

PointNet: 针对三维分类点云分类与分割的深度学习模型

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas

Stanford University

翻译: 浙江大学软件学院 2021 年夏令营 CAD 第三小组

Abstract

点云是一种重要的几何数据结构。由于其无序性，大多数研究者将这些数据转换为有序的三维体素网格或图像集合。然而，这会导致不必要的数据冗余并引起一些问题。在本文中，我们设计了一种直接处理点云的新型神经网络，这种网络很好的考虑了输入中的点的置换不变性。我们构建了名叫 PointNet 的神经网络，它为物体分类、零件分割和场景语义分析等应用程序提供了统一的结构。PointNet 结构虽然简单，但却十分高效且有效。从经验上，它表现出了与现有技术水平相当甚至更好的性能。从理论上，我们为解释这种网络学习到了什么和为什么这种网络能够对输入数据的扰动和损坏保持高度健壮性进行了分析。

1. 引入

本文探索了能够处理 3D 几何数据比如点云和网格的深度学习模型。为了实现权重分享和其他核函数优化，典型的卷积结构要求输入数据的格式高度规则，比如图形网格和 3D 体素。由于点云或网格并不是有序格式，大多数研究人员通常会将这些数据转换为常规的 3D 体素或图像集合（例如视图），然后再将它们输入到深层网络中。然而，这种数据表示转换会使结果数据变不必要的大量增加，同时也会引入能模糊数据自然不变性的量化伪影。

因此我们重点介绍一种不同于以往只使用简单点云的 3D 几何神经网络，并将其命名为 PointNets。点云是简单且统一的结构，能够避免网格的组合不规则性

和复杂性，因此更容易学习。然而，PointNet 依旧受限于点云只是点的一个集合这样一个事实，因此其内部元素的排列是不变的，需要在网络计算中做一些对称性处理。而更进一步的刚性运动的不变性也需要考虑。

我们构建的 PointNet 是一个统一的架构，这体现在将点云作为输入，输出整体输入或相对于输入的部分点块的类标签。该模型的基础构架非常的简单，因为在初始阶段，每个点的处理方法完全相同且独立。在基础设置中，每个点仅用它的三个坐标 (x, y, z) 来表示。额外维度可以通过计算法线和其他局部或者全局特征来添加得到。

我们这个方法的关键在于使用对称的 max pooling 函数。网络有效地学习一组优化函数/准则，从点云中选择感兴趣的点或者含有信息的点，并编码选择的原因。网络最终的全连接层把这些学到的最优值聚合到如上所述的整个形状的全局描述符（形状分类）上或者用于预测每个点标签（形状分割）。

我们的输入格式很容易应用刚性或者仿射变换，因为每个点都是独立变换的。因此我们可以在应用 PointNet 处理点云之前添加一个与数据无关的空间转换网络来规范化点云数据，以此来进一步提高结果的表现。

我们提供了方法的理论分析和实验评估，展示了我们的网路可以近似任何连续的集合函数。更有趣的是，事实证明我们的网络通过学习一组稀疏的关键点来概括一个输入点云，这些关键点根据可视化结果在视觉上大致对应对象的骨架。理论分析解释了为什么我们的 PointNet 对于输入点云的微小扰动，以及由极端点插入或删除点带来的数据损坏有着良好的健壮性。

我们对大量参照数据集（包括形状分类、零件分割、场景分割）进行实验，比较了 PointNet 和目前基于

* 表示同等贡献。

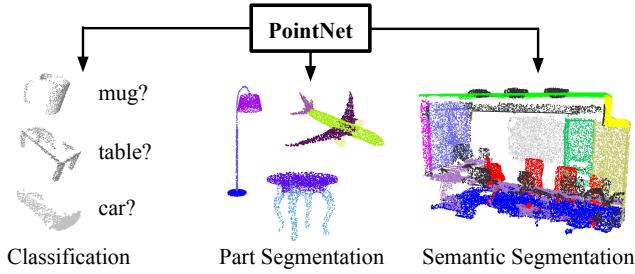


图 1. PointNet 的应用 我们提出了一个新的神经网络结构，其输入为一个原始的点云（点的集合），没有经过体素化或预渲染。该网络具有一个统一的架构，能够学习全局和局部的点特征，从而为各种三维识别任务提供了一个简单、高速而有效的方法。

多视角与体积表示的最先进的方法。在统一的构架下，PointNet 不仅在速度上更快，同时也表现出了和现有技术相当甚至更好的性能。我们工作的主要贡献如下：

- 我们设计了一个适用于处理 3D 无序点集的新型深度网络模型
- 我们展示了如何训练这样的网络模型来处理 3D 形状分类、零件分割和场景语义分析等任务
- 我们从实验和理论的角度分析了方法的稳定性和效率
- 我们演示了网络中所选择神经元计算出的 3D 特征，并对其性能进行了直观的解释

使用神经网络处理无序集合的问题是一个非常普遍和根本的问题，我们希望我们的想法也可以应用到其他领域。

2. 相关工作

点云特征 点云大部分的现有特征都是针对特定任务手工制作的。点特征通常编码某些统计特性，并且被设计成对于某些变换是不变的，这些变换通常被划分为内在的 [2, 24, 3] 或外在的 [20, 19, 14, 10, 5]。它们还可以归类为局部特征和全局特征。对于特定任务，找到最优的特征组合并非易事。

3D 数据的深度学习 3D 数据具有多种流行的表示形式，从而有各种学习方法。Volumetric CNNs: [28, 17, 18] 这是在体素形状上应用 3D 卷积神经网络的先驱。然而，

由于数据的稀疏性和 3D 卷积的计算成本，体积表示受到其分辨率的限制。FPNN [13] 和 Vote3D [26] 提出了处理稀疏性问题的特殊方法；然而，他们的操作仍然是在稀疏的体积上，处理非常大的点云对他们来说是一种挑战。Multiview CNNs: [23, 18] 试图将 3D 点云或形状渲染成 2D 图像，然后应用 2D 卷积神经网络对它们进行分类。通过精心设计的图像 CNN，这一系列方法在形状分类和检索任务方面取得了突出的性能表现。[21]。然而，将它们扩展到场景理解或其他 3D 任务例如点分类、形状完成上时表现很普通。Spectral CNNs: 最近的一些工作 [4, 16] 在网格上使用了光谱 CNN。但是这些方法目前受限于流形网格（例如有机物体），而且如何将它们扩展到非等距形状（例如家具）上并不明显。Feature-based DNNs: [6, 8] 首先通过提取传统的形状特征将 3D 数据转换成向量，然后使用全连接网络对形状进行分类。我们认为它们受限于所提取特征的表示能力。

无序集的深度学习 从数据结构的角度来说，点云是无序的向量集合。当大部分深度学习工作集中在规则输入表示，如序列（语音和语言处理）、图像和体积（视频或 3D 数据）上时，很少有在点集上做深度学习工作的。

Oriol Vinyals 等人最近的一项工作 [25] 研究了这个问题。他们使用具有注意机制的读取-处理-写入网络来处理无序输入集，同时展示他们的网络具有对数字进行排序的能力。然而，由于他们的工作重点是泛型集合和 NLP 应用，因此缺少了几何体在集合中的作用。

3. 问题陈述

我们设计了一个能够将无序的点集作为输入的深度学习网络框架。一个点云可以由若干个三维点的集合表示 $\{P_i | i = 1, \dots, n\}$ ，其中每个点 P_i 都是其空间坐标 (x, y, z) 和其他额外的特征频道（例如颜色、范数等）。出于简洁性和清晰性，除非特别指出，我们只使用三维空间坐标，即 (x, y, z) 作为输入点的频道。

对于物品分类的任务，输入点可以直接采样自某个形状，或是某个场景的点云预先分割好的结果。我们提出的深度网络为 k 个候选分类输出 k 个得分。对于语义分割任务，输入可以使一个用于零件部分分割的单个物体，也可以是一个 3D 场景下用于区域零件分隔的子

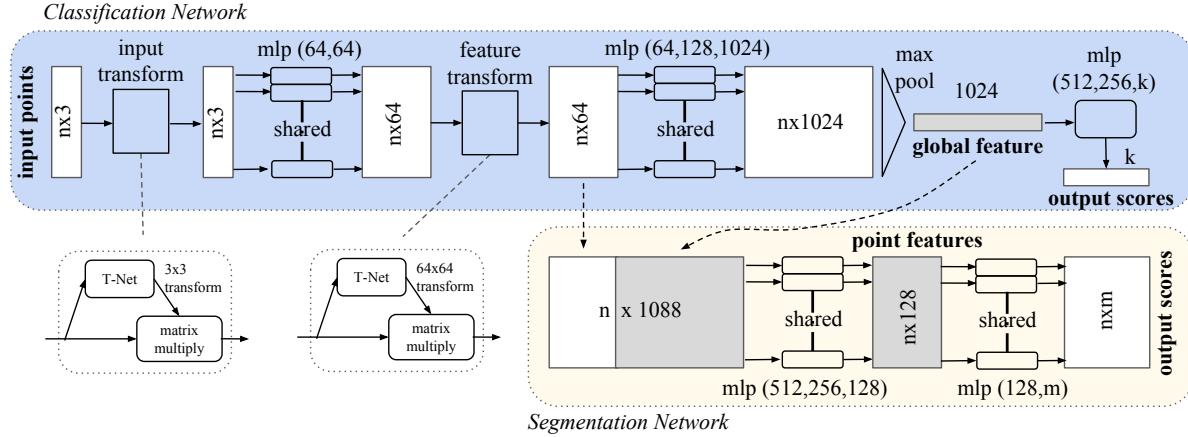


图2. PointNet 架构分类网络将 n 个点作为输入，作用输入和特征变换，然后通过 max pooling 聚合点的特征，输出的是 k 种分类的分类分数。分割网络是分类网络的拓展，它连接局部和全局特征以及每个点输出的分数。“mlp” 代表多层感知机，括号中的数字是层的大小。Batchnorm 用于所有应用了 ReLU 的层。Dropout 用于分类网络中的最后一个多层次感知机。

空间。我们的模型将会为 n 个点与 m 个语义子分类输出 $n \times m$ 个得分。

4. 点集上的深度学习

我们提出的 PointNet 网络结构 (4.2节) 的灵感主要来源于4.1节所述的 \mathbb{R}^n 内点集的属性。

4.1. \mathbb{R}^n 内点集的属性

网络的输入是来自欧几里得空间的点的子集，主要有以下三种主要的性质：

- 无序性。不同于图片的像素或者立方体的体素，点云是一些无特定顺序的点的集合。换句话说，一个要处理 N 个 3D 点集的网络需要对按序输入的点集的 $N!$ 种排序保持不变。
- 点之间有相互作用。这些点来自具有距离度量的空间。这意味着这些点不是孤立的，而且相邻的点形成一个具有意义的子集。因此，模型需要能够从邻近点中捕获局部结构，以及局部结构之间的组合相互作用。
- 转换的不变性。作为一个几何体，已习得的点集表示需要对一些变换保持不变。例如，对一些点一起做旋转和平移变换既不会改变全局的点云类别、也不会改变点的分割。

4.2. PointNet 的结构

我们的完整网络架构图如图2所示，其中用于分类的网络和用于分割的网络占据了网络结构中的很大一部分。请阅读图2的题注以了解流程。

我们的网络有三个关键模块：用作对称函数、聚类所有点的信息的最大池化层；一个局部和全局信息结合的结构；和两个对齐了输入点和点的特征的联合对齐网络。

我们将在以下的单独段落中讨论这些设计选择背后的原因。

用于处理无序输入的对称函数 为了使模型对输入排序不变，存在三种策略：1) 将输入排列为规范顺序；2) 将输入看作训练 RNN 的序列，但通过各种排列增加了训练数据；3) 用一个简单的对称函数来聚类每个点的信息。这里，对称函数将 n 个向量作为输入，然后输出一个不受输入顺序变化影响的新向量。例如， $+$ 和 $*$ 操作符是对称二元函数。

尽管排序听起来像是一个简单的解决方法，但实际上谈到一般意义上的点的扰动时，在高维空间中并不存在稳定的排序。这可以很容易地通过矛盾来显示。如果这样一种排序策略存在，那么它定义了高维空间和一维实线之间的双向映射。不难看出，谈及点扰动时要求排序的稳定等同在维度降低时保持映射在空间上的接近度，一般情况下这项任务是无法实现的。因此，排序并不能完全解决排序问题，而且因为排序问题的存

在，网络很难学习到从输入到输出的一致映射。如实验（图5）中所示，我们发现直接在排序点集上应用多层感知机的性能不好，尽管略优于直接处理无序输入。

使用 RNN 的方法是将点集作为顺序信号，并希望用随机排序的序列训练 RNN，这样 RNN 将会变得和输入顺序无关。然而，在“OrderMatters” [25] 中，作者已经证明顺序确实很重要而且无法完全被忽略。尽管 RNN 对于较短（几十个）的序列的输入排序已经具有相对良好的鲁棒性，但是很难扩展到像点集这样有成千上万的输入元素上。根据实验，我们也证明了基于 RNN 的模型在性能上不如我们所提出的方法（图5）。

我们的想法是通过对集合中转换元素应用对称函数来近似在点集上定义的一般函数：

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n)), \quad (1)$$

其中 $f : 2^{\mathbb{R}^N} \rightarrow \mathbb{R}$, $h : \mathbb{R}^N \rightarrow \mathbb{R}^K$ 和 $g : \underbrace{\mathbb{R}^K \times \dots \times \mathbb{R}^K}_n \rightarrow \mathbb{R}$ 是对称函数。

根据实验，我们的基础模块十分简单：我们通过多层感知机来近似 h ，并通过单个变量函数和最大池化函数的组合来近似 g 。通过实验发现这样的模块结构表现的很好。通过 h 的集合，我们可以学到一些 f 来捕获点集的不同属性。

尽管我们的关键模块看起来很简单，但它也具有很精彩的地方（参见5.3节），在一些不同的应用中能够表现出很强的性能（参见5.1节）。由于我们模块的简单性，我们也提供了4.3节中的理论分析。

局部和全局信息聚类 上一节的输出组成了一个向量 $[f_1, \dots, f_K]$ ，这是输入集的全局标签。我们可以在形同全局特征上很容易训练 SVM 或多层次感知机分类器以进行分类。然而，点的分割需要结合局部信息和全局信息。我们能够通过简单而高效的方式来实现这一目标。

我们的解决方法可以在图2 (*Segmentation Network*) 中看到。在计算完全全局点云特征向量后，我们将全局特征和每一个点的特征连接起来反馈给每一个点特征。然后我们基于组合的点特征提取新的每个点的特征，这样每个点特征都考虑了局部和全局信息。

通过这样的修改，我们的网络能够预测依赖于局部几何和全局语义的每个点数量。例如我们可以准确地预测出每个点的法线（图中的补充），验证网络能够汇总来自该点的局部领域的信息。在实验环节中，我们也

表明我们的模型可以在形状部分分割和场景分割方面实现最先进的性能。

联合对齐网络 如果点云在经历了某些几何变换（比如刚性变换），那么点云的语义标签必须是不变的。因此我们希望已习得的点集表示不受这些变换影响。

一个自然的解决方法是在特征提取前将所有的输入集对齐到规范空间。Jaderberg 等人 [9] 介绍了一种通过采样和插值来对齐二维图像的空间变换的方法，通过在 GPU 上特别定制的层来实现。

与 [9] 相比，我们的点云输入形式使我们能够以更简单的方式来实现这一目标。我们不需要发明任何新的层，也不需要像图像任务那样引入任何别名。我们通过一个小型网络（图 2 中的 T-net）预测仿射变换矩阵，并直接将该变换作用于输入点的坐标。这一小型网络本身类似于大型网络，由独立的点特征提取、最大池化和全连接层组成。更多关于 T-net 的细节将在附录中介绍。

这个想法可以进一步扩展到特征空间的对齐。我们可以在点的特征上插入另一个对齐网络，并预测特征变换矩阵以对齐来自不同输入点云的特征。然而，特征空间中的变换矩阵比空间变换矩阵的维度高很多，这极大地增加了优化的难度。因此，我们在 softmax 的训练损失中增加了一个正则化项。我们约束特征变换矩阵接近于正交矩阵：

$$L_{reg} = \|I - AA^T\|_F^2, \quad (2)$$

其中 A 是小型网络预测得到的特征对齐矩阵。正如我们所期待的，一个正交变换将不会损失任何输入的信息。我们发现通过增加正则化项，优化变得更加稳定，而且我们的模型实现了更好的性能。

4.3. 理论分析

通用近似性 我们首先展示我们的神经网络具备通用近似到连续的集合函数上的能力。由于集函数的连续性，直观上我们认为，一个输入点集上的微小扰动不应当引起函数值的剧烈变化，例如其分类任务或分割任务的表现分数。

形式化地，令 $\mathcal{X} = \{S : S \subseteq [0, 1]^m \text{ and } |S| = n\}$, $f : \mathcal{X} \rightarrow \mathbb{R}$ 在 \mathcal{X} 上关于赫斯多夫距离 $d_H(\cdot, \cdot)$ (即: $\forall \epsilon > 0, \exists \delta > 0$, for any $S, S' \in \mathcal{X}$, if $d_H(S, S') < \delta$, 则

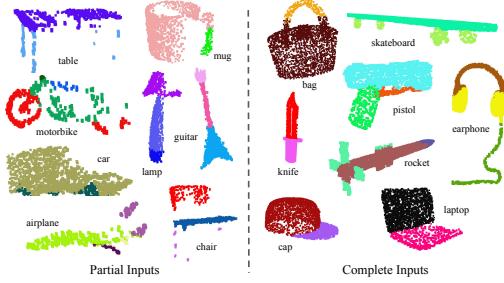


图 3. 零件分割的量化结果 我们可视化了横跨 16 个分类的 CAD 零件的分割结构。我们分别展示了部分模拟的 Kinect 扫描结果（左）和完全的 ShapeNet CAD 模型的结果（右）。

有 $|f(S) - f(S')| < \epsilon$ 的连续的集函数。我们的理论认为在池化层给定足够多的神经元， f 能被无限近似，即，在 (1) 中的 K 已经足够大。

Theorem 1. 令 $f : \mathcal{X} \rightarrow \mathbb{R}$ 是一个关于赫斯多夫距离 $d_H(\cdot, \cdot)$ 的集函数。 $\forall \epsilon > 0$, \exists 一个连续函数 h 和一个对称函数 $g(x_1, \dots, x_n) = \gamma \circ \text{MAX}$, 使得任意 $S \in \mathcal{X}$,

$$\left| f(S) - \gamma \left(\text{MAX}_{x_i \in S} \{h(x_i)\} \right) \right| < \epsilon$$

其中 x_1, \dots, x_n 是一系列在 S 中任意顺序排列的元素， γ 是一个连续函数，且 MAX 是一个取 n 个向量最为输入，然后返回一个元素层面上的最大值的最大化向量运算符。

该定理的理论证明参见本文的附加材料。其主要思想是，在最差情况下，通过将空间划分成等体积的体素，网络的学习结果是到将点云转换为体积表示。而在实际情况下，网络会学习一个聪明得多的策略来探测空间，正如我们将在点函数的可视化中呈现的那样。

瓶颈维度与稳定性 从理论上和实践上，我们发现我们网络的表达性收到池化层维度的强烈影响。即，(1) 中的 K 。在此我们提供了一个分析，同时也揭示了与我们模型的稳定性相关的一些属性。

我们定义 $\mathbf{u} = \text{MAX}_{x_i \in S} \{h(x_i)\}$ 为网络 f 的子网络。其中 f 将分布在 $[0, 1]^m$ 中的点集映射到了 K 维空间的向量。接下来的理论分析指出，输入中的微小扰动或额外的噪音点不会严重影响我们网络的输出结果。

Theorem 2. 令 $\mathbf{u} : \mathcal{X} \rightarrow \mathbb{R}^K$ 使得 $\mathbf{u} = \text{MAX}_{x_i \in S} \{h(x_i)\}$ 且 $f = \gamma \circ \mathbf{u}$. 则，

	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [28]	volume	1	77.3	84.7
VoxNet [17]	volume	12	83.0	85.9
Subvolume [18]	volume	20	86.0	89.2
LFD [28]	image	10	75.5	-
MVCNN [23]	image	80	90.1	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	89.2

表 1. ModelNet40 上的分类结果 我们的网络在一系列 3D 输入的深度网络中达到最先进的效果。

- (a) $\forall S, \exists \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X}, f(T) = f(S)$ if $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$;
- (b) $|\mathcal{C}_S| \leq K$

关于上述定理，我们进一步阐释其推论。(a) 说明 $f(S)$ 在针对输入干扰时保持不变，只要 \mathcal{C}_S 中的点被保留。同时也不会受到在 \mathcal{N}_S 以内的额外噪声点的影响。(b) 说明 \mathcal{C}_S 只包含一些有限数量的点，取决于 (1) 中的 K 。换句话说， $f(S)$ 事实上是完全由一个元素数量小于或等于 K 的有限子集 $\mathcal{C}_S \subseteq S$ 决定的。因此，我们将 \mathcal{C}_S 称为 S 的关键点集，将 K 称为 f 的瓶颈维度。

结合 h 的连续性，我们的模型关于点的扰动、缺失或额外的噪声点的健壮性得以解释。其健壮性的提升类似于机器学习模型中的稀疏原则。**直观上看，我们的模型能够通过一个稀疏的关键点来概括出形状**。在实验部分，我们可以看到从一个物体骨架的关键点。

5. 实验

实验被分成了四部分。首先，我们展示了 PointNets 可以应用于多个 3D 识别任务 (Sec 5.1)。其次，我们提供了详细的实验来验证我们的网络设计 (Sec 5.2)。最后，我们可视化了网络学到的内容 (Sec 5.3) 并且分析了时间和空间复杂度 (Sec 5.4)。

5.1. 应用

在本节中，我们将展示我们的网络是如何经过训练，来完成 3D 目标分类、目标零件分割和场景语义分割任务¹ 即使我们基于全新的数据表示（点集）进行研

¹ 更多应用样例，例如形状对应和基于点云的 CAD 模型检索包含在补充资料中。

	mean	aero	bag	cap	car	chair	ear	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table	board
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271	
Wu [27]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8	
Yi [29]	81.4	81.0	78.4	77.7	75.7	87.6	61.9	92.0	85.4	82.5	95.7	70.6	91.9	85.9	53.1	69.8	75.3	
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1	
Ours	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6	

表 2. ShapeNet 数据集上的分割结果评价指标是点集的 mIoU(%)。我们与两种传统方法 [27] [27] 以及我们提出的三维卷积网络对比。PointNet 方法在 mIoU 上达到最佳表现。

究，我们也能够在多个任务的基准测试中获得相当的甚至更好的性能。

3D 目标分类 我们的网络学习了可用于目标分类的全局点云特征。我们在 ModelNet40 [28] 形状分类基准上评估了我们的模型。总共有来自 40 个人造目标类别的 12311 个 CAD 模型，分为用于训练的 9843 个和用于测试的 1468 个。以前的方法主要集中于体积和多视图图像表示，我们提出了第一个直接处理原始点云的方法。

我们根据表面区域在网格面上均匀地采样了 1024 个点并将它们归一化为一个单位球体。在训练过程中，我们还通过沿上轴随机旋转对象并使用均值为 0 和标准差为 0.02 的高斯噪声抖动每个点的位置来动态地增强点云。

在表 1 中，我们将我们的模型与以前的相关工作，以及我们基于从点云中提取的传统特征（点密度、D2、形状轮廓等）MLP 的基准线进行比较。我们的模型在基于 3D 输入（体积和点云）的所有方法中实现了最优的性能。仅仅使用全连接层和最大池化，我们的网络在推理速度上获得了强大的领先优势，并且也可以很容易地在 CPU 中并行化。但我们的方法与基于多视图的方法 (MVCNN [23])，之间仍然存在一点小差距，我们认为这是精细几何细节的丢失导致的，而这些细节可以被渲染图像所捕获。

3D 目标零件分割 零件分割是一项具有挑战性的细粒度 3D 识别任务。给定 3D 扫描或网格模型，任务是为每个点或面分配所属部分类别标签（例如椅子腿、杯柄）。

我们对来自 [29] 的 ShapeNet 部分数据集进行评估，其中包含来自 16 个类别的 16881 个形状，总共标

注了 50 个部分。大多数目标类别都标有两到五个部分。对应的真实标签注释标记在形状上的采样点上。

我们将零件分割制定转换为每点分类问题。评估指标是点上的 mIoU。对于类别 C 的每个形状 S，计算形状的 mIoU：对于类别 C 中的每个部分类型，计算真实和预测之间的 IoU。如果真实点和预测点的并集为空，则将一部分 IoU 计为 1。然后我们对类别 C 中所有部分类型的 IoU 进行平均以获得该形状的 mIoU。为了计算类别的 mIoU，我们取该类别中所有形状的 mIoU 的平均值。

在本节中，我们将我们的分割版 PointNet (图 2 的修改版，分割网络) 和两种均利用了逐点几何特征和形状之间的对应关系的传统方法 [27] 和 [29]，以及我们自有的 3DCNN 基准线进行了比较。有关 3DCNN 的详细修改和网络架构，请参阅补充说明。

在表 2，我们报告了每个类别和平均 IoU(%) 分数。我们观察到平均 IoU 有 2.3% 的提高，并且我们的网络在大多数类别中都优于基准方法。

我们还对模拟 Kinect 扫描的任务进行了实验，来测试这些方法的稳健性。对于 ShapeNet 部分数据集中的

	mean IoU	overall accuracy
Ours baseline	20.12	53.19
Ours PointNet	47.71	78.62

表 3. 场景语义分割结果评价指标是 13 个类（结构和家具加上其他）的平均 IoU 和根据点集计算的分类准确率。

	table	chair	sofa	board	mean
# instance	455	1363	55	137	
Armeni et al. [1]	46.02	16.15	6.78	3.91	18.22
Ours	46.67	33.80	4.76	11.72	24.24

表 4. 场景中 3D 物体检测结果评价指标是 3D 体积块中 IoU 阈值为 0.5 的平均准确率。

每个 CAD 模型，我们使用 BlensorKinectSimulator [7] 从六个随机视点生成不完整的点云。我们根据这些完整形状和部分扫描数据，使用相同的网络架构和训练设置训练我们的 PointNet。结果表明，我们仅损失了 5.3% 的平均 IoU。在图 3 中，我们展示了针对完整数据和部分数据的定性结果。可以看到，虽然仅提供部分数据的任务相当具有挑战性，但我们的预测是合理的。

场景语义分割 我们的零件分割网络可以很容易地扩展到语义场景分割，其中点标签成为语义目标类别，而不是目标的部分标签。

我们在斯坦福 3D 语义解析数据 [1] 上进行了实验。该数据集包含来自 Matterport 扫描仪的 3D 扫描结果，有 6 个大型室内区域组成，总共包括 271 个房间。扫描中的每个点都来自 13 个类别（椅子、桌子、地板、墙壁等以及杂物）的语义标签中的一个进行注释。

为了准备训练数据，我们首先按房间分割点，然后将房间采样为面积为 $1\text{m} \times 1\text{m}$ 的块。我们训练 PointNet 的分割版来预测每个块中的每个点类。每个点由 XYZ、RGB 和基于房间的归一化位置（从 0 到 1）的 9 维向量表示。在训练时，我们实时地从每个块中随机采样 4096 个点。在测试时，我们对所有点进行测试。我们遵循与 [1] 相同的协议，使用 k-fold 策略进行训练和测试。

我们将我们的方法与使用人工设计点特征的基准线进行比较。基准线提取相同的 9 维局部特征和三个额外特征：局部点密度、局部曲率和法线。我们使用标准 MLP 作为分类器。结果如表 3 所示，其中我们的 PointNet 方法明显优于基准线方法。在图 4 中，我们展示了定性分割结果。我们的网络能够输出平滑的预测，并且对缺失点和遮挡具有鲁棒性。

基于我们网络的语义分割输出，我们进一步构建了一个使用连接的组件进行目标采样的 3D 目标检测系统（有关详细信息，请参阅补充说明）。我们在表 4 中与之前最先进的方法进行了比较。之前的方法基于滑动形状方法（带有 CRF 后处理），这种方法使用的支持向量机针对体素网格中的局部几何特征和全局房间上下文特征进行训练。我们的方法在所报告的家具类别上的效果大大优于它。



图 4. 语义分割的定性结果上一行是带颜色纹理的点云。底下是输出的语义分割结果（以点云展示），输出结果展示同输入的相机角度一致。

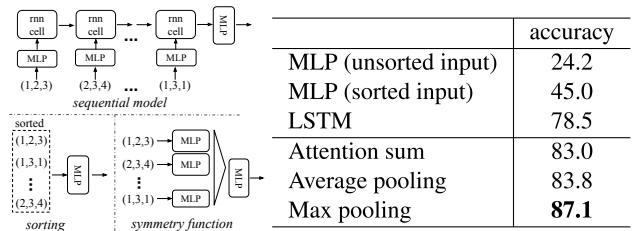


图 5. 三种实现顺序不变性的方法 神经元尺寸为 $64, 64, 64, 128, 1024$ 的 5 层隐藏层的多层感知机 (MLP) 应用在点集上，所有点共享一个简单的 MLP 副本。靠近输出的 MLP 共有两层，尺寸为 512,256。

5.2. 架构设计分析

在本节中，我们通过控制验证我们的设计选择实验。我们还展示了我们网络超参数的影响。

与替代顺序不变方法的比较 如第 4.2 节所述，至少有三种选择可用于使用无序集合输入。我们使用 ModelNet40 形状分类问题作为测试平台来比较这些选项，以下两个控制实验也将使用此任务。

我们比较的基准线（如图 5 所示）包括：未排序和已排序的 $n \times 3$ 数组的多层感知机，将输入点视为序列的 RNN 模型，以及基于对称函数的模型。我们实验的对称操作包括最大池化、平均池化和基于注意力的加权和。注意方法类似于 [25] 中的方法，即从每个点特征预测一个标量分数，然后通过计算 softmax 跨点对分数进行归一化。然后根据归一化分数和点特征计算加权和。如图 5 所示，最大池化操作以较大的优势幅度实现了最佳性能，这验证了我们的选择。

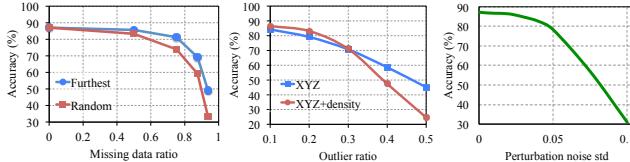


图 6. PointNet 鲁棒性测试评价指标是 ModelNet40 测试集上的总体分类准确率。左侧: 删除点的情况 Furthest 意思是对 1024 个原始点采用最远点采样。中间: 离群点插入的情况。离群点均匀的分布在单位球体中。右侧: 点扰动的情况。单独对每个点进行高斯噪声扰动

输入和特征转换的有效性 在表 5 中, 我们展示了我们的输入和特征转换 (用于对齐) 的积极影响。有趣的是, 最基本的架构已经取得了相当合理的结果。使用输入转换可提高 0.8% 的性能。正则化损失是高维变换起作用所必需的。通过结合转换和正则化项, 我们实现了最佳性能。

鲁棒性测试 我们展示了我们的 PointNet 不仅简单有效, 还对各种输入损坏具有鲁棒性。我们使用与图 5 的最大池化网络相同的架构。输入点被归一化为一个单位球体。结果如图 6。

A 对于缺失点, 当有 50% 的点缺失时, 准确率仅下降 2.4%, 并且对于最近和随机输入采样则是 3.8%。我们的网络在训练期间看到了异常点你, 那么它对异常点也具有鲁棒性,。我们评估了两种模型: 一种在具有 (x, y, z) 坐标的点上进行训练; 另一个在 (x, y, z) 上加上点密度。即使当 20% 的点是异常值, 网络也有超过 80% 的准确率。图 6 右侧显示了网络对点扰动的鲁棒性。

5.3. 可视化 PointNet

在图 7, 我们可视化了一些样本形状 S 的关键点集 \mathcal{C}_S 和上边界形状 \mathcal{N}_S (如 Thm 2 所述)。两个形状之

Transform	accuracy
none	87.1
input (3x3)	87.9
feature (64x64)	86.9
feature (64x64) + reg.	87.4
both	89.2

表 5. 输入特征转换的效果评价指标是 ModelNet40 测试集上的总体分类准确率

间的点集将给出完全相同的全局形状特征 $f(S)$ 。

从图 7 可以清楚地看出, 关键点集 \mathcal{C}_S 中对最大池化特征有贡献的点, 总结了形状的主干。上界形状 \mathcal{N}_S 说明了与输入点云 S 具有相同全局形状特征 $f(S)$ 的最大可能点云。 \mathcal{C}_S 和 \mathcal{N}_S 反映了 PointNet 的鲁棒性, 这意味着丢失一些非关键点不会改变全局形状特征 $f(S)$ 。

\mathcal{N}_S 是通过将边长为 2 的立方体中的所有点经过网络前向传播后, 选择点函数值 $(h_1(p), h_2(p), \dots, h_K(p))$ 不大于全局形状描述符的点 p 来构建的。

5.4. 时间和空间复杂度分析

表 6 总结了我们 PointNet 分类网络的空间 (网络中的参数个数) 和时间 (浮点运算/样本) 复杂性。我们还将 PointNet 与之前工作中的一组具有代表性的基于体积和多视图的架构进行了比较。

尽管 MVCNN [23] 和 Subvolume (3D CNN) [18] 实现了高性能, PointNet 的计算成本更低, 效率更高 (以 FLOPs/样本衡量: 效率分别提高 141 倍和 8 倍)。此外, 就网络中的 #param (参数少 17 倍) 而言, PointNet 的空间效率比 MVCNN 高得多。此外, PointNet 更具可扩展性——它的空间和时间复杂度为 $O(N)$ ——与输入点的数量呈线性关系。然而, 由于卷积占用计算时间, 多视图方法的时间复杂度随图像分辨率平方级增长, 基于体积卷积的方法随着体积大小成立方级增长。

根据经验, PointNet 能够处理超过每秒 100 万个点用于点云分类 (约 1K 对象/秒) 或语义分割 (约 2 个房间/秒), 在 TensorFlow 上使用 1080XGPU, 显示出实时应用的巨大潜力。

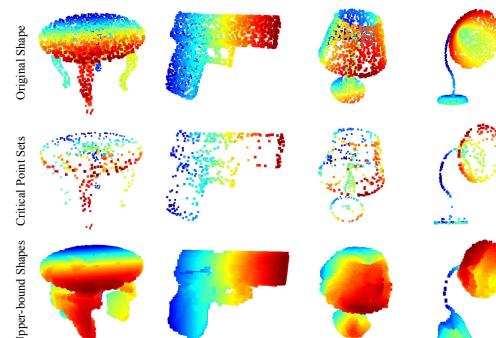


图 7. 关键点集和上界形状关键点集和上界形状共同决定了给定形状的全局形状特征, 任何处于关键点集和上界形状间的点云都有同样一致的特征。我们通过色彩编码来体现物体的深度信息。

	#params	FLOPs/sample
PointNet (vanilla)	0.8M	148M
PointNet	3.5M	440M
Subvolume [18]	16.6M	3633M
MVCNN [23]	60.0M	62057M

表 6. 处理 3D 数据分类的深度学习架构的时间和空间复杂度 PointNet (vanilla) 是分类版的 PointNet 除去输入和特征转化部分的版本。FLOP 代表浮点运算操作。“M”代表百万。Subvolume 和 MVCNN 都使用了多次旋转和多个视角池化后的数据，没有这些操作他们的表现会大大下降

6. 结论

在这项工作中，我们提出了一种新颖的、可以直接使用点云输入的深度神经网络 PointNet。我们的网络为许多 3D 识别任务提供了统一的方法，这些 3D 识别任务包括目标分类、零件分割和语义分割，同时在标准基准上获得与现有技术相当或更好的结果。我们还提供理论分析和可视化来理解我们的网络。

致谢 作者非常感谢三星 GRO 授予的资助、ONR MURI N00014-13-1-0341 的资助、NSF 授予资助 IIS-1528025、Google 重点研究奖、来自 Adobe 公司的礼物和 NVIDIA 的硬件捐赠的支持。

参考文献

参考文献

- [1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 6, 7
- [2] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011. 2
- [3] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1704–1711. IEEE, 2010. 2
- [4] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2
- [5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003. 2
- [6] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong. 3d deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2015. 2
- [7] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree. BlenSor: Blender Sensor Simulation Toolbox Advances in Visual Computing. volume 6939 of *Lecture Notes in Computer Science*, chapter 20, pages 199–208. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2011. 7
- [8] K. Guo, D. Zou, and X. Chen. 3d mesh labeling via deep convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 35(1):3, 2015. 2
- [9] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS 2015*. 4
- [10] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999. 2
- [11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, volume 6, pages 156–164, 2003. 5
- [12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 14
- [13] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas. Fpnn: Field probing neural networks for 3d data. *arXiv preprint arXiv:1605.06240*, 2016. 2
- [14] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):286–299, 2007. 2
- [15] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 16

- [16] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 37–45, 2015. 2
- [17] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2015. 2, 5, 11, 12
- [18] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 2, 5, 8, 9
- [19] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009. 2
- [20] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391. IEEE, 2008. 2
- [21] M. Savva, F. Yu, H. Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, H. Su, S. Bai, X. Bai, et al. Shrec’ 16 track large-scale 3d shape retrieval from shapenet core55. 2
- [22] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003. 14
- [23] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV, to appear*, 2015. 2, 5, 6, 8, 9
- [24] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009. 2
- [25] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015. 2, 4, 7
- [26] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. *Proceedings of the Robotics: Science and Systems, Rome, Italy*, 1317, 2015. 2
- [27] Z. Wu, R. Shou, Y. Wang, and X. Liu. Interactive shape co-segmentation via label propagation. *Computers & Graphics*, 38:248–254, 2014. 6, 11
- [28] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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, 5, 6, 12
- [29] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 6, 11, 19

附录

A. Overview

该文件为主要论文提供了额外的定量结果, 技术细节和更多定性的测试示例。

在 Sec B 中, 我们扩展了健壮性测试, 以比较不完整输入下的 PointNet 和 VoxNet。在 Sec C 中我们提供了更多有关神经网络架构和训练参数的详细信息, 在 Sec D 中我们描述了场景中的检测流程。Sec E 显示了 PointNet 的更多应用, 而 Sec F 显示了更多的分析实验。Sec G 为我们在 PointNet 上的理论提供了证明。最后, 我们在 Sec H 中显示了更多的可视化结果。

B. PointNet 和 VoxNet 的比较 (Sec 5.2)

我们扩展了第 5.2 节鲁棒性测试中的实验, 以比较 PointNet 和 VoxNet [17] (一个用于体积表示的代表性结构) 对输入点云中缺失数据的鲁棒性。两个网络都在以 1024 个点作为输入的相同训练测试集上进行训练。对于 VoxNet, 我们将点云体素化为 $32 \times 32 \times 32$ 的网格, 并通过绕上轴随机旋转和抖动来增加训练数据。

在测试时, 输入点按一定比例随机丢弃。由于 VoxNet 对旋转很敏感, 它的预测使用来自点云的 12 个视点的平均分数。如图 8 所示, 我们的 PointNet 对缺失点更加鲁棒。当一半的输入点丢失时, VoxNet 的准确率急剧下降, 从 86.3% 到 46.0%, 相差 40.3%, 而我们的 PointNet 的性能下降只有 3.7%。这可以通过我们 PointNet 的理论分析和解释来解释——它正在学习使用 *critical points* 的集合来总结形状, 因此它对缺失数据非常健壮。

C. 网络架构和训练细节 (Sec 5.1)

PointNet 分类网络 由于论文正文中已经说明了基本的体系结构, 因此我们在此处提供有关联合对齐/变换网络和训练参数的更多详细信息。

第一个转换网络是一个 mini-PointNet, 它直接以未处理的点云为输入, 并回归成一个 3×3 的矩阵。这个网络由一个每个点共享的 $MLP(64, 128, 1024)$ 网络 (输出尺寸分别为 $64, 128, 1024$), 一个点间 max pooling 和两个全连接层 (输出尺寸分别为 $512, 256$) 组成。输出矩阵初始值为单位矩阵。除了最后一层, 其余所有层都应用了 ReLU 和 Batch Normalization。第二个变换

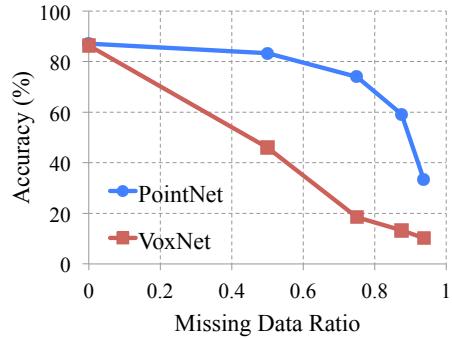


图 8. 在不完整输入数据下的 PointNet v.s. VoxNet [17] 指标是 ModelNet40 测试集上的整体分类准确率。请注意, VoxNet 使用 12 个视点平均, 而 PointNet 仅使用点云的一个视图。显然 PointNet 对缺失点具有更强的鲁棒性。

网络与第一个结构相同, 除了输出矩阵尺寸为 64×64 。输出矩阵也被初始化为单位阵。将正则化损失 (权重为 0.001) 添加到 softmax 分类损失中, 以使矩阵接近正交。

训练时, 在估计类别概率前, 对最后一层全连接层 (维度 256) 应用了 keep ratio 为 0.7 的 dropout。Batch Normalization 的衰减率从初始的 0.5 逐渐增加到 0.99。使用 adam 优化器, 初始学习率设置为 0.001, momentum 为 0.9, batch size 为 32。学习率每迭代 20 次下降一半。在 TensorFlow 下用 GTX1080 GPU 和 ModelNet 数据集训练, 网络需要 3-6 个小时收敛。

PointNet 分割网络 分割网络是 PointNet 分类网络的一个延伸。对于每个点, 将局部点特征 (第二个转换网络之后的输出) 和全局特征 (max pooling 的输出) 连接在一起。在分割网络中不使用 dropout。训练参数与分类网络相同。

关于形状零件分割的任务, 我们对基本分割网络体系结构进行了一些修改 (正文中的图 2) 以实现最佳性能。如图 9 所示。我们添加了一个 one-hot 向量来显示输入类别并与 max pooling 的输出连接。我们还在某些层中增加了神经元个数, 并添加了 skip links 以收集不同层中的局部点特征, 并将它们连接起来以形成点特征输入到分割网络中。

尽管 [27] 和 [29] 独立地处理每个对象类别, 但由于缺少某些类别的训练数据 (第一行显示了数据集中所有类别的形状总数), 我们训练了跨类别的 PointNet (但是使用 one-hot 向量来指示类别)。为了进行公平比较,

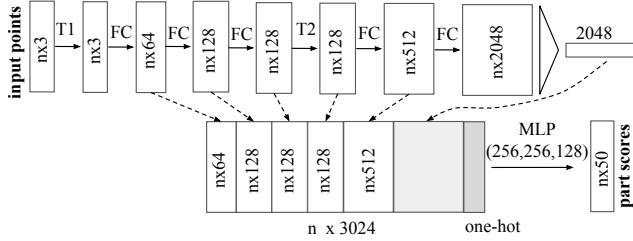


图 9. 零件分割的网络架构 T1 和 T2 是输入点和特征的对齐/变换网络。FC 是在每个点上操作的全连接层。MLP 是每一点上的多层感知器。One-hot 是一个大小为 16 的向量，表示输入形状的类别。

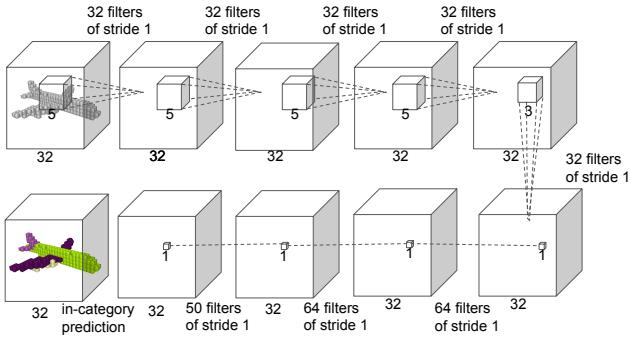


图 10. 基准 3D CNN 分割网络. 该网络是完全卷积化的，并且可以预测每个体素的部分分数

在测试这两个模型时，我们只预测给定特定对象类别的部分标签。

对于语义学分割任务，应用的是与论文主体中相同的基本网络结构，如图 2。

使用 ShapeNet part 数据集训练需要大约 6-12 小时，使用 Stanford semantic parsing 数据集训练大概需要半天。

基准 3D CNN 分割网络 在 ShapeNet 零件分割实验中，我们将建议的分割版本 PointNet 与两种传统方法以及 3D 体积 CNN 网络基准进行了比较。在图 10 中，我们展示了我们使用的基准 3D 体积 CNN 网络。我们将众所周知的 3D CNN 架构，例如 VoxNet [17] 和 3DShapeNets [28] 推广到完全卷积的 3D CNN 分割网络。

对于给定的点云，我们首先将其转换为具有 $32 \times 32 \times 32$ 分辨率的占用网格的体积表示形式。然后，依次使用五个具有 32 个输出通道和步长为 1 的 3D 卷积核来提取特征。每个体素的感受野为 19。最后，将大小

为 $1 \times 1 \times 1$ 的 3D 卷积序列附加到计算出的特征图上，以预测每个体素的分割标签。除最后一层外，所有层均使用 ReLU 和 Batch Normalization。该网络是跨类别训练的，但是，为了与给出对象类别的其他基准方法进行比较，我们仅考虑给定对象类别中的输出得分。

D. Details on Detection Pipeline (Sec 5.1)

我们基于语义分割结果和我们的对象分类 PointNet 构建了一个简单的 3D 对象检测系统。

我们使用带有分割分数的连接组件来获取场景中的对象提议。从场景中的一个随机点开始，我们找到它的预测标签，并使用 BFS 搜索附近具有相同标签的点，搜索半径为 0.2 米。如果结果集群有超过 200 个点（假设 $1m \times 1m$ 区域中有 4096 个点样本），则集群的边界框被标记为一个对象提议。对于每个提议的对象，它的检测分数计算为该类别的平均点分数。在评估之前，面积/体积极小的提案被裁剪。对于桌子、椅子和沙发，边界框会延伸到地板，以防腿与座椅/表面分开。

我们观察到，在某些房间（例如礼堂）中，许多物体（例如椅子）彼此靠近，其中连接的组件无法正确地分割出单个的物体。因此，我们利用我们的分类网络并使用滑动形状方法来缓解椅子类的问题。我们为每个类别训练一个二元分类网络，并使用分类器进行滑动窗口检测。结果框通过非最大抑制进行裁剪。将来自连接组件和滑动形状的建议框组合起来进行最终评估。

在图 11 中，我们展示了目标检测的准确率-召回率曲线。我们训练了六个模型，其中每个模型都在五个区域进行训练并在左侧区域进行测试。在测试阶段，每个模型都在它从未见过的区域进行测试。将所有六个区域的测试结果汇总以生成 PR 曲线。

E. 更多应用 (Sec 5.1)

从点云中检索模型 我们的 PointNet 为每个给定的输入点云学习全局形状特征。我们期望几何相似的形状具有相似的全局特征。在本节中，我们将在形状检索应用程序上测试我们的猜想。更具体地说，对于由 ModelNet 划分的测试集中每个给定查询形状，我们计算其由我们的分类 PointNet 给出的全局签名（分数预测层之前的层的输出），并通过最近邻搜索在划分的训练集中检索相似的形状。结果如图 12 所示。

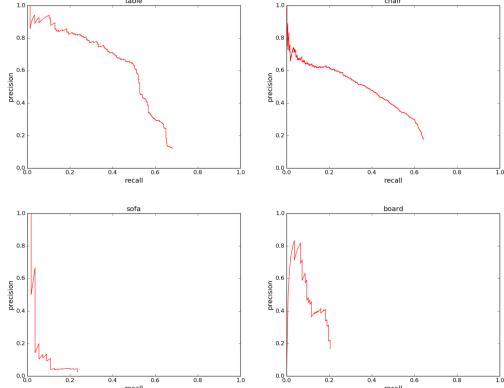


图 11. 用于 3D 点云中对象检测的准确率-召回率曲线。我们在所有的六个区域上对四个类别进行了评估：桌子、椅子、沙发和木板。IoU 阈值在体积上表现为 0.5。

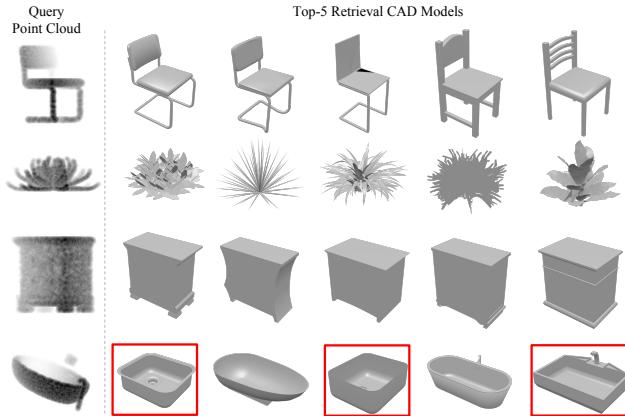


图 12. 从点云中检索模型。对于每个给定的点云，我们从 ModelNet 划分的测试集中检索前 5 个相似的形状。从上到下，我们展示了椅子、植物、床头柜和浴缸查询的示例。错误类别的检索结果用红色框标记。

形状对应 在本节中，我们展示了 PointNet 学习的点特征可以潜在地用于计算形状对应。给定两个形状，我们通过匹配激活全局特征中相同维度的点对来计算它们的 *critical point sets* C_S 之间的对应关系。Fig 13 和 Fig 14 显示了检测到的两个相似椅子和桌子之间的形状对应关系。

F. 更多结构分析 (Sec 5.2)

瓶颈维度和输入点数量的有效性 根据第一个输入层的大小和输入点云的数量，我们的模型的表现会有所变化，接下来我们将阐释其变化。从图 15 中我们可以看出，随着我们提高点的数量，性能变得越来越好。在大约 1000 个点时，达到饱和。最大的层尺寸同样起了重

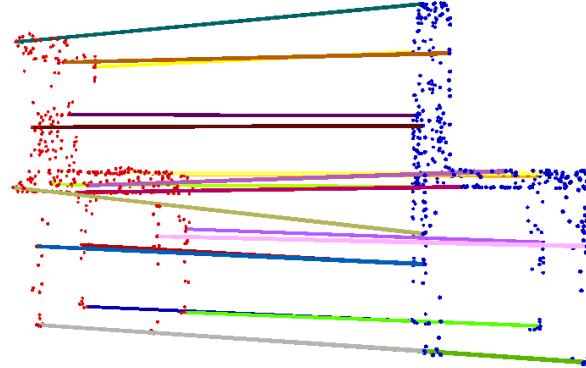


图 13. 两个椅子之间的形状对应。为了可视化的清晰性，我们只展示了 20 个随机挑选的映射对。

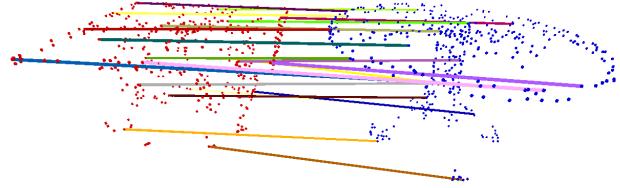


图 14. 两张桌子之间的形状对应。为了可视化的清晰性，我们只展示了 20 个随机挑选的映射对。

要作用，让我们将层大小从 64 增大到 1024 之后，表现提升了 2 – 4%。这说明我们需要足够多的，覆盖整个三维空间的点特征函数来分辨不同的形状。

值得一提的是，即便有 64 个点作为输入（从网络上采样的点中最远的点），我们的网络仍然能够取得令人满意的效果。

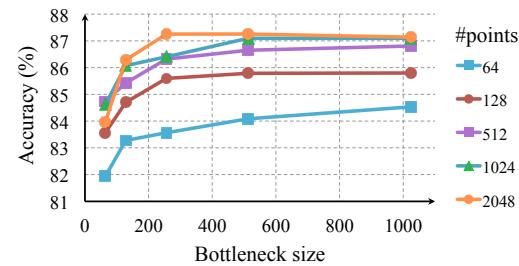


图 15. 瓶颈尺寸和输入点数量的效果该指标为 ModelNet40 上的总体的分类准确度。

MNIST 数字分类 在我们关注 3D 点云的学习的同时，我们在用我们的网络在 2D 点云上做了实验，即像素集。

为了将一个 MNIST 图像转换成一个 2D 点云，我们设置了阈值像素值，将所有值大于 128 的像素（用图

像中的 (x, y) 坐标表示) 加入到集合中。我们使用了一个大小为 256 的集合大小。如果集合中有多于 256 个像素, 我们就从中随机地抽样; 如果集合中的元素少于 256 个, 我们便用集合中存在的点来做填充 (由于我们的最大化操作, 使用哪一个点来填充并不会影响结果)。

如表 7 所示, 我们将其和若干个基准线进行比较 (包括将输入图像作为有序向量的多层感知机、将输入图像作为一个从 $(0,0)$ 像素到 $(27,27)$ 像素的循环神经网络、以及一个简易的卷积神经网络)。虽然其中表现最好的模型仍然是一个仔细调试过的卷积神经网络 (达到了低于 0.3% 的错误率), 但 PointNet 的表现在二维点集上的表现已经符合预期。

	输入	错误率 (%)
多层感知机 [22]	vector	1.60
LeNet5 [12]	image	0.80
我们的 PointNet	point set	0.78

表 7. **MNIST 分类结果** 我们将我们的模型和其他简单的深度学习结构进行比对, 从而说明我们的模型在二维点云上的表现仍然符合预期。

法线预测 在分割版本的 PointNet 中, 为了为局部的点提供上下文信息, 局部的点特征与全局的点特征被拼接在了一起。但这种拼接方式不能保证模型学习到上下文。在该实验中, 我们通过展示我们的分割网络具备预测点法线 (一个由某个点的邻居所决定的局部几何特征) 的能力, 来验证我们的设计。

我们用一种监督学习的方式修改并训练了我们的分割 PointNet 来回归到真正的点法线。我们仅仅改变了分割 PointNet 的最后一层, 使其为每一个点预测法向量。对于损失函数, 我们使用了余弦距离的绝对值。

图 16 对比了我们的 PointNet 法线预测结果 (左) 和从网格中计算出来的真正的法线 (右)。从图中可以看到, 我们得到了一个合理的法线重建结果。在一些区域, 我们的预测结果甚至比真正的法线 (通常包括一些翻转了的法线方向) 更加平滑和连续。

分割健壮性 如 5.2 和 B 讨论, 由于全局的形状特征是从一些 **关键点** 中提取出来的, 我们的点云对于数据干扰和点的缺失并不敏感。在这一部分中, 我们展示了我们的模型在面对分割任务时, 仍然保持着健壮性。每一

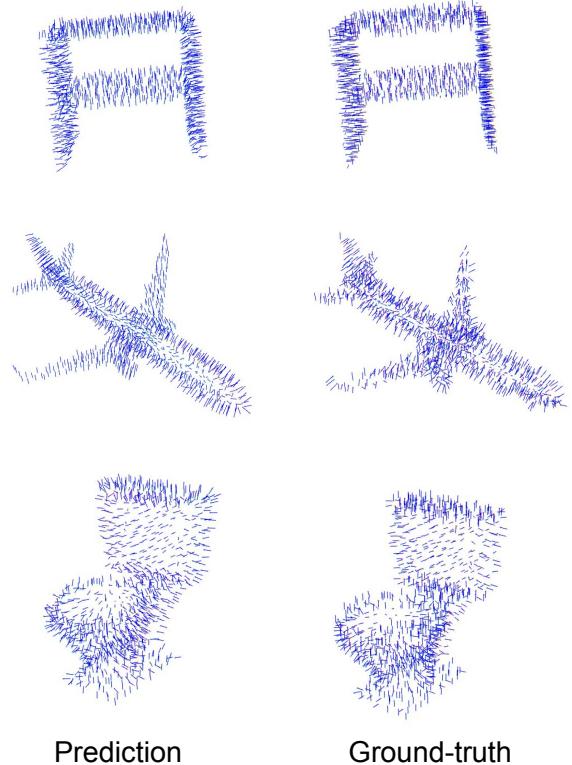


图 16. **PointNet 法线重建结果**。改图展示了在一个示例点云中所有点的法线重建结果与从网格中计算出来的真正的法线结果。

个点的标签是根据两部分来预测的: 各个点的特征的结合, 与模型学习到的全局形状特征。在图 17 中, 我们展示了给定输入点云 S (左) 的分割结果、关键点集 \mathcal{C}_S (中) 和上界形状 \mathcal{N}_S 。

在 5.2 三维物体零件分割中, 我们将我们提出的 PointNet 模型应用在了 CAD 模型的语义零件分割上。我们的分割 PointNet (如图 2 分割网络) 在完整的 ShapeNet 网络上达到了最优的效果。意料之中的, 我们的模型在局部数据 (如: 模拟的 Kinect 扫描结果) 中也表现的很好。由于真实世界中的扫描通常因为遮挡问题而变得十分不完整, 模型在面对局部输入时的健壮性非常重要, 是评估其实际应用价值的关键。表 8 概括了在面对完整的和局部的数据时, 我们的 PointNet 与作为基准线的三维卷积神经网络方法得出的效果。

网络面对未知形状分类的通用性 在图 18 中, 我们可视化了 **关键点集** 和 **上界形状** 在面对来自从未出现在 ModelNet 或 ShapeNet 中的未知分类 (脸部、马、兔

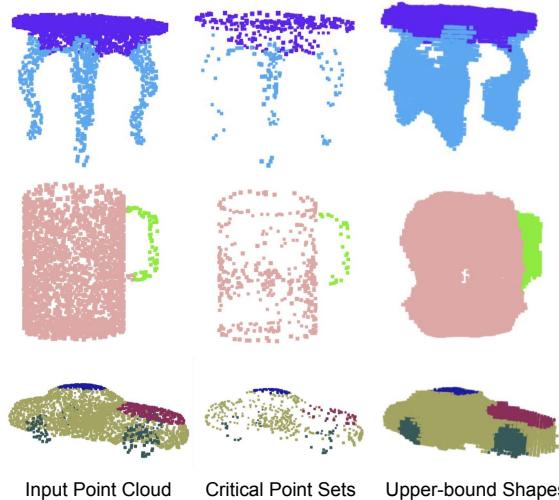


图 17. 分割结果的一致性 我们用图例说明一些样例物体点云的关键点集 \mathcal{C}_S 和上界形状 \mathcal{N}_S 的分割结果。可以得出从 \mathcal{C}_S 到 \mathcal{N}_S 的形状簇拥有共同的分割结果。

	完整输入	局部输入
3D CNN	75.3	69.7
Ours PointNet	80.6	75.3

表 8. 局部扫描的分割结果 衡量指标为所有形状的 mIoU。在完整数据上训练我们的点网络时，我们进行了旋转增强，以便与从多个角度生成的模拟 Kinect 扫描进行公平的比较。两个网络分别在完整数据和部分数据上进行训练，然后在不同的子测试集上进行了测试。

子、茶壶) 的新性状。结果表明，我们的模型学习到的各个点的函数是具备通用性的。然而，由于我们主要使用具备大量平面结构的人造物品来训练模型，重建出来的上界形状同样也包含了更多的平面表面。

G. 定理证明 (Sec 4.3)

令 $\mathcal{X} = \{S : S \subseteq [0, 1] \text{ and } |S| = n\}$.

若以下条件满足，则 $f : \mathcal{X} \rightarrow \mathbb{R}$ 是一个 \mathcal{X} 上关于赫斯多夫距离 $d_H(\cdot, \cdot)$ 的连续函数

$\forall \epsilon > 0, \exists \delta > 0$, 对于任意一个 $S, S' \in \mathcal{X}$, if $d_H(S, S') < \delta$, then $|f(S) - f(S')| < \epsilon$.

我们证明了 f 可以被对称函数和连续函数的复合函数任意逼近

Theorem 1. 假设 $f : \mathcal{X} \rightarrow \mathbb{R}$ 是一个关于赫斯多夫距离 $d_H(\cdot, \cdot)$ 的连续集函数。 $\forall \epsilon > 0$, \exists 一个连续函数 h 和一个对称函数 $g(x_1, \dots, x_n) = \gamma \circ MAX$, 其中 γ 一个连

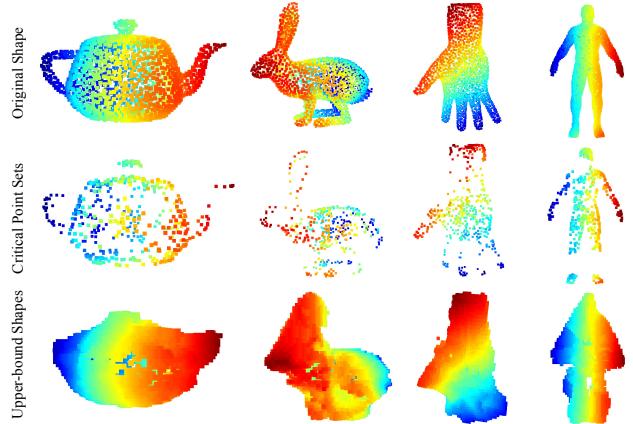


图 18. 未知物体的关键点集和上界形状 我们可视化了茶壶，兔子，手，人体的关键点集和上界形状。通过这些不在 ModelNet 和 ShapeNet 数据集的物体来测试 PointNet 学习到的点函数的通用性。图中用颜色编码反应深度信息。

续函数, MAX 是一个取 n 个向量最为输入, 然后返回一个元素层面上的最大值的最大化向量运算符, 对于任意一个 $S \in \mathcal{X}$,

$$|f(S) - \gamma(MAX(h(x_1), \dots, h(x_n)))| < \epsilon$$

其中 x_1, \dots, x_n 是按指定顺序从 S 中提取的元素。

证明. 通过 f 的连续性, 引入 δ_ϵ 来说明 $|f(S) - f(S')| < \epsilon$ 对于任意一个 $S, S' \in \mathcal{X}$ if $d_H(S, S') < \delta_\epsilon$.

定义 $K = \lceil 1/\delta_\epsilon \rceil$, 将 $[0, 1]$ 均分成 K 个间隔, 同时定义辅助函数, 将点映射到它所在间隔的左端:

$$\sigma(x) = \frac{\lfloor Kx \rfloor}{K}$$

令 $\tilde{S} = \{\sigma(x) : x \in S\}$

$$|f(S) - f(\tilde{S})| < \epsilon$$

因为 $d_H(S, \tilde{S}) < 1/K \leq \delta_\epsilon$.

令 $h_k(x) = e^{-d(x, [\frac{k-1}{K}, \frac{k}{K}])}$ 为一个轻指示函数, 其中 $d(x, I)$ 是点的间隔距离。令 $\mathbf{h}(x) = [h_1(x); \dots; h_K(x)]$, $\mathbf{h} : \mathbb{R} \rightarrow \mathbb{R}^K$.

令 $v_j(x_1, \dots, x_n) = \max\{\tilde{h}_j(x_1), \dots, \tilde{h}_j(x_n)\}$ 代表第 j -th 区间中 S 中点的占比。令 $\mathbf{v} = [v_1; \dots; v_K]$, $\mathbf{v} : \underbrace{\mathbb{R} \times \dots \times \mathbb{R}}_n \rightarrow \{0, 1\}^K$ 是一个对称函数, 代表每个区间中 S 中点的占比。

定义 $\tau : \{0, 1\}^K \rightarrow \mathcal{X}$ 为 $\tau(v) = \{\frac{k-1}{K} : v_k \geq 1\}$, 代表占用向量到表示每个占用区间左端集合的映射。易得:

$$\tau(\mathbf{v}(x_1, \dots, x_n)) \equiv \tilde{S}$$

其中 x_1, \dots, x_n 是按指定顺序从 S 中提取的元素。

令 $\gamma : \mathbb{R}^K \rightarrow \mathbb{R}$ 为一个连续函数, 对于 $v \in \{0, 1\}^K$ 有 $\gamma(\mathbf{v}) = f(\tau(\mathbf{v}))$ 对于 $v \in \{0, 1\}^K$. 则

$$\begin{aligned} & |\gamma(\mathbf{v}(x_1, \dots, x_n)) - f(S)| \\ &= |f(\tau(\mathbf{v}(x_1, \dots, x_n))) - f(S)| < \epsilon \end{aligned}$$

注意 $\gamma(\mathbf{v}(x_1, \dots, x_n))$ 可以被写成以下形式:

$$\begin{aligned} \gamma(\mathbf{v}(x_1, \dots, x_n)) &= \gamma(\text{MAX}_{x_i \in S}(\mathbf{h}(x_1), \dots, \mathbf{h}(x_n))) \\ &= (\gamma \circ \text{MAX})(\mathbf{h}(x_1), \dots, \mathbf{h}(x_n)) \end{aligned}$$

显然 $\gamma \circ \text{MAX}$ 是一个对称函数。 \square

接下来给出定理 2 的证明。定义 $\mathbf{u} = \text{MAX}_{x_i \in S}\{h(x_i)\}$ 为 f 的子网络, 表示 $[0, 1]^m$ 上点集到 K -维向量的映射。以下定理表明输入点集中较少点缺失或者额外微小点扰动不会改变我们网络的输出:

Theorem 2. 假设 $\mathbf{u} : \mathcal{X} \rightarrow \mathbb{R}^K$ 其中 $\mathbf{u} = \text{MAX}_{x_i \in S}\{h(x_i)\}$ 且 $f = \gamma \circ \mathbf{u}$. 则,

(a) $\forall S, \exists \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X}, f(T) = f(S)$ if $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$;

(b) $|\mathcal{C}_S| \leq K$

证明. 显然, $\forall S \in \mathcal{X}, f(S)$ 由 $\mathbf{u}(S)$ 所决定。所以我们只需要证明 $\forall S, \exists \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X}, f(T) = f(S)$ if $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$.

对于输出向量 \mathbf{u} 的第 j 维, 存在至少一个 $x_j \in \mathcal{X}$ 使得 $h_j(x_j) = \mathbf{u}_j$, 其中 h_j 是 h 中第 j 维输出向量。将 \mathcal{C}_S 作为所有 $x_j, j = 1, \dots, K$ 的聚合, 则 \mathcal{C}_S 满足上述条件。

因此对于 f , 附加任意额外的点 x 使得 \mathcal{C}_S 的每一维满足 $h(x) \leq \mathbf{u}(S)$ 而不改变 \mathbf{u} 。从而得出, 可以把所有这样点添加到 \mathcal{N}_S 来得到 \mathcal{T}_S 。 \square

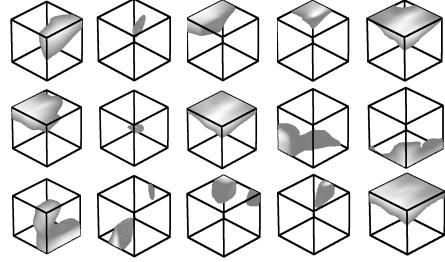


图 19. 点函数可视化 对于每个点函数 h , 计算位于原点的边长为 2 的立方体中每个点 p 的 $h(p)$ 值, 该区域覆盖了训练 PointNet 时输入数据规范化的单位球区域。在这幅图中, 我们可视化所有 $h(p) > 0.5$ 的点 p , 函数值用体素亮度表示。我们随机选取 15 个点函数并可视化其活跃区域。

H. 更多的可视化

分类可视化 我们使用 t-SNE[15]把从我们的分类 PointNet 中得到的点云全局结果 (1024-dim) 降维嵌入到 2D 空间中。图 20 显示了 ModelNet40 用于测试的分割形状的嵌入空间。相似的形状根据它们的语义类别分别聚集到一起。

分割可视化 我们在完整的 CAD 模型和模拟 Kinect 部分扫描上提供了更多的分割结果。我们还可视化了具有错误分析的失败案例。图 21 和图 22 显示了在完整 CAD 模型及其模拟 Kinect 扫描上生成的更多分割结果。图 23 说明了一些失败案例情况。请阅读错误分析的说明。

场景语义解析可视化 我们在图 24 中给出了语义解析的可视化, 其中我们显示了对应两个办公室和一个会议室, 用于语义分割和目标检测的输入点云、预测和真实标签。该区域和房间在训练集中是不可见的。

点函数可视化 我们的分类 PointNet 为每个点计算 K (我们在此可视化中取 $K = 1024$) 维的点特征, 并通过最大池化层将所有每点局部特征聚合为单个 K 维向量, 从而形成全局形状描述符。

为了更深入地了解每个已经训练好的点函数 h 所检测的内容。我们在图 19 中可视化了具有较高的点函数值 $f(p_i)$ 的点 p_i 。该可视化清晰地表明, 不同的点函数能够检测分散在整个空间中的不同区域中形状各异的点。

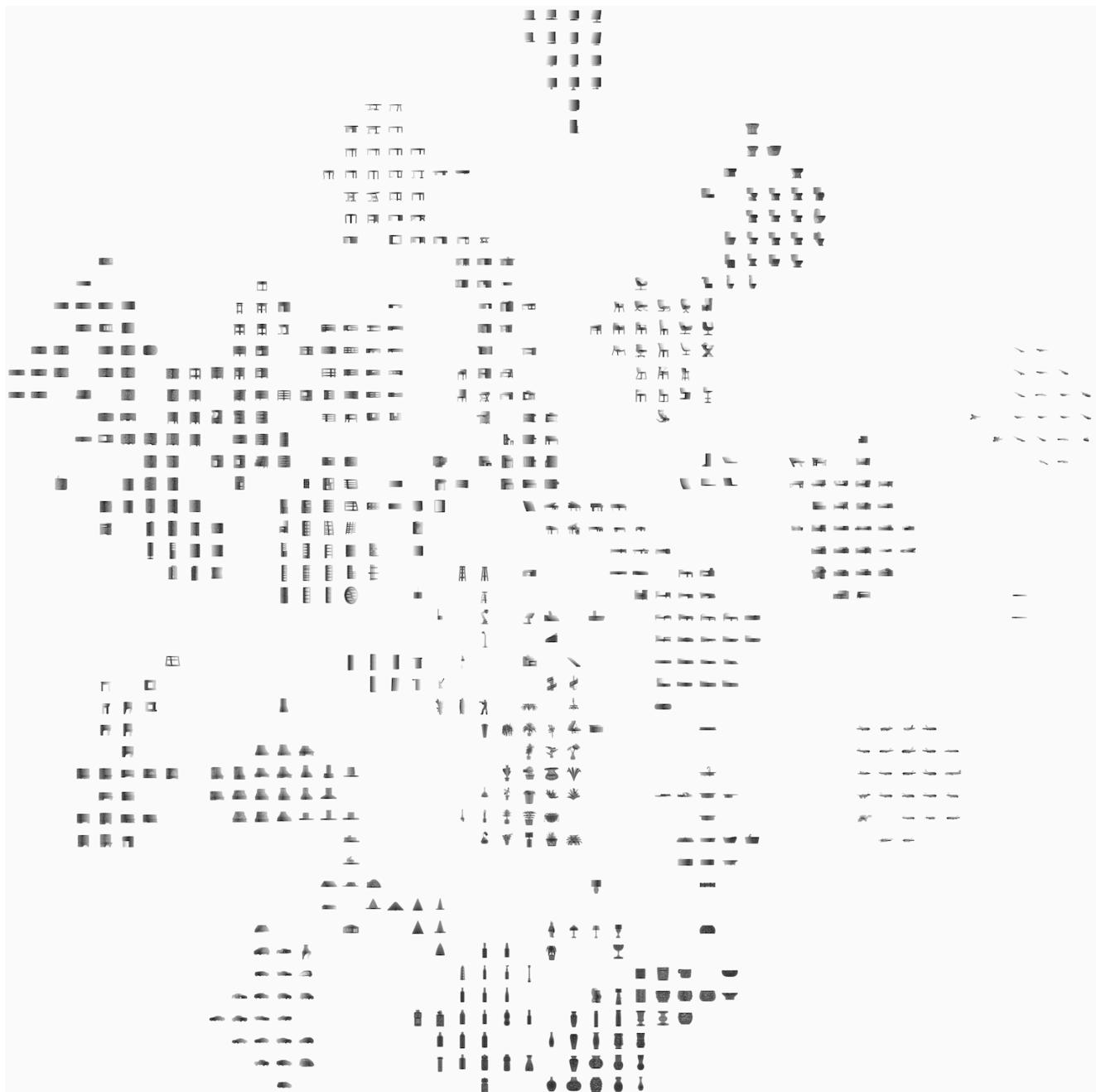


图 20. 所学到的形状全局特征的二维嵌入 我们使用 t-SNE 技术将学习到的全局形状特征可视化为 ModelNet40 测试时拆分的形状。

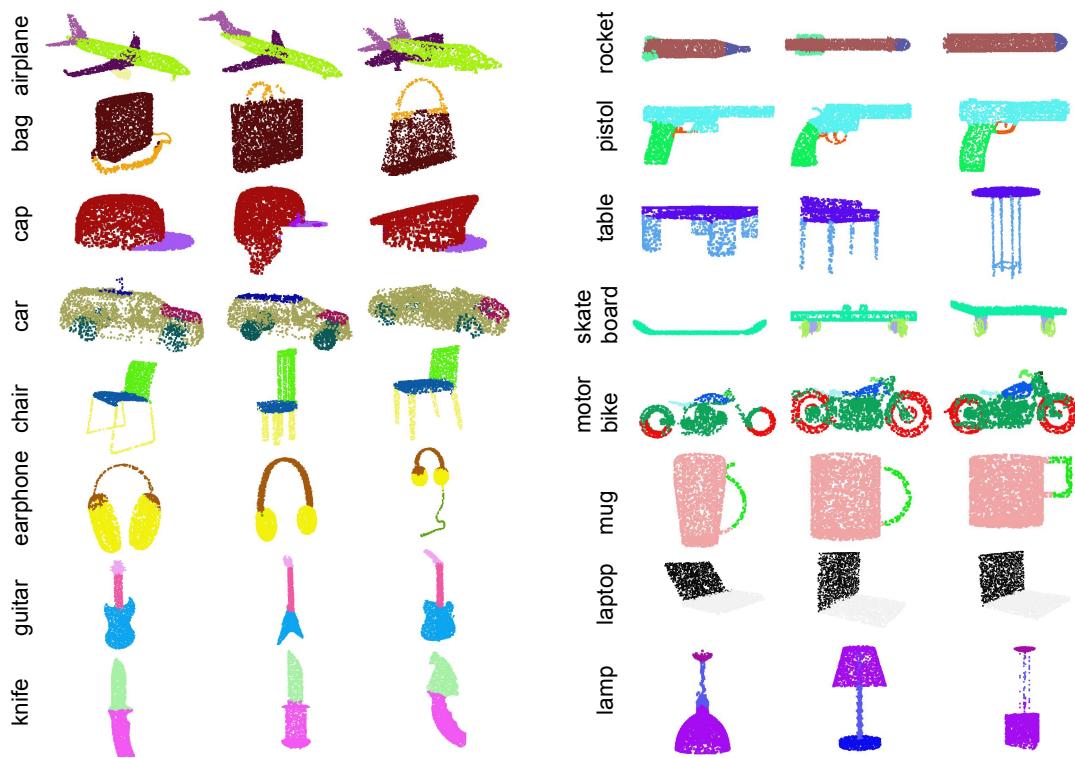


图 21. PointNet 在完整的 CAD 模型上的分割结果

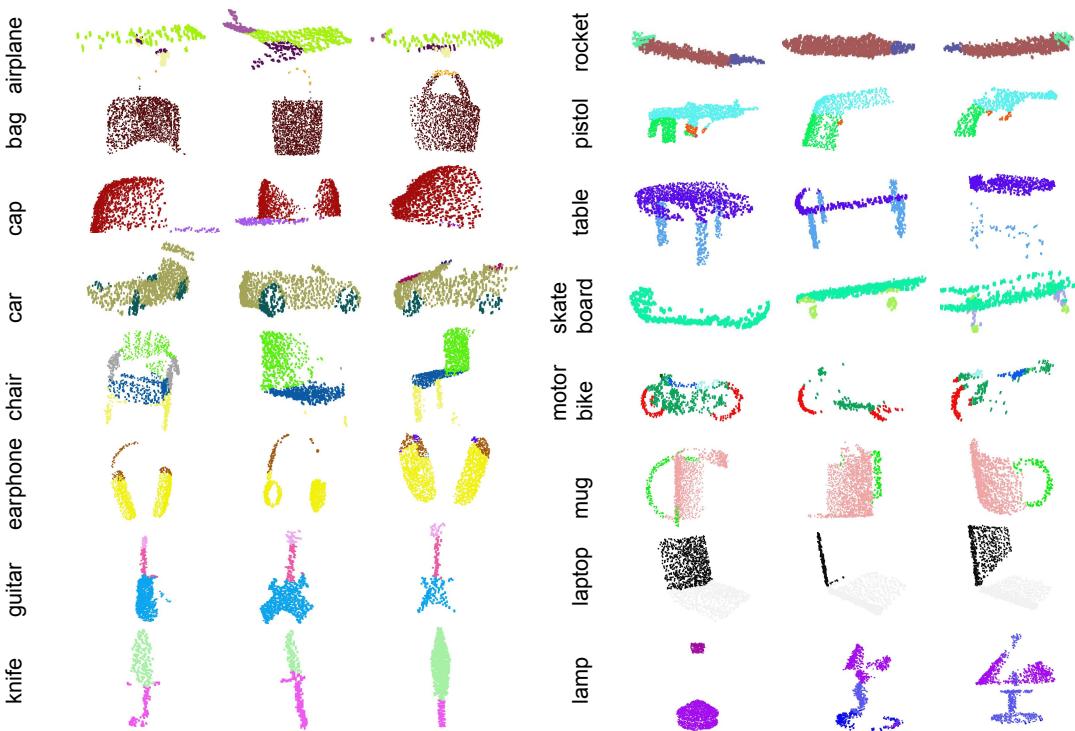


图 22. PointNet 在模拟 Kinect 扫描上的分割结果

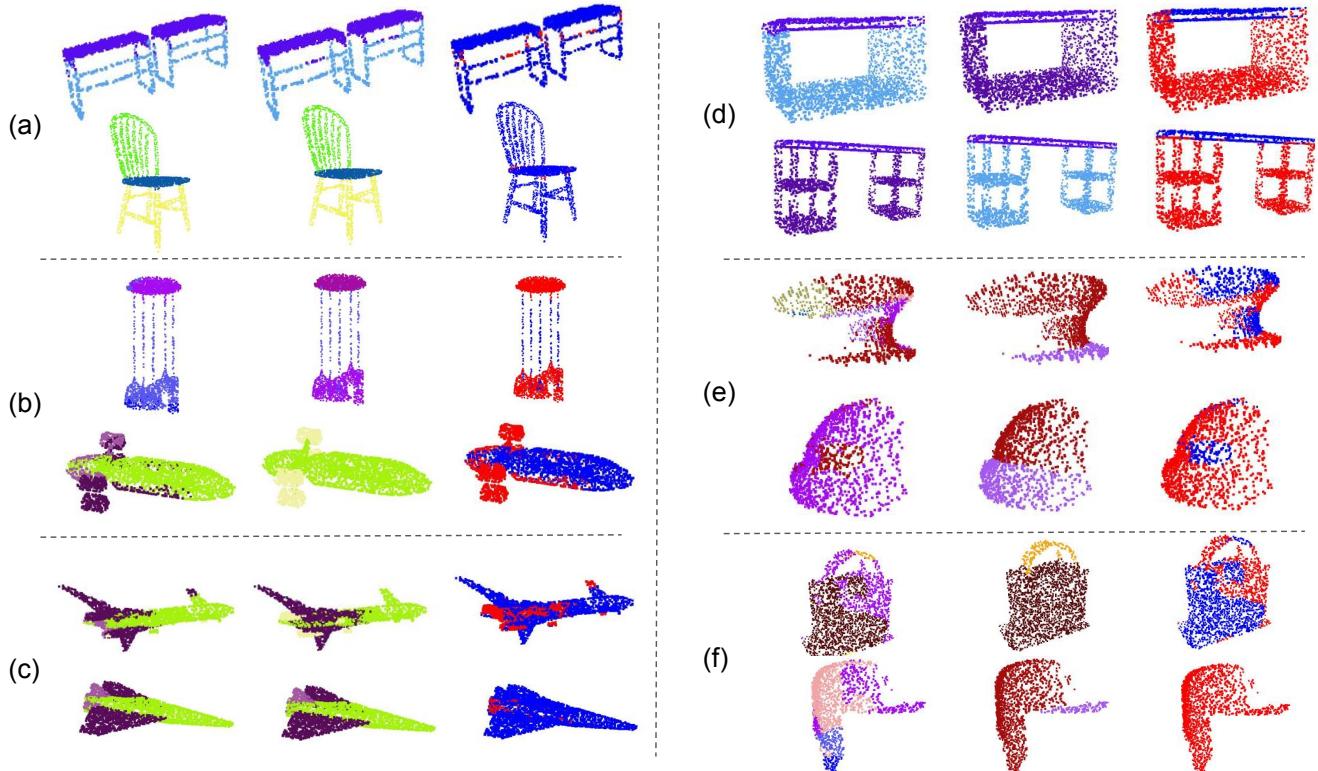


图 23. PointNet 分割失败案例在此图中，我们总结了分割应用程序中的六种常见错误。在第一列和第二列中分别给出了预测和真实分割结果，而差异图经计算后显示在第三列。红点对应于给定点云中标记错误的点。(a) 说明了最常见的失败案例：边界上的点被错误标记。在示例中，桌椅腿和顶部之间的交叉线附近的点的标签预测不准确。然而，大多数分割算法都存在这个错误。(b) 显示了奇怪形状的错误。例如，图中所示的吊灯和飞机在数据集中非常少见。(c) 表明小部分可以被附近的大部分覆盖。例如，飞机的喷气发动机（图中黄色）被错误地分类为机身（绿色）或机翼（紫色）。(d) 显示了由形状部分固有的模糊性引起的错误。例如，图中两张桌子的两个底部被分类为桌腿和桌脚 ([29] 中的其他类别)，而真实分割则是相反的。(e) 显示了由部分扫描的不完整性导致的错误。对于图中的两个帽子，几乎一半的点云都缺失了。(f) 显示了当某些目标类别的训练数据太少而无法涵盖足够多样性时产生的失败案例。对于此处显示的两个类别，整个数据集中只有 54 个袋子和 39 个帽子。

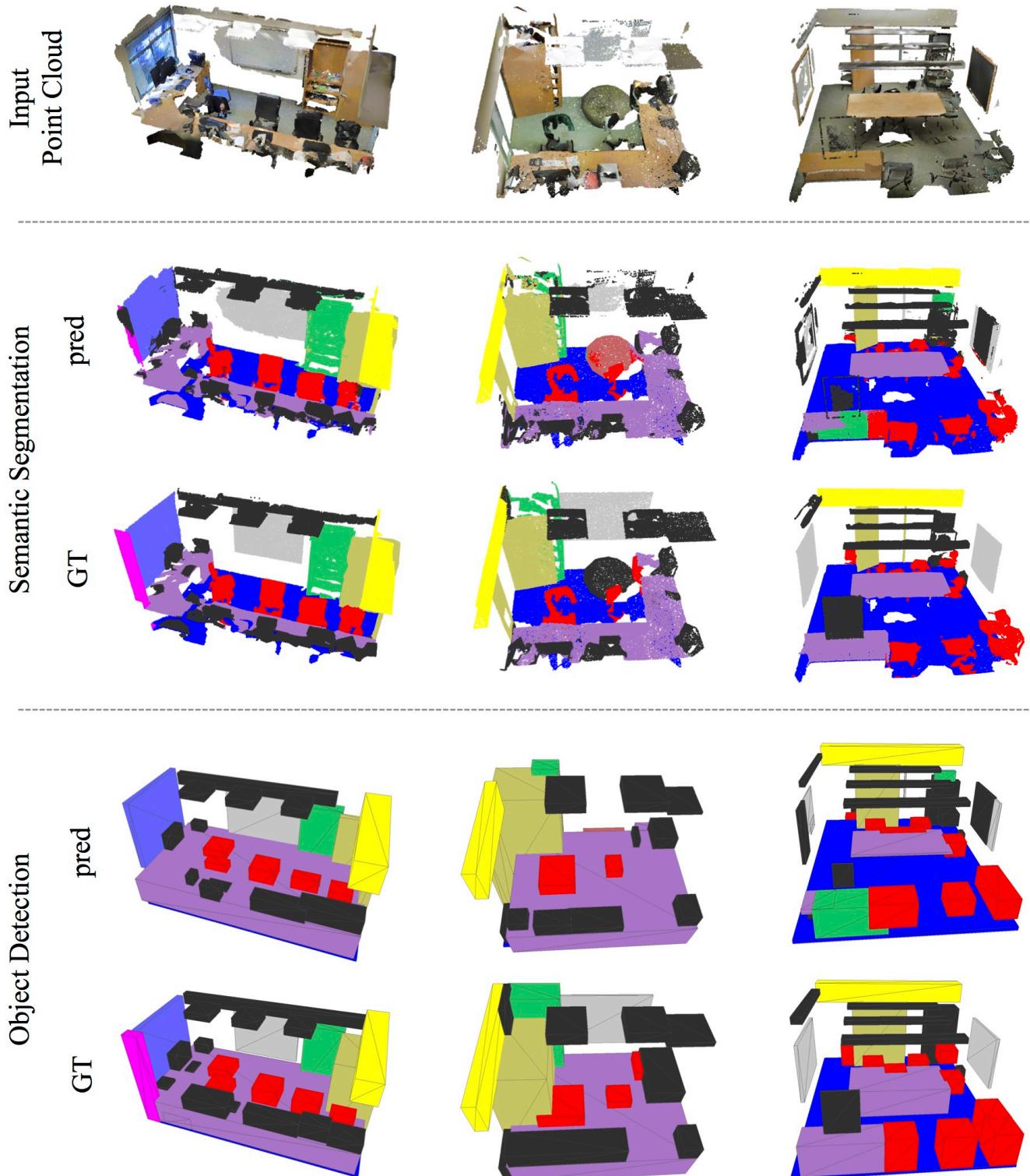


图 24. 语义分割和目标检测示例第一行是输入点云，为了清晰度，其中墙壁和天花板被隐藏了。第二行和第三行分别是点的语义分割的预测和真实分割，其中属于不同语义区域的点的颜色不同（椅子为红色，桌子为紫色，沙发为橙色，木板为灰色，书柜为绿色，地板为蓝色，窗户为紫色，横梁为黄色，柱子为洋红色，门为卡其色和杂物为黑色）。最后两行是带有边界框的目标检测，其中预测框来自基于语义分割预测的连接部分。