

INTERPRETABLE AFFORDANCE DETECTION ON 3D POINT CLOUDS WITH PROBABILISTIC PROTOTYPES

Maximilian Xiling Li, Korbinian Rudolf, Nils Blank, Rudolf Lioutikov

Intuitive Robots Lab

Karlsruhe Institute of Technology, Germany

{maximilian.li, nils.blank, lioutikov}@kit.edu

ABSTRACT

Robotic agents need to understand how to interact with objects in their environment, both autonomously and during human-robot interactions. Affordance detection on 3D point clouds, which identifies object regions that allow specific interactions, has traditionally relied on deep learning models like PointNet++, DGCNN, or PointTransformerV3. However, these models operate as black boxes, offering no insight into their decision-making processes. Prototypical Learning methods, such as ProtoPNet, provide an interpretable alternative to black-box models by employing a “this looks like that” case-based reasoning approach. However, they have been primarily applied to image-based tasks. In this work, we apply prototypical learning to models for affordance detection on 3D point clouds. Experiments on the 3D-AffordanceNet benchmark dataset show that prototypical models achieve competitive performance with state-of-the-art black-box models and offer inherent interpretability. This makes prototypical models a promising candidate for human-robot interaction scenarios that require increased trust and safety.

1 Introduction

The ability of robotic agents to effectively operate in real-world environments hinges on their ability to identify possible interactions with their surroundings. This involves navigating through spaces [1, 2], interacting with various objects [3, 4], and engaging with other agents or humans [5]. Central to this understanding is the concept of affordance [6], which refers to the potential interactions an environment offers. For instance, in a kitchen environment, a robot must recognize which utensils can perform specific tasks, such as containing liquids or cutting ingredients, and determine safe grasping points [7]. This is particularly crucial in scenarios involving handovers to human collaborators [8, 9, 10].

Despite significant advancements in sensor quality, existing 2D RGB and 2.5D RGB-D datasets [11, 12, 13] lack detailed geometry information about object shapes in 3D space. However, such information is likely to be highly important to humans to detect a flat surface as *sittable* or recognize likely grasping points. 3D-AffordanceNet [14] addresses this gap by offering a benchmark dataset of 3D model point clouds with point-wise affordance probability scores obtained from human annotations. However, the task of 3D affordance detection has primarily used conventional deep learning architectures such as PointNet++ [15] or DGCNN [16] with black box reasoning, making them less suitable for scenarios requiring increased trust and safety.

Explainable AI (XAI) has gained importance in developing transparent and trustworthy AI systems. Post-hoc interpretation methods aim to explain why a trained model made a specific decision. Techniques like GradCAM [17] have been widely used to analyze convolutional neural networks (CNNs) for image processing and have been adapted to point cloud processing networks [18]. In contrast to post-hoc methods, inherently interpretable models are indepen-

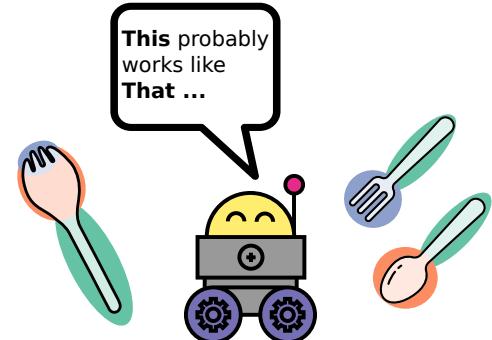


Figure 1: Prototypes provide explanations by displaying the similarities of learned representations to image features.

dent of external methods and produce their own explanations. Prototypical Parts Networks like ProtoPNet [19] and its successors [20, 21, 22] offer inherent interpretability and have been successfully applied to image classification tasks. The inherent interpretability is provided by a prototype layer, which computes similarity scores between the input embeddings and the stored prototype vectors. The prototype learning network, therefore, offers case-based reasoning in the manner of “this looks like that”. Probabilistic prototypes [22] further improve on this by learning a probability distribution as their prototypes, thus outputting confidence scores in addition to the prototype similarities.

Our main contribution is the integration of prototypes into point cloud processing models for interpretable 3D affordance detection. We build upon the probabilistic prototype formulation proposed in [22] for the interpretability and rely on state-of-the-art point cloud segmentation models as PointNet++ [15] as feature encoder. We analyze the performance of the prototypical point cloud model on the 3D AffordanceNet benchmark and demonstrate the effectiveness of prototypes in comparison to state-of-the-art black-box methods. Analyzing the learned prototypes provides insights into the network’s internal reasoning process and shows the prototypes’ usability as an explanation.

2 Related Work

Affordance Detection. Affordance detection seeks to identify which region of an input scene enables specific interactions. Early research in computer vision focused on pixel-wise affordance detection on RGB images using convolutional neural networks (CNNs) [23, 24, 25, 26]. Recent studies propose transformer-based architectures [27, 28, 29] and pretrained foundation models [30, 31] for affordance detection. Early research on affordance detection for robotics utilized 2.5D RGB-D images [32, 33, 34]. However, these methods do not account for the entire 3D geometry of objects.

PartNet [35] is a dataset of 3D object models with part-level annotations originally intended for object detection tasks. 3D-AffordanceNet [14] builds on this dataset to provide the largest benchmark dataset with probabilistic affordance scores for 23K object shapes and 18 affordance labels. The 3D-AffordanceNet benchmark includes experiments using PointNet++ [15] and DGCNN [16]. LG-AffordNet [36] proposes a novel local geometry descriptor for encoding the 3D point cloud. OpenAD [37] integrates a CLIP text encoder into the network architecture for open vocabulary affordance detection. DTNet [38] proposes a transformer-based architecture. The models were evaluated on the 3D-AffordanceNet based on their capability of affordance detection but lack interpretability since they rely on black-box reasoning.

Our method introduces interpretability to affordance detection models through a layer containing probabilistic prototypes. This approach applies to any point-based model and increases the transparency of its decision-making process while achieving competitive performance on the 3D-AffordanceNet benchmark.

Interpretable Prototype Learning Prototype learning approaches have been used as inherently interpretable methods for image classification. Typically, these architectures base the inference on the similarities of the input embeddings to prototype vectors. A prototype layer before the output layer calculates the similarities and passes them to the output layer for inference. The prototypes are network parameters optimized during the backpropagation of the loss through the entire network. Prototype learning approaches based on autoencoders offer high interpretability, as the learned prototype can be reconstructed from the latent space [39]. However, these methods are unsuitable for segmentation tasks because the learned prototypes represent entire images, complicating their use for per-pixel segmentation.

Parts-based approaches like ProtoPNet [19, 20] and its successors [40, 41, 22] are based on latent image patches of size 1×1 . They offer case-based reasoning in a “this looks like that” manner. ProtoSeg successfully adapted ProtoPNet’s approach for semantic image segmentation [21]. Other prototype approaches for image segmentation propose using a clustering algorithm like EM for learning the prototypes [42, 43].

However, these methods focus on the 2D RGB image domain. We propose using interpretable prototype learning for 3D point cloud segmentation to understand the model’s output better and increase trust.

Prototypes for Point Cloud Segmentation. Point cloud segmentation of LiDAR or RGB-D scenes remains a significant research focus for developing autonomous agents like household robots or autonomous cars. This section highlights the use of prototype learning for segmentation tasks on point clouds. ProtoTransfer [44] proposes a cross-modal, hybrid learning scheme with shared prototypes between 2D image and 3D LiDAR scan modalities. NAPL [45] introduces a second prototype learning branch to the network to dynamically learn the number of prototypes per class. Other approaches utilize prototype learning for few- and zero-shot semantic segmentation of 3D scenes [46, 47, 48]. Recent methods also apply prototype learning for part [49] and instance [50] segmentation of 3D model point clouds.

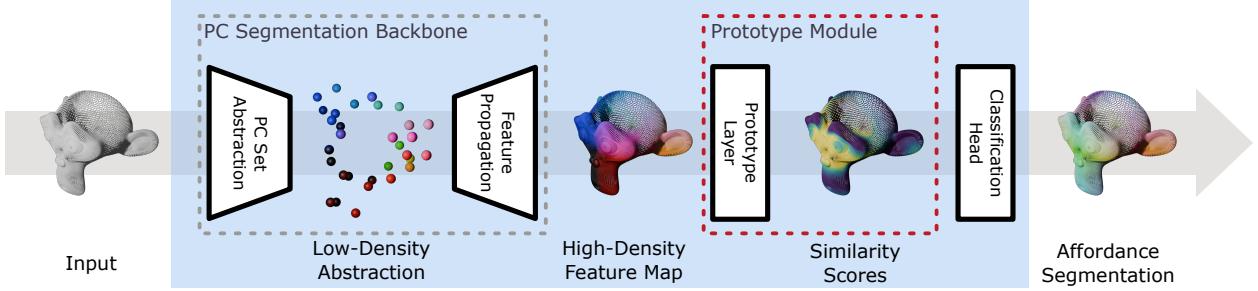


Figure 2: Architecture Overview of extended point cloud processing model: A point cloud segmentation backbone, e.g., PointNet++ or DGCNN, produces a feature map. A prototype layer computes prototype similarity scores from which the classification head generates the segmentation maps.

While these approaches use prototypical methods for point cloud segmentation, they do not consider the inherent interpretability offered by prototypes in a segmentation task [42]. We explicitly integrate the prototypes to increase interpretability.

3 Point Cloud Networks with Inherent Interpretability

This section integrates the probabilistic prototype layer introduced in [51] into a point cloud processing model. As backbone models, we use established point cloud networks, such as PointNet++ [15], DGCNN [16], and PointTransformerV3[51]. An overview of the resulting architecture is depicted in Figure 2.

3.1 Preliminaries

For affordance detection on 3D point clouds, the point cloud X is defined as a set of S points $\{x_1, x_2, \dots, x_S\}$, where each point $x_i \in \mathbb{R}^3$ represents the Cartesian coordinates (x, y, z) in 3D space. Each point cloud has a ground truth annotation Y with affordance labels $\{y_1, y_2, \dots, y_S\}$ for each point x_i . Each point label y_i is the probability score for affordance class $a \in A$. A point cloud neural network, such as PointNet++ or DGCNN, produces a point cloud embedding Z , where each point, in addition to the Cartesian coordinates, holds a latent feature vector $z \in \mathbb{R}^D$ with D feature dimensions.

3.2 Probabilistic Prototypes

We employ the probabilistic prototypes for the prototype module on the hypersphere defined in [22]. Each probabilistic prototype p is a triplet (α, μ, σ) with anchor vector $\alpha \in \mathbb{R}^D$, mean $\mu \in \mathbb{R}$ and standard deviation $\sigma \in \mathbb{R}$.

The prototype module consists of two layers that compute the density for cosine similarities of the embeddings to the anchors. The first layer calculates the cosine similarity $s(z|\alpha)$ between a latent vector z to a prototype anchor α . The second layer returns the probability density function (PDF) activation of a learned distribution over cosine similarities. The PDF for a prototype is parametrized by the mean similarity μ and standard deviation σ . As in [22], we use the truncated Gaussian distribution $\tau(s; \mu, \sigma)$ with bounds $[-1, 1]$ as PDF since the cosine similarity is bound to the same interval. The formulation as a probabilistic prototype with activation $\phi_{HyperPG}(z|p) = \tau(s(z, \alpha_p); \mu_p, \sigma_p)$ with learned standard deviation σ allows for different degrees of specialization between the prototypes.

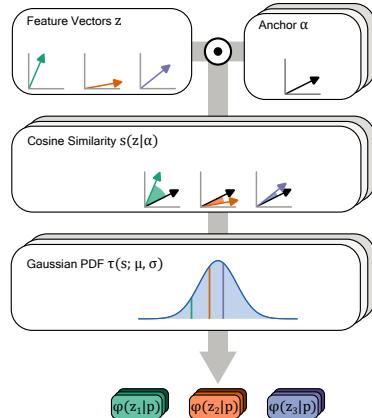


Figure 3: Illustration of the probabilistic prototypes on the hypersphere [22].

3.3 Loss Functions

The prototypes are network parameters treated as such during the training. Therefore, the loss function must incorporate the prototypes' structuring and optimization. Prior work on interpretable prototypes proposes a multi-objective loss function, which includes a task-specific loss, as well as clustering and separation losses for prototype assignment [19, 22]. For the task-specific loss, state-of-the-art methods on the 3D-AffordanceNet benchmark [14, 36] proposes the sum of the cross-entropy loss \mathcal{L}_{CE} and Dice loss $\mathcal{L}_{\text{Dice}}$ as a task-specific loss. The cross-entropy loss encourages the model to produce correct predictions. The Dice loss, as defined in [14, 36], is employed to mitigate the class imbalance between different affordance classes, especially the inhibited point cloud regions without any affordance annotation.

To optimize the prototype assignments, ProtoPNet introduces in [19] a cluster loss $\mathcal{L}_{\text{Clst}}$ and a separation loss \mathcal{L}_{Sep} for class-specific assignments based on the Euclidean distance. In [22], the losses are fitted to work on a similarity metric. Combining the cluster and separation losses structures the latent space by forming dense clusters of embeddings for the same affordance while keeping different affordance clusters distant. The cluster loss for the affordance-based prototypes

$$\mathcal{L}_{\text{Clst}} = -\frac{1}{N} \sum_{i=1}^N \frac{1}{|A|} \sum_{a \in A} \max_{p_a \in P_a} \max_{z_{i,a} \in Z_a} \phi(z_{i,a}, p_a) \quad (1)$$

encourages point embeddings $z_{i,a}$ belonging to affordance class a to be similar to one affordance-specific prototype p_a under the chosen similarity metric, resulting in tighter clusters in latent space. In contrast, the separation loss

$$\mathcal{L}_{\text{Sep}} = \frac{1}{N} \sum_{i=1}^N \frac{1}{|A|} \sum_{a \in A} \max_{p_{-a} \notin P_a} \max_{z_{i,a} \in Z_a} \phi(z_{i,a}, p_{-a}) \quad (2)$$

aims to increase the distances to other prototypes not belonging to affordance class a .

To train the entire model to achieve both high task performance and well-shaped prototype clusters, the multi-objective loss

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{Clst}} + \mathcal{L}_{\text{Sep}} \quad (3)$$

is employed. In contrast to other prototypical part networks like ProtoPNet [19], no weighting of the different loss terms is required. The weighing mediates between significant differences in the scales of the losses, which is necessary when using the virtually unbound Euclidean distance in the cluster and separation loss. Since the cosine similarity and the PDF are significantly more restricted, the scales of the cluster and separation losses are closer, and weighing is unnecessary.

3.4 Network Architecture

The network architecture of prototypical networks for point clouds, as presented in Figure 2, is similar to prototypical networks for RGB images. A backbone model acts as an encoder and calculates a high-density feature map from the input point cloud. The prototype module calculates the activation as described above. The classification head processes the prototype activations, infers the prediction for the affordances, and transforms them into pseudo-probabilities using softmax.

4 Experimental Setup

4.1 Dataset Description

The experiments were performed on the 3D-AffordanceNet benchmark dataset [14]. The dataset comprises 22,949 object shapes with 23 semantic object categories and is annotated with 18 affordance labels. Each object shape is provided as a point cloud of 2048 points represented by Cartesian XYZ positions. As the 3D-AffordanceNet's testing dataset is not public, the official training and validation split of 16,082 and 2,285 object shapes is used. We extend the affordance class annotations with a background class *No Label* for all points without any affordance annotation and apply data augmentation in the form of *RandomJitter* ± 0.05 , *RandomShuffle* of the order of points, and *RandomRotation*.

4.2 Implementation Details

We test the prototypical network with a segmentation backbone based on PointNet++ [15], DGCNN [16], and PointTransformerV3 [51]. In the evaluation, we compared the prototypical network to baseline implementations of the

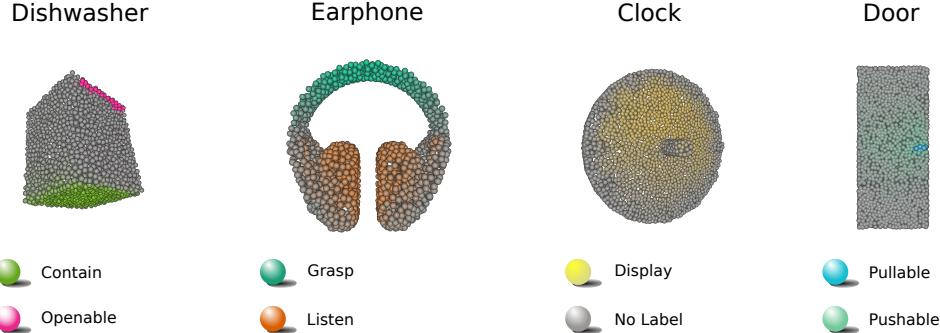


Figure 4: Examples for Affordance Prediction using the prototypical model with PointNet++ backbone. The point color indicates the affordance label and the predicted probability.

segmentation backbone models without prototype modules and losses. The depth of the feature map produced by the segmentation backbone model is set to $D = 128$, which we identified as a sufficiently high dimension. In the evaluation of the performance of the networks, we set the number of prototypes per class to $P_a = 3$ because we obtained the best results with this setting. We consider a wider range of prototypes per class in the ablation studies.

The prototypical network and the baselines are implemented with Pytorch. Our implementation builds upon the public code of the 3D-AffordanceNet benchmark [14] as published on GitHub¹ and relies on their implementation of the data loader and evaluation metrics.

4.3 Training Procedure

All models were trained for 25 epochs because we observed no further improvement for any of the models after that point. We used a batch size of 96 to optimally use the 48 GB VRAM of the single RTX 6000 GPU we used for training. The model parameters were optimized using AdamW [52] with a learning rate of 0.001 and weight decay of 1e-8, which proved to bring the best results. The training time per model was around 1.5 hours.

4.4 Evaluation Metrics

The experiments use the evaluation metrics proposed by the 3D-AffordanceNet benchmark [14], namely mean Intersection over Union (mIoU), mean Average Precision (mAP), mean Area Under the Curve (mAUC) and Mean Squared Error (MSE). For MSE, lower values indicate better performance, as there is less spatial difference between the ground truth and predicted affordances. For mIoU, mAP, and mAUC, higher values reflect a greater accuracy and precision in affordance detection, thus indicating better performance.

5 Experimental Results

Table 1: Performance comparison of PointNet++ and a prototypical model with a PointNet++ backbone on the 3D AffordanceNet validation dataset.

Affordance Metric	PointNet++	Prototypical Model
mIoU \uparrow	18.6	21.5
mAP \uparrow	46.5	50.9
mAUC \uparrow	79.9	83.1
MSE \downarrow	0.02	0.02

Table 1 presents the performance metrics on the 3D-AffordanceNet [14] validation dataset averaged over all affordance classes. The model extended by prototypes outperforms the baseline implementations in every metric. Most notably, with PointNet++ as the Segmentation backbone, the prototypical model achieves 4.4% higher mAP and 2.9% higher mIoU.

¹<https://github.com/Gorilla-Lab-SCUT/AffordanceNet> - CommitID d959a52

Table 2: Validation metrics per affordance class for the PointNet++ baseline and our model with PointNet++ as backbone.

Affordances	PointNet++				Prototypical Model with PointNet++				Change in mIoU
	mIoU ↑	mAP ↑	mAUC ↑	MSE ↓	mIoU ↑	mAP ↑	mAUC ↑	MSE ↓	
Grasp	20.2	49.6	75.8	0.012	23.4	54.5	78.2	0.011	+3.24
Contain	13.4	44.3	77.1	0.024	13.6	49.4	79.3	0.023	+0.18
Lift	0.2	19.6	70.3	0.000	5.06	34.5	86.8	0.000	+4.86
Openable	8.5	34.0	84.3	0.008	12.1	41.0	88.1	0.007	+3.58
Layable	3.5	31.0	74.0	0.001	15.0	47.6	82.4	0.001	+11.55
Sittable	37.7	74.0	93.5	0.021	40.5	76.4	94.5	0.019	+2.78
Support	19.6	46.8	88.1	0.050	20.9	48.2	88.9	0.049	+1.29
Wrap grasp	9.2	38.0	70.6	0.012	13.7	42.2	74.7	0.013	+4.48
Pourable	16.6	44.4	79.1	0.008	22.2	51.9	84.5	0.007	+5.63
Move	21.3	54.8	81.1	0.081	23.2	55.8	82.0	0.080	+1.94
Display	39.1	67.6	90.4	0.010	35.1	68.0	90.4	0.007	-3.96
Pushable	1.0	21.1	77.0	0.001	1.12	17.2	77.4	0.001	+0.12
Pull	0.1	6.6	63.0	0.000	0.08	7.83	61.9	0.000	-0.02
Listen	23.2	47.2	78.0	0.001	28.0	58.8	81.8	0.001	+4.83
Wear	2.9	31.9	64.7	0.002	9.45	44.3	71.6	0.001	+6.55
Press	16.1	41.3	85.7	0.001	20.3	42.6	87.9	0.001	+4.19
Cut	30.9	63.4	90.1	0.001	29.6	60.6	90.9	0.001	-1.33
Stab	34.9	83.5	98.8	0.000	37.6	80.9	98.6	0.000	+2.67
No Label	55.2	84.4	76.7	0.181	57.1	85.6	78.2	0.174	+1.93
Average	18.6	46.5	79.9	0.022	21.5	50.9	83.1	0.021	+2.87

Detailed per affordance class validation metrics for the PointNet++ backbone experiment are presented in [Table 2](#). When using prototypes, the performance increases for nearly every affordance class in every metric, except for *display* (-3.96% mIoU), *cut* (-1.33% mIoU), and *pull* (-0.02% mIoU). XPointNet shows the biggest improvements for *layable* (+11.55% mIoU), *wear* (+6.55% mIoU) and *pourable* (+5.63% mIoU). [Figure 4](#) illustrates some example affordance predictions for the model using prototypes.

5.1 Interpretability Analysis

The usage of prototypes in point cloud-processing models is a further step in the development of inherently interpretable models and follows the basic prototypical framework set by [\[19\]](#) for ProtoPNet, which provides inherent interpretability for image classification in the pattern of “this looks like that”. [Figure 5](#) shows some examples in the adapted pattern “this <object> can <afford> like <training objects>”. For each point cloud, the colors indicate the activation pattern for the most strongly activated prototype for the predicted affordance.

For the first affordance example, *contain*, the prototype activation patterns wind around the object. This corresponds to a structure a human observer would expect to be present in an object designated to contain something: an enclosed void with an opening to put something into the hollow object.

For the affordance *sittable*, the prototypes are mainly active on flat surfaces in chair-like shapes. Notably, even though the table objects have a flat surface, they are not among the most similar objects for this prototype. This indicates that the backbone successfully encodes shape information about the entire object, not just the local geometry.

The depicted prototype for the affordance *layable* demonstrates the need for learning multiple prototypes per class. This prototype is highly activated by one table leg on each object. This, by itself, would not be sufficient to successfully predict the affordance *layable*. However, in conjunction with other prototypes, this prototype encodes enough information about the object shape to detect the affordance correctly.

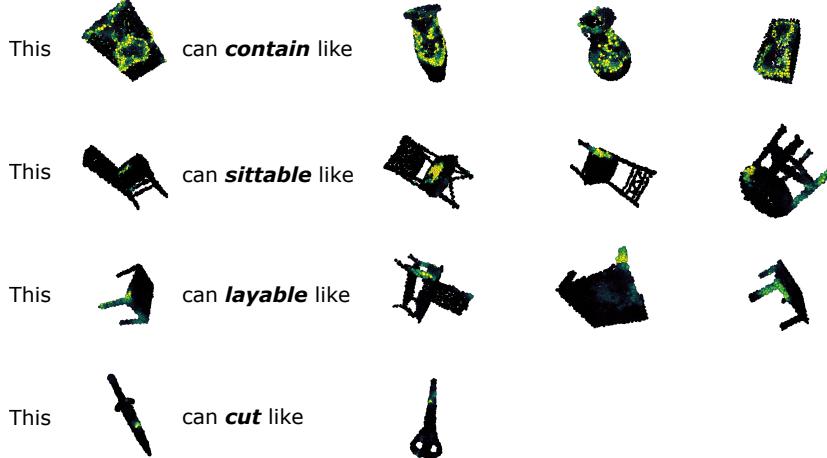


Figure 5: Prototypes provide insights into a model’s reasoning by highlighting point cloud segments with high activations for new inputs and providing similar activation regions for known samples from the training data.

Lastly, the affordance *cut* shows a possible shortcoming of such explanations. Although the prediction and prototype are correct, the prototype activations mainly focus on a single point of the point cloud and not along the entire edge of the object, as a human would expect. The thinness of the object could cause this effect, as the backbone network might encode the point cloud’s border too differently than the blade’s center. Another explanation could be that the prototypes are not yet fully optimized. Such “debugging” information is another valuable insight that prototypical models offer ML practitioners aiming to solve downstream tasks, which is impossible with pure black-box models.

5.2 Ablation: Number of Prototypes

The number of prototypes corresponds to potential clusters in the latent space [42]. Choosing the number of prototypes is, therefore, a crucial hyperparameter for the prototypical model’s performance. To analyze the behavior of the network in the context of different numbers of prototypes, we conduct an ablation over the number of prototypes.

Table 3: Affordance Detection metrics for the prototypical model with varying numbers of prototypes per class.

Metrics	Number of Prototypes Per Class			
	1	3	5	10
mIoU	20.4	21.5	18.0	17.3
mAP	49.3	50.9	45.4	44.6
mAUC	82.7	83.1	79.6	78.6
MSE ↓	0.030	0.020	0.040	0.040

Table 3 summarizes the evaluation metrics for the prototypical model with different numbers of prototypes per class. The network with three prototypes per affordance performs overall the best on the chosen metrics. With an increase in the number of prototypes, the predictive performance of the network decreases. This could indicate that the increase in clusters overly fragments the latent space. When using the one prototype per affordance, the model shows a slower increase of the predictive power throughout the training and reaches a worse performance than the network with three prototypes per affordance. However, the evaluation scores are close to the scores of the model with three prototypes per class, which warrants closer inspection.

Table 4 presents the mIoU scores for the three classes with the highest difference between the prototypical model with $P_a = 3$ and the model with $P_a = 1$. A single prototype per affordance performs well for classes that occur on continuous, relatively flat surfaces, such as *contain*, *sittable*, or *display*. However, this approach is less practical for affordances that depend on broader contextual information. For example, the affordance *listen* is primarily associated with headphones and may require more awareness of a surface’s context. Similarly, affordances like *lift* or *press* exhibit

Table 4: Main mIoU differences for the ablation study over the number of prototypes per affordance class (P_a).

Affordance Class	$P_a = 3$	$P_a = 1$	Difference
Press	20.3	13.4	+6.9
Lift	5.1	0.2	+4.9
Listen	28.0	25.1	+2.9
Display	35.1	38.8	-3.7
Sittable	40.5	44.4	-3.9
Contain	13.6	19.9	-6.3

more significant shape variability, likely forming multiple clusters in latent space. These more complex affordances benefit from multiple learned prototypes per class to accurately reflect the various clusters.

5.3 Ablation: Backbones

To better understand the influence of the prototypical layer on the task of affordance detection rather than a specific model, we evaluate the performance of additional backbones. We choose DGCNN[16], following 3DAffordanceNet[14], as well as PointTransformerV3 [51] as representation for transformer-based models. Because DGCNN is slower to converge, the baseline DGCNN model and prototypical model with DGCNN backbone were trained for 250 epochs.

Table 5 lists the evaluation metrics for the additional baselines and the corresponding prototypical models. All four models perform worse than their PointNet++ counterparts. However, this experiment demonstrates the feasibility of extending different point cloud processing networks with a prototype layer with comparable performance and added advantage of inherent interpretability.

Table 5: Affordance Detection metrics for the prototypical model with DGCNN [16] and PointTransformerV3 (PTv3) [51] backbones.

Metrics	Direction	DGCNN		PointTransformerV3	
		Baseline	Proto. Model	Baseline	Proto. Model
mIoU	↑	16.4	18.4	14.1	12.8
mAP	↑	44.5	44.6	38.3	35.4
mAUC	↑	79.9	80.0	84.2	81.1
MSE	↓	0.050	0.050	0.063	0.066

5.4 Multi-Label Prediction

The 3D-AffordanceNet Benchmark provides multiple affordance labels per point, enabling multi-label prediction instead of multi-classification. We tested the prototypical model and PointNet++ in this setting. Following [14, 36], both models use a class-specific classification head (CSC) with independent binary classification and sigmoid activation for each class. Since points can have multiple labels, affordance-specific prototypes are not possible, so prototypes are potentially shared across affordance classes.

Table 6 shows that in this more difficult setting, performance scores for both models are lower than in the previous experiment, but the prototype model’s advantage over PointNet++ is more pronounced (3% for mIoU and nearly 6% for mAP). However, interpretation becomes more difficult with shared prototypes. **Figure 6** shows prototype activation on a drawer/cupboard, with high activation on the object’s exterior. This could indicate response to the enclosed space, similar to *contain* prototypes in **Figure 5**, or identification of the *openable* affordance of the front doors.

Metrics	Multi-Label Experiment	
	PointNet++	Prototypical Model
mIoU	14.9	17.9
mAP	41.8	47.1
mAUC	84.8	86.8
MSE ↓	0.040	0.030

Table 6: Affordance Detection metrics of PointNet++ and the prototypical model with PointNet++ backbone on the 3D AffordanceNet validation dataset with multi-label annotations.



Figure 6: Example prototype activation with non-affordance-specific prototypes for multi-label prediction.

6 Conclusion

This work introduces prototypical learning to point cloud processing. Specifically, we extended existing point-based models for affordance detection by adding a prototype layer as defined in [22], signifying the adaptability of prototypes to different settings and data types. The resulting prototypical networks achieve state-of-the-art performance on the 3D AffordanceNet benchmark while providing inherent interpretability through the learned prototypes.

Future research could enhance interpretability by learning shaped prototypes with Cartesian XYZ coordinates similar to the approach in [20]. This would allow the prototypes to capture entire multi-point segments of the point clouds, which could be visualized. However, ensuring predictive performance with such a mechanism would remain challenging.

The prototypical models’ combination of inherent interpretability and high affordance detection performance makes them a prime candidate for embodied intelligent agents, particularly in scenarios requiring increased model trust and safety levels, such as human-robot interaction.

References

- [1] M. Wang, R. Luo, A. O. Önal, and T. Padir, “Affordance-based mobile robot navigation among movable obstacles,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2734–2740, 2020. [1](#)
- [2] L. Gregorians and H. J. Spiers, *Affordances for Spatial Navigation*, pp. 99–112. Cham: Springer International Publishing, 2022. [1](#)
- [3] C. Yin and Q. Zhang, “Object affordance detection with boundary-preserving network for robotic manipulation tasks,” *Neural Computing and Applications*, vol. 34, no. 20, pp. 17963–17980, 2022. [1](#)
- [4] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, “Affordances from human videos as a versatile representation for robotics,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13778–13790, June 2023. [1](#)
- [5] J. Lemée, D. Vachtsevanou, S. Mayer, and A. Ciortea, “Signifiers for conveying and exploiting affordances: from human-computer interaction to multi-agent systems,” *Annals of Mathematics and Artificial Intelligence*, vol. 92, no. 4, pp. 815–835, 2024. [1](#)
- [6] J. J. Gibson, *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press, 2014. [1](#)
- [7] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, “RoboCasa: Large-scale simulation of everyday tasks for generalist robots,” in *Robotics: Science and Systems (RSS)*, 2024. [1](#)
- [8] J. Aleotti, V. Micelli, and S. Caselli, “An affordance sensitive system for robot to human object handover,” *International Journal of Social Robotics*, vol. 6, no. 4, pp. 653–666, 2014. [1](#)
- [9] P. Ardón, M. E. Cabrera, E. Pairet, R. P. A. Petrick, S. Ramamoorthy, K. S. Lohan, and M. Cakmak, “Affordance-aware handovers with human arm mobility constraints,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3136–3143, 2021. [1](#)
- [10] J. Jian, X. Liu, M. Li, R. Hu, and J. Liu, “Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14713–14724, Oct. 2023. [1](#)
- [11] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, “Object-based affordances detection with convolutional neural networks and dense conditional random fields,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5908–5915, 2017. [1](#)

- [12] S. Thermos, G. Potamianos, and P. Daras, “Joint object affordance reasoning and segmentation in rgb-d videos,” *IEEE Access*, vol. 9, pp. 89699–89713, 2021. [1](#)
- [13] Z. Khalifa and S. A. A. Shah, “A large scale multi-view rgbd visual affordance learning dataset,” in *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 1325–1329, 2023. [1](#)
- [14] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, “3d affordancenet: A benchmark for visual object affordance understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1778–1787, June 2021. [1, 2, 4, 5, 8](#)
- [15] 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* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017. [1, 2, 3, 4](#)
- [16] 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 (TOG)*, 2019. [1, 2, 3, 4, 8](#)
- [17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. [1](#)
- [18] J. Tayyub, M. Sarmad, and N. Schönborn, “Explaining deep neural networks for point clouds using gradient-based visualisations,” in *Computer Vision – ACCV 2022* (L. Wang, J. Gall, T.-J. Chin, I. Sato, and R. Chellappa, eds.), (Cham), pp. 155–170, Springer Nature Switzerland, 2023. [1](#)
- [19] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, “This looks like that: Deep learning for interpretable image recognition,” in *Advances in Neural Information Processing Systems*, 2019. [2, 4, 6](#)
- [20] J. Donnelly, A. J. Barnett, and C. Chen, “Deformable protopnet: An interpretable image classifier using deformable prototypes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10265–10275, June 2022. [2, 9](#)
- [21] M. Sacha, D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński, “Protoseg: Interpretable semantic segmentation with prototypical parts,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1481–1492, Jan. 2023. [2](#)
- [22] M. X. Li, K. F. Rudolf, N. Blank, and R. Lioutikov, “An overview of prototype formulations for interpretable deep learning,” 2025. [2, 3, 4, 9](#)
- [23] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, “Detecting object affordances with convolutional neural networks,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2765–2770, 2016. [2](#)
- [24] T.-T. Do, A. Nguyen, and I. Reid, “Affordancenet: An end-to-end deep learning approach for object affordance detection,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5882–5889, 2018. [2](#)
- [25] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler, “Learning to act properly: Predicting and explaining affordances from images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [2](#)
- [26] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, “Learning affordance grounding from exocentric images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2252–2261, June 2022. [2](#)
- [27] X. Chen, T. Liu, H. Zhao, G. Zhou, and Y.-Q. Zhang, “Cerberus transformer: Joint semantic, affordance and attribute parsing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19649–19658, June 2022. [2](#)
- [28] J. Chen, D. Gao, K. Q. Lin, and M. Z. Shou, “Affordance grounding from demonstration video to target image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6799–6808, June 2023. [2](#)
- [29] S. A. A. Shah and Z. Khalifa, “Hierarchical transformer for visual affordance understanding using a large-scale dataset,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11371–11376, 2023. [2](#)
- [30] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani, “One-shot open affordance learning with foundation models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3086–3096, June 2024. [2](#)
- [31] A. Rai, K. Buettner, and A. Kovashka, “Strategies to leverage foundational model knowledge in object affordance grounding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1714–1723, June 2024. [2](#)
- [32] D. I. Kim and G. S. Sukhatme, “Semantic labeling of 3d point clouds with object affordance for robot manipulation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5578–5584, 2014. [2](#)
- [33] X. Li, S. Liu, K. Kim, X. Wang, M.-H. Yang, and J. Kautz, “Putting humans in a scene: Learning affordance in 3d indoor environments,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [34] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, “Affordance detection of tool parts from geometric features,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1374–1381, 2015. [2](#)

- [35] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, “Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [36] R. A. Tabib, D. Hegde, and U. Mudenagudi, “Lgafford-net: A local geometry aware affordance detection network for 3d point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 5261–5270, June 2024. [2](#), [4](#), [8](#)
- [37] T. Nguyen, M. N. Vu, A. Vuong, D. Nguyen, T. Vo, N. Le, and A. Nguyen, “Open-vocabulary affordance detection in 3d point clouds,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5692–5698, 2023. [2](#)
- [38] X.-F. Han, Y.-F. Jin, H.-X. Cheng, and G.-Q. Xiao, “Dual transformer for point cloud analysis,” *IEEE Transactions on Multimedia*, vol. 25, pp. 5638–5648, 2023. [2](#)
- [39] O. Li, H. Liu, C. Chen, and C. Rudin, “Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI’18/IAAI’18/EAAL’18*, (New Orleans, Louisiana, USA), AAAI Press, 2018. [2](#)
- [40] D. Rymarczyk, Ł. Struski, M. Górszczak, K. Lewandowska, J. Tabor, and B. Zieliński, “Interpretable image classification with differentiable prototypes assignment,” in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 351–368, Springer Nature Switzerland, 2022. [2](#)
- [41] Y. Ukai, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, “This looks like it rather than that: ProtoKNN for similarity-based classifiers,” in *The Eleventh International Conference on Learning Representations*, 2023. [2](#)
- [42] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, “Rethinking semantic segmentation: A prototype view,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2582–2593, June 2022. [2](#), [3](#), [7](#)
- [43] N. Moradinasab, L. S. Shankman, R. A. Deaton, G. K. Owens, and D. E. Brown, “Protogmm: Multi-prototype gaussian-mixture-based domain adaptation model for semantic segmentation,” 2024. [2](#)
- [44] P. Tang, H.-M. Xu, and C. Ma, “ProtoTransfer: Cross-modal prototype transfer for point cloud segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3337–3347, 2023. [2](#)
- [45] Y. Zhao, J. Wang, X. Li, Y. Hu, C. Zhang, Y. Wang, and S. Chen, “Number-adaptive prototype learning for 3D point cloud semantic segmentation,” in *Computer Vision – ECCV 2022 Workshops* (L. Karlinsky, T. Michaeli, and K. Nishino, eds.), (Cham), pp. 695–703, Springer Nature Switzerland, 2023. [2](#)
- [46] N. Zhao, T.-S. Chua, and G. H. Lee, “Few-shot 3d point cloud semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8873–8882, June 2021. [2](#)
- [47] S. He, X. Jiang, W. Jiang, and H. Ding, “Prototype adaption and projection for few- and zero-shot 3d point cloud semantic segmentation,” *IEEE Transactions on Image Processing*, vol. 32, pp. 3199–3211, 2023. [2](#)
- [48] S. Zhao and X. Qi, “Prototypical votenet for few-shot 3d point cloud object detection,” in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 13838–13851, Curran Associates, Inc., 2022. [2](#)
- [49] Y. Su, X. Xu, and K. Jia, “Weakly supervised 3d point cloud segmentation via multi-prototype learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7723–7736, 2023. [2](#)
- [50] R. Royen, L. Denis, and A. Munteanu, “Protoseg: A prototype-based point cloud instance segmentation method,” 2024. [2](#)
- [51] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, “Point transformer v3: Simpler, faster, stronger,” in *CVPR*, 2024. [3](#), [4](#), [8](#)
- [52] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019. [5](#)