PointNet

A Preprint

Authors present framework that allows to use unordered set of points directly.

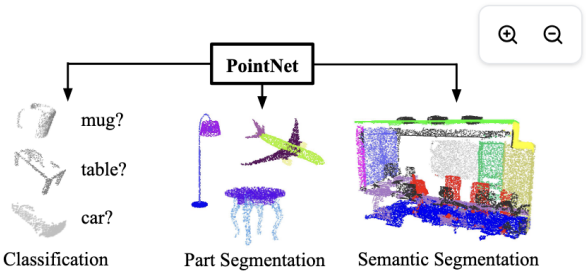

Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

Authors only use the (x, y, z) coordinate as our point's channels. Proposed deep network outputs k scores for all the k candidate classes. For semantic segmentation, the input can be a single object for part region segmentation, or a sub-volume from a 3D scene for object region segmentation.

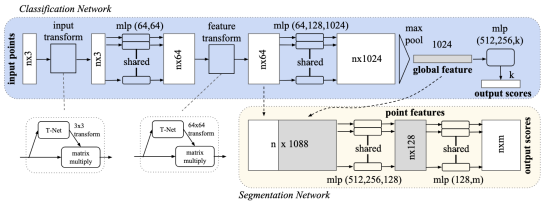

Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

The network consists of mainly 3 parts: 2 aggregating alignment networks and symetric max pooling. It is also important to make points invariant, so 3 strategies for input to model exists:

1. (I) Sort input into a canonical order
2. (II) Treat the input as a sequence to train an RNN, but augment the training data by all kinds of permutations
3. (III) Use a simple symmetric function to aggregate the information from each point

$$f(\{x_1, x_2, \ldots\}) \approx g(h(x_1), \ldots h(x_n))$$

Authors approximate h by MLP and g by a composition of a single variable function and a max pooling function.

Joint Alignment Network. To ensure invariance to geometric transformations (e.g., rigid transformations), point cloud representations should be aligned to a canonical space before feature extraction. Inspired by spatial transformers in images [?], we propose a simpler solution for point clouds. A small network (T-net) predicts an affine transformation matrix applied directly to the input coordinates, avoiding the need for new layers or issues like aliasing.

This idea extends to the feature space, though the higher dimensionality of the feature transformation matrix complicates optimization. To address this, we introduce a regularization term encouraging the matrix to be close to orthogonal:

$$\mathcal{L}_{\text{reg}} = \left\| I - AA^\top \right\|_F^2,$$

where A is the predicted feature transformation matrix. This regularization stabilizes training and improves model performance.

3D Object Classification. Our network learns global point cloud features for object classification. We evaluate it on a benchmark dataset containing over 12,000 CAD models from 40 object categories. Unlike previous methods relying on voxel grids or multi-view images, our approach directly processes raw point clouds. We uniformly sample 1,024 points per shape and normalize them to a unit sphere. During training, we apply random up-axis rotations and Gaussian jittering (standard deviation 0.02). Our method achieves strong classification accuracy and significantly faster inference, due to its lightweight architecture using only fully connected layers and max pooling. While multi-view image-based methods still show slightly better accuracy, this is likely due to their ability to capture fine geometric details.

3D Object Part Segmentation. We evaluate fine-grained part segmentation on a dataset with over 16,000 shapes across 16 categories, annotated with up to 50 part labels. Most shapes are labeled with 2 to 5 parts. Segmentation is formulated as a per-point classification task, and performance is reported using mean IoU (mIoU). For each shape, we compute IoU per part type and average them; missing predictions are assigned an IoU of 1. Our segmentation network outperforms traditional baselines and a 3D CNN reference model, improving mIoU by 2.3% on average.

To assess robustness, we generate incomplete point clouds using a simulator, creating partial scans from multiple viewpoints. Training our network on both complete and partial data shows only a 5.3% drop in mIoU, demonstrating good generalization. Qualitative results confirm that the model produces reasonable segmentations even under occlusion and missing data.

Semantic Segmentation in Scenes. The same architecture extends naturally to semantic scene segmentation, where each point is labeled with a semantic class (e.g., wall, floor, chair). We use a large-scale indoor dataset of 3D scans, each annotated point belonging to one of 13 semantic categories. Point clouds are divided by room and split into 1m×1m blocks for training. The network successfully generalizes to room-scale scenes and captures semantic structure effectively.

The network also should recognize a chair whether it s upright, upside down, or rotated. This is known as transformation invariance. Instead of forcing the network to learn every possible rotation, PointNet learns to canonicalize the input first. It does this with small, dedicated networks called T-Nets (Transformation Networks). Basically, T-Net is a mini-PointNet that looks at the input points and predicts a small transformation matrix (e.g., a 3x3 matrix for rotation). This predicted matrix is then used to rotate the input point cloud into a standard, aligned orientation before the main feature extraction happens. The network learns the best way to align objects to make its own job easier.