim to push the boundaries of point cloud processing and affordance prediction, paving the way for more intelligent and capable vision-guided robotic systems.

VII. References Section

References

[1] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017.

[2] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3d affordancenet: A benchmark for visual object affordance understanding," 2021.

[3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017.

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, 2019.

[5] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Proceedings of the European Conference on Computer Vision*, 2020.

[6] Z. Zhu, H. Chen, X. He, W. Wang, J. Qin, and M. Wei, "Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator," 2023.

[7] Y. Cai, T. Shen, S.-S. Huang, and H. Huang, "Self-supervised depth completion guided by 3d perception and geometry consistency," 2023.

[8] Z. Wang, Q. Xu, F. Tan, M. Chai, S. Liu, R. Pandey, S. Fanello, A. Kadambi, and Y. Zhang, "Mvdd: Multi-view depth diffusion models," 2023.

[9] M. Elhashash and R. Qin, "Select-and-combine (sac): A novel multi-stereo depth fusion algorithm for point cloud generation via efficient local markov netlets," 2023.

[10] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," 2018.

[11] Y. Yang, W. Zhai, H. Luo, Y. Cao, J. Luo, and Z.-J. Zha, "Grounding 3d object affordance from 2d interactions in images," 2023.

[12] A. Myers, C. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015, pp. 1374–1381, 06 2015.

[13] Y. Yang, W. Zhai, H. Luo, Y. Cao, and Z.-J. Zha, "Lemon: Learning 3d human-object interaction relation from 2d images," 2023.

[14] Q. Lu, C. Chen, W. Xie, and Y. Luo, "Pointngcnn: Deep convolutional networks on 3d point clouds with neighborhood graph filters," *Computers Graphics*, vol. 86, pp. 42–51, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849319301748

[15] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual mlp framework," 2022.

[16] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on $\mathcal{X}$-transformed points," 2018.
[17] D. Lu, Q. Xie, L. Xu, and J. Li, "3dctn: 3d convolution-transformer network for point cloud classification," 2022.
[18] F. Yu, K. Liu, Y. Zhang, C. Zhu, and K. Xu, "Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation," 2022.
[19] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015. [Online]. Available: http://arxiv.org/abs/1512.03012
[20] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, p. 187–199, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1007/s41095-021-0229-5
[21] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," 2021.
[22] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," 2022.