# Unsupervised Learning of Shape and Pose with Differentiable Point Clouds

论文链接：<https://arxiv.org/abs/1810.09381>

Github：<https://github.com/eldar/differentiable-point-clouds>

## 摘要

我们解决了从一组未标记的特定类别图像中学习准确的 3D 形状和相机姿势的问题。 我们训练卷积网络**通过最小化重投影误差**来预测单个图像的形状和姿势：给定一个物体的多个视角，预测形状到预测相机姿势的投影应该与提供的视角匹配。为了处理姿势歧义，我们引入了一组姿势预测器，然后我们将其提炼为单个“学生”模型。**为了有效地学习高保真形状，我们用点云表示形状，并设计了一个公式，允许对这些形状进行可微分投影**。我们的实验表明，姿态预测器的蒸馏集合学会了准确估计姿态，而点云表示允许预测详细的形状模型。

## 1. 介绍

我们生活在一个三维世界中，正确理解其体积结构对于表演和计划至关重要。 然而，我们主要通过二维投影来感知世界。基于这些投影，我们能够推断出周围物体的三维形状和姿态。 这种体积形状的感知是如何从仅从二维投影中观察而来的？是否可以设计具有类似功能的学习系统？

深度学习方法最近显示出解决这些问题的希望 [25, 20]。给定一组对象的视角和相应的相机姿势，这些方法通过重投影误差学习 3D 形状：给定估计的形状，可以将其投影到已知的相机视图并与提供的图像进行比较。这些生成的投影与训练样本之间的差异为改进形状估计提供了训练信号。这种类型的现有方法有两个常见的限制。首先，**这些方法假设所有提供的图像的相机姿势都是已知的。这是一个实际和生物学上不切实际的假设**：典型的智能代理只能访问其观察结果，而不能访问其相对于世界上物体的精确位置。其次，**形状被预测为低分辨率（通常为 体素）体素化体积**。这种表示只能描述对象的非常粗略的形状。应该可以从 2D 监督中学习更精细的形状细节。

在本文中，我们**仅从投影中学习高保真形状模型，没有地面实况相机姿势**。由于两个原因，这种设置具有挑战性。首先，**估计形状和姿势是一个先有鸡还是先有蛋的问题**：如果没有良好的形状估计，就不可能学习准确的姿势，因为投影不会提供信息，反之亦然，需要准确的姿势估计来学习形状。其次，**姿态估计容易出现由歧义引起的局部最小值**：一个对象可能从两个角度看起来相似，如果网络收敛到在所有情况下只预测其中一个，它将无法学习预测另一个。我们发现第一个问题可以**通过联合优化形状和姿势预测器得到很好的解决**：在实践中，即使使用相对嘈杂的姿势预测，也可以学习到良好的形状估计。然而，第二个问题会导致姿势估计的严重错误。为了解决这个问题，我们**训练了一个多样化的姿势预测器集合，并将它们提炼成一个单一的学生模型**。

为了允许学习高保真形状，我们使用点云表示，与之前工作中使用的体素形成对比。**点云允许计算效率高的处理，可以产生高质量的形状模型** [6]，并且在概念上很有吸引力，因为它们可以被视为“以物质为中心”，而不是“以空间为中心”的体素网格。**为了在没有明确 3D 监督的情况下学习点云，我们实现了一个可微投影算子，给定一个点集和一个相机姿势，生成一个 2D 投影——轮廓、彩色图像或深度图**。我们将公式称为“可微分点云”。

我们在从**一个物体的单个图像估计形状和相机姿态的任务上评估所提出的方法**。该方法成功地学会了预测形状和姿势，相对于使用地面实况相机姿势训练的模型，性能仅略有下降。 当提供足够高分辨率的图像作为监督时，基于点云的公式允许有效学习高保真形状模型。我们展示了从轮廓中学习点云，如果在训练期间有彩色图像可用，则用颜色增强点云。最后，我们展示了点云表示如何允许自动发现对象之间的语义对应关系。

## 2. 相关工作

从二维投影重建三维形状在计算机视觉中有着悠久的历史，构成了 3D 重建领域。对该领域的回顾超出了本文的范围。但是，我们简要列出了几种相关方法。Cashman 和 Fitzgibbon [2] 使用轮廓和关键点注释从小类特定图像集合中重建可变形形状模型，Vicente 等人。 [22] 将类似的方法应用于大规模 Pascal VOC 数据集，Tulsiani 等人。 [18] 通过利用计算机视觉技术减少所需的监督。即使在小数据范围内，这些方法也显示出令人印象深刻的结果；然而，**它们在表现多样化和复杂的形状方面存在困难**。 Loper 和 Black [12] 实现了可微分渲染器并将其应用于综合分析。我们的工作在精神上是相似的，但在点云上运行，并将可微渲染的思想与深度学习相结合。罗丹等人的方法。 [14] 在技术上与我们的类似，因为它使用一组高斯密度函数对人体进行建模，并使用物理驱动的光传输方程来渲染它们。与我们的方法不同，表示没有集成到学习框架中，并且需要小心地初始放置高斯，使其不适合任意形状类别的自动重建。此外，投影方法与高斯的数量成二次方比例，这限制了所表示形状的最大保真度。

最近，通过基于深度学习的方法解决了从 2D 监督中学习 3D 结构的任务。这些方法通常基于重投影误差——将预测的 3D 形状的 2D 投影与 GT 的 2D 投影进行比较。严等人。 [25] 通过基于选择沿射线的最大占用值的投影操作，从轮廓中学习 3D 形状。图尔西亚尼等人。 [20] 设计了一种基于射线碰撞概率的可微公式，并将其应用于从轮廓、深度图、彩色图像和语义分割图中学习。林等人。 [11] 通过深度图表示点云，并使用高分辨率网格和逆深度最大池化重新投影它们。与我们同时，加藤等人。 [8] 提出了一种网格的可微渲染器，并将其用于学习基于网格的对象形状表示。**所有这些方法都需要与用于训练的 2D 投影相对应的精确的相机姿势 GT。相比之下，我们的目标是放宽这种不切实际的假设，只从预测中学习**。

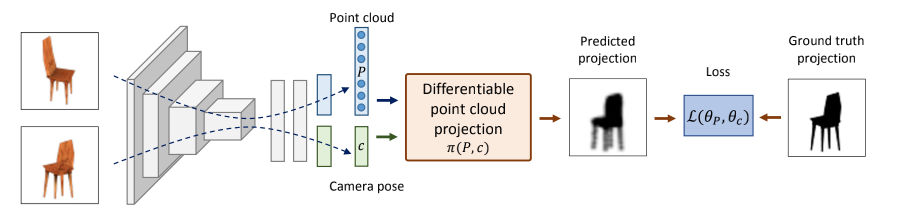
Rezende 等[13] 探索了几种基于 2D 视角对 3D 形状进行生成建模的方法。其中一种方法不需要了解相机姿势GT；然而，它仅在纹理几何图元的简单数据集上进行了演示。与我们的提交最相关的是 Tulsiani 等人的并发工作[21]。 这项工作将可微射线一致性公式 [20] 扩展到没有姿势监督的学习。该方法是基于体素的，并使用强化学习和基于 GAN 的先验来处理无监督姿势学习的复杂性。相比之下，**我们使用点云表示，使用集成来预测姿势，并且不需要相机姿势的先验**。

表示问题是使用体积数据进行深度学习的核心。最常用的结构是体素网格——2D 像素化图像的直接 3D 对应物 [5, 23]。这种相似性允许卷积网络架构从 2D 到 3D 的简单转换。 然而，不利的一面是，**体素网格表示会导致内存和计算密集型架构**。这激发了寻找替代选择的动机。现有的解决方案包括八叉树 [17]、网格 [8、26]、基于部件的表示 [19、10]、多视图深度图 [15]、对象骨架 [24] 和点云 [6、11]。我们选择在这项工作中使用点云，**因为它们不像体素网格那样过度完备并且允许有效的网络架构，但同时比基于网格或基于骨架的表示更灵活。**

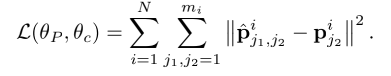
## 3. Single-view Shape and Pose Estimation

我们解决了从对象的单个视角预测对象的三维形状和相机姿势的任务。假设我们给定了 K 个对象的视角数据集 D，第 i 个对象有 个视图：。 这里 表示彩色图像，而 表示来自同一视图的某种模态（轮廓、彩色图像的深度图）的投影。每个视图都可能伴随着相应的相机姿态 ，**但更有趣的情况是当相机姿态未知时**。在本节的其余部分，我们将重点讨论这个更困难的场景。

该模型的概述如下图1所示。假设我们给定了同一对象的两个图像 x1 和 x2。我们使用参数函数逼近器从其中一张图片预测 3D 形状（由点云表示）： ，另一张图片来预测相机姿态： 。在我们的例子中， 和 是**共享大部分参数的卷积网络**。形状和姿势都使用全连接层预测为固定长度的向量。



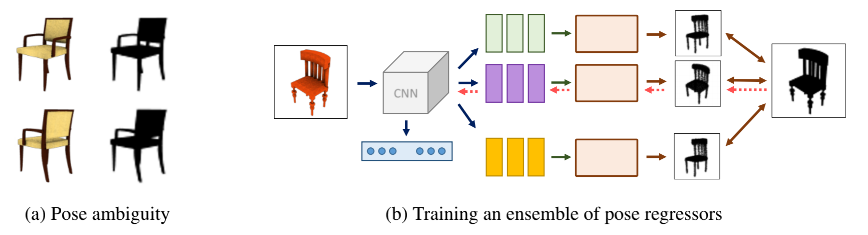
给定预测，我们从预测视图渲染预测形状：，其中 π 表示第 4 节中描述的可微点云渲染器。损失函数是该预测投影与GT的差异。 我们在这项工作中对所有模态都使用标准 MSE，对整个数据集求和：



直观地说，此训练过程要求对于同一对象的所有视角对，预测点云的渲染与提供的GT视图匹配。

**使用蒸馏集成来估计姿势**。

我们发现上述基本实现无法预测准确的姿势。 **这是由局部极小值引起的：姿势预测器收敛于估计从后面看的所有对象，或从前面看的所有对象**。 事实上，基于轮廓，即使是人类也很难区分某些视图，见下图 2 (a)。



为了缓解这个问题，我们引入了 K 个姿势回归器 的集合（见上图 2 (b)），而不是单个姿势回归器 ，并使用“后视损失”（hindsight loss）训练系统 [7, 4]：



这个想法是**每个预测器都学会专注于姿势的子集，并且它们一起涵盖了整个可能值范围**。 不需要特殊措施来确保这种专业化：如果网络架构合适，它会作为随机权重初始化的结果自然出现。也就是说，不同的姿态预测器需要有几个（根据我们的经验至少有 3 个）非共享层。

在训练集成的同时，我们使用集成中的最佳模型作为老师，将其蒸馏为单个回归器。这个最佳模型是根据损失选择的，如上式。 **在测试时，我们丢弃集成并使用蒸馏回归器来估计相机姿势**。 训练学生的损失计算为由四元数表示的两个旋转之间的角差：，其中 Re 表示四元数的实部。我们发现标准 MSE 损失在回归旋转时表现不佳。

**网络架构**。

我们使用具有两个分支的卷积网络来实现形状和姿势预测器。 网络从一个卷积编码器开始，总共有 7 层，其中 4 层的步幅为 2。接下来是 2 个共享的全连接层，之后网络**分成两个分支进行形状和姿势预测**。形状分支是一个带有一个隐藏层的 MLP。 **N个点的点云被预测为一个维度为3N（点位置）或6N（位置和RGB值）的向量**。姿势分支是一个 MLP，每个姿势预测器都有一个共享隐藏层和两个隐藏层。**相机姿态被预测为四元数**。在集成模型中，我们使用 K = 4 个姿势预测器。“学生”模型是另一个具有相同架构的分支。

## 4. Differentiable Point Clouds

我们模型的一个关键组件是**可微点云渲染器 π**。给定一个点云 P 和一个相机位姿 c，它会生成一个视图 p = π(P, c)。点云可能具有与其相关联的信号（例如颜色），在这种情况下，可以将信号投影到视图上。

该方法的高级思想是通过**用密度函数表示点来平滑点云**。形式上，我们假设点云是一组 N 个元组 ，每个元组包括点位置，尺寸参数 和相关的信号 （例如，RGB 颜色）。在我们的大多数实验中，size参数是一个二维向量，包括各向同性高斯的协方差和比例因子。不过，通常 可以表示任意参数分布：例如，在补充材料中，我们展示了具有全协方差矩阵的高斯实验。**尺寸参数可以手动指定，也可以与点位置一起学习**。

整体可微渲染管道如下图 3 所示。出于说明目的，我们在图中显示 2D 到 1D 投影，但实际上我们执行 3D 到 2D 投影。我们首先通过对应于感兴趣的相机位姿 c 的投影变换 Tc 将点的位置变换到标准坐标系：。 变换 Tc 考虑了外部和内部相机参数。我们还计算了变换后的尺寸参数 （确切的变换规则取决于使用的分布）。我们设置相机变换矩阵，使得变换后的投影相当于沿第三轴的正交投影。

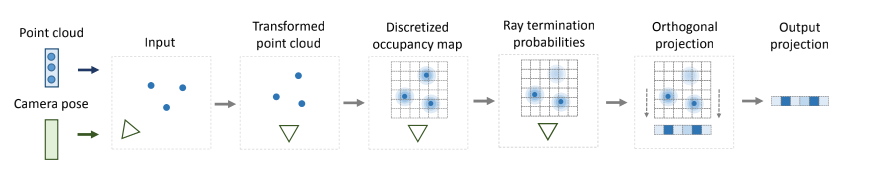
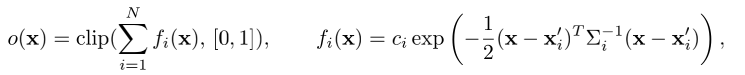


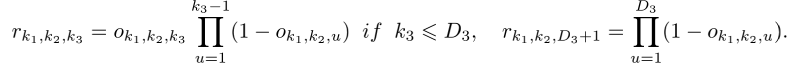
Fig3 点云的可微分渲染。我们出于说明目的展示 2D 到 1D 投影，但实际上我们执行 3D 到 2D 投影。这些点根据相机参数进行变换、平滑和离散化。我们通过光线追踪的形式进行遮挡推理，最后正交投影结果。

为了考虑梯度流，我们用平滑函数 (·) 表示每个点<>。 在这项工作中，我们将 设置为按比例缩放的高斯密度。点云的占用函数是单独的逐点函数的裁剪和：

（3）

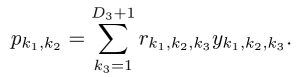
其中 是尺寸参数。 我们将得到的函数离散化为分辨率为 D1×D2×D3 的网格。请注意，第三个索引对应于投影轴，索引 1 是离相机最近的，而 D3 是离相机最远的。

在将生成的体积投影到平面之前，我们需要**确保来自遮挡点的信号不会干扰前景点**。 为此，我们使用类似于 Tulsiani 等人 [20]的可微光线追踪公式进行遮挡推理。我们将占用 o 转换为光线终止概率 r，如下所示：

（4）

直观地说，如果一个cell的占用率 高且所有先前的占用率 { 低，则该小区具有高终止概率 。 额外的背景cell 用于确保终止概率总和为 1。

最终，将体积映射到平面上：

（5）

这里 y 表示被映射的信号，它定义了结果的模态。为了获得轮廓，我们设置 = 1 -。 对于深度图，我们设置 = k3/D3。最后，为了投影与点云相关的信号 y，例如颜色，我们将 y 设置为标准化信号分布的离散版本：

### 4.1 实现细节

从技术上讲，算法中**最复杂的部分是将点云转换为体积**。我们已经尝试了此步骤的两种实现：一种简单灵活（我们将其称为 basic），另一种不太灵活但效率更高（我们将其称为fast）。我们使用标准的 Tensorflow [1] 操作实现了这两个版本。在高层次上，在basic实现中，每个函数 fi 在单独的体积网格上计算，并将结果相加。这允许函数类的选择具有灵活性，但会导致**计算和内存需求随着点数 N 和网格 V 的体积线性增长**，从而导致复杂度 O(NV)。fast版本更优雅地扩展，为 O(N + V )。这是以**对所有函数 fi 使用相同内核为代价的**。fast实现分两步执行操作：首先使用三线性插值将所有点放在网格上，然后使用内核进行卷积。补充资料中提供了更多详细信息。

## 实验

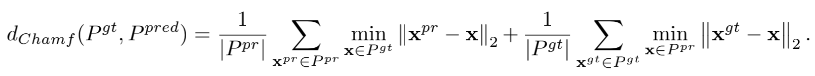
### 5.1 实验设置

**数据集**。

我们对来自 ShapeNet [3] 数据集的 3D 模型进行了实验。我们专注于相关工作中通常使用的 3 个类别：椅子、汽车和飞机。 我们遵循 Tulsiani 等人[20]的训练/测试标准和数据生成程序：将模型分成训练集、验证集和测试集，并使用随机光源位置和随机相机方位角（分别从从 [0◦, 360◦) 和 [-20◦, 40°]）和仰角渲染每个模型的 5 个随机视图。

**验证指标**。

我们使用倒角距离作为我们的主要评估指标，因为**它已被证明与人类对形状相似性的判断有很好的相关性** [16]。 给定地面实况点云 和预测点云 ，距离定义如下：

（6）

上述公式中的两个求和有明确的直观含义。第一个通过计算距离预测点最近的GT点平均多远来评估预测点云的精度。第二个测量预测点云对GT的覆盖范围：平均而言，最近的预测点距GT点有多远。

为了测量位姿误差，我们使用与 Tulsiani 等人[21]相同的指标：准确度（预测姿势在GT 30°以内的样本百分比）和中值误差（以度为单位）。在开始姿势和形状评估之前，我们在验证集中的前 20 个模型上使用迭代最近点 (ICP) 算法将网络学习的规范姿势与数据集中的规范姿势对齐。 补充资料中提供了更多详细信息。

**训练细节**。

我们使用 Adam 优化器 [9] 对网络进行了 600,000 次mini-batch迭代训练。我们使用了 batch=16 （4 个对象的 4 个视角）。我们使用了 0.0001 的固定学习率和标准动量参数。我们在大多数实验中都使用了fast投影，除非另有说明。我们根据用于监督的GT投影的分辨率来改变点云中的点数和投影操作中使用的体积的分辨率。我们使用与训练样本同侧的体积（例如， 体积用于 投影），我们使用 2000 点用于 投影，8000 点用于 投影，16,000 点用于 投影。

在预测密集点云时，我们发现**将 dropout 应用于网络的预测以确保点在形状上的均匀分布很有用**。 仅选择所有预测点的一个子集进行投影和损失计算的Dropout效应。在第 5.2 节和第 5.3 节中报告的实验中，我们从设置概率为90%的非常高的 dropout 开始，并在训练结束时将其线性降低到 0。我们还实施了点尺寸参数的计划，在训练过程中从投影体积大小的 5% 线性减少到 0.3%。在所有实验中学习了点的比例系数。 附录中显示了消融研究。

**计算效率**

基于点云的方法的一个实际优势是它不需要使用基于体素的方法所需的 3D 卷积解码器。**这提高了效率并允许该方法更好地扩展到更高的分辨率**。对于分辨率 32，这些方法的训练时间大致相当。对于 64，我们方法的训练时间大约为 1 天，而基于体素的对应方法为 2.5 天。对于 128，我们的方法的训练时间是 3 天，而基于体素的方法不适合 12Gb 的 GPU 内存和我们的批量大小。

### 5.2 使用已知的姿态估计形状

**和基线的比较结果**

我们首先使用已知的GT相机姿势和基于轮廓的训练，在简单设置中将提议的公式与现有方法进行基准测试。我们与 Yan 等人 [25]的 Perspective Transformer Networks (PTN) 进行了比较，Tulsiani 等人 [20]的可微光线一致性 (DRC)，Lin 等人 [11]的高效点云生成 (EPCG)，以及我们方法的基于体素的对应物。 PTN 和 DRC 仅适用于 的输出体素网格分辨率。 EPCG 使用点云表示，与我们的方法相同。 然而，在最初的工作中，EPCG 仅在不切实际的设置中进行了评估，即每个对象有 100 个随机视角并从 8 个固定视图（立方体的角）进行预训练。 我们在这项工作中使用的更现实的设置中重新训练此方法——每个对象 5 个随机视角。

定量结果如下表 1 所示。我们的基于点云的公式（Ours）在所有情况下都优于其基于体素的公式（Ours-V）。**当提供高分辨率训练信号时，它会有所改善，并且比基于体素的方法更能从中受益**。总体而言，与最佳基线相比，我们的最佳模型（分辨率为 128）将平均误差降低了 30%。一个有趣的观察是，在低分辨率下，PTN 表现非常好，紧随其后的是我们基于点云的公式。 但是请注意，PTN 公式仅适用于从轮廓中学习，不能轻易推广到其他模式。

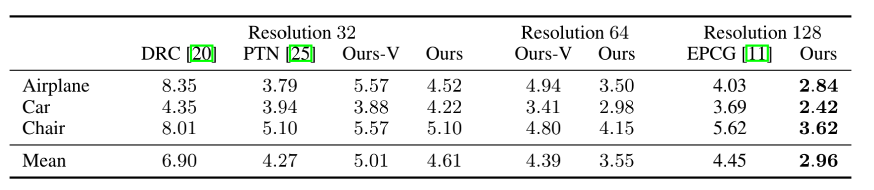


Table1 已知相机位姿下形状预测的定量结果。我们报告归一化点云之间的 Chamfer 距离（乘以 100 的结果）。我们基于点云的方法 (Ours) 优于基于体素的对应方法 (Ours-V)，并且受益于更高分辨率的训练样本。

我们的模型比点云方法 EPCG 实现了 50% 的改进，虽然 EPCG 使用了深度图训练，这是比我们模型使用的轮廓更强的监督信息。仅使用轮廓监督进行训练时，EPCG 的平均误差为 8.20，比我们的模型差 2.7 倍， 我们相信我们的模型更成功，因为我们的渲染过程是可微的 w.r.t。 所有三个点的坐标，而 Lin 等人的方法。 – 仅 w.r.t. 深度。

**带颜色的点云**

**我们的公式支持除轮廓以外的其他监督训练，例如颜色**。在下图 4 中，我们展示了使用我们的方法学习彩色点云的定性结果。 尽管不同视图之间的照明和阴影变化带来了挑战，但该方法能够正确学习彩色点云。**对于具有复杂纹理的对象，预测的颜色会变得模糊**（最后一个示例）。

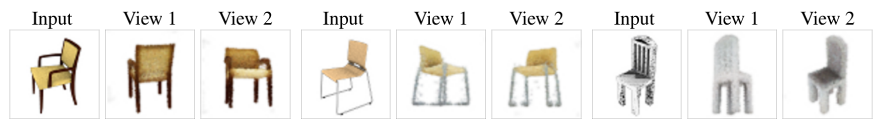


Fig4 学习彩色点云。最好在屏幕上观看。 我们显示输入图像，以及来自其他视角的预测点云的两个渲染结果。大部分的颜色保存得很好，但细节可能会丢失。

**可学习的协方差**

在上面报告的实验中，我们学习了**所有点都具有相同的各向同性协方差矩阵的点云**。我们进行了额外的实验，其中协方差矩阵与点位置共同学习，从而可以更灵活地表示形状。 结果报告在补充材料中。

### 5.3 预测形状和姿势

我们现在放弃在训练期间使用相机姿势GT的不切实际假设，并尝试预测形状和相机姿势。 在这些实验中，我们将 64 像素分辨率的GT用于我们的方法。我们与 Tulsiani 等人[21] 的并发多视图一致性 (MVC) 方法进行了比较,使用作者报告的姿势估计结果和作者提供的预训练模型进行形状评估。

下表 2 中提供了定量结果。尽管无法很好地预测姿势，但我们的 naive 模型（Ours naive）学习了相当准确的形状（比 MVC 差 7%）。我们的解释是，**为相似的投影预测错误的姿势不会显着阻碍形状预测器的训练**。由完整模型（Ours）预测的形状更加精确：比 MVC 准确 28%，仅比姿势GT准确率低 10%（如表 1 所示）。由于多样化的集成公式，姿势预测得到了显着改善。 因此，我们的姿势预测结果在这两个指标上平均略好于 MVC [21]，在中值误差方面甚至比使用姿势GT标签的训练结果更好（如 Tulsiani 等人[21]报道的那样）。

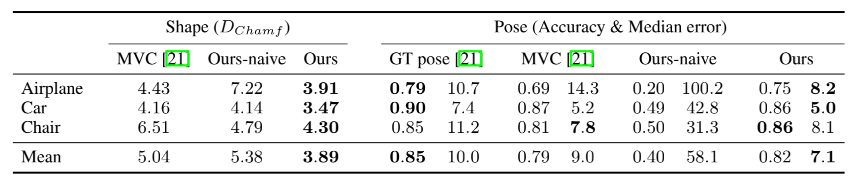


Table2 形状和姿势预测的定量结果。每个指标的最佳结果以粗体突出显示。 我们方法的简单版本可以很好地预测形状，但无法预测准确的姿势。 完整版可以很好地预测形状和姿势

下图 5 显示了使用不同方法生成的形状的定性比较。即使是naive模型（Ours-naive）的结果也优于 MVC [21]。引入姿势集成可以学习更准确的姿势，从而获得更精确的形状。 这些结果证明了点云表示优于基于体素的表示。**点云特别适合表示精细的细节**，例如椅子的细腿。我们还展示了所提出方法的典型失败案例。其中一架飞机旋转了 180 度，因为网络无法找到哪个方向被认为是正确的。其中两把椅子的形状与真实形状有些不同。**这是因为训练问题的复杂性以及可能的过度拟合。然而，这些形状看起来细致而逼真**。

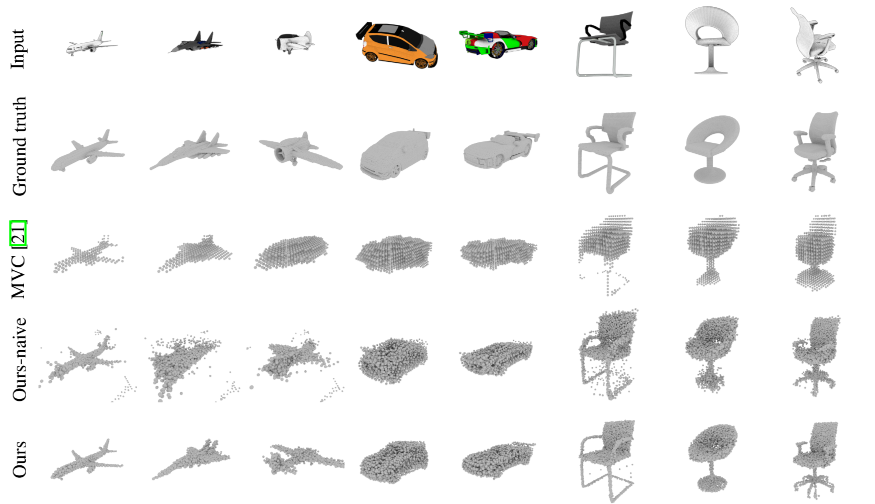


Fig5 形状预测的定性结果。最好在屏幕上观看。 我们使用单个姿势预测器（Ours-naive）的朴素模型预测的形状比 MVC [21] 的更详细。具有一组姿势预测器（我们的）的模型会生成更清晰的形状。点云表示允许保留精细的细节，例如细椅子腿

### 5.4 Discovery of semantic correspondences

除了更高的形状保真度外，**“以物质为中心”的点云表示比“以空间为中心”的体素表示还具有另一个优势：不同预测点云中的点之间存在自然对应关系**。由于我们用全连接层预测点，因此可以预期由相同输出单元生成的不同形状的点具有相似的语义含义。我们凭经验验证了这一假设。我们从椅子类别的验证集中选择两个实例作为模板（如图 6 的左上角所示），并手动注释与特征部分相对应的 3D 关键点位置，例如座位角、腿尖等. 然后，对于每个关键点，我们选择距离关键点一小段距离内的预测云中的所有点，并计算两个模板之间的点索引的交集。（两个对象实例之间的索引交集并不是绝对必要的，但我们发现它可以稍微提高所得到的对应关系的质量。）然后我们在其他几个对象实例上用这些索引可视化点，用不同的颜色突出显示每组点.结果如图 6 所示。正如假设的那样，选定的点往往代表不同对象实例中的相同对象部分。请注意，没有对这个目标施加明确的监督：语义对应会自动出现。我们将此归因于模型学习输出形状空间的规则、平滑表示的隐含能力，这通过对相同的对象部分重用相同的点来促进。

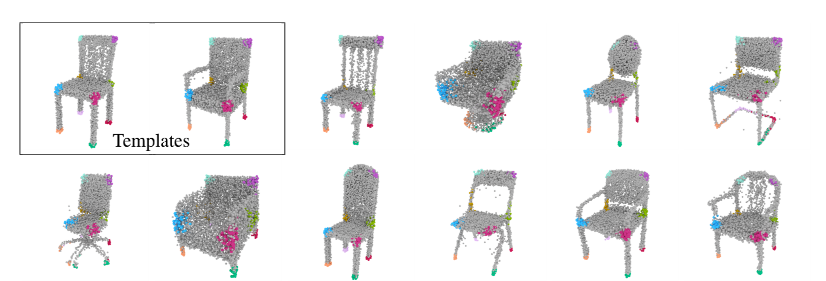


Fig 6 发现语义对应。 相同颜色的点对应于跨不同实例的点云中的相同子集。 这些点是在两个模板实例上选择的（左上角）。

## 结论

我们提出了一种使用点云表示来学习 3D 对象的姿势和形状的方法，该方法仅给出它们的 2D 投影。广泛的验证表明，**点云在效率和准确性方面优于基于体素的表示**。我们的工作为未来的研究开辟了多种途径。首先，我们的投影方法**需要一个明确的体积来执行遮挡推理**。我们认为这只是一个实现细节，未来可能会通过自定义渲染过程放宽。其次，由于该方法不需要准确的相机姿势GT，因此可以应用于从现实世界数据中学习。从彩色图像或视频中学习会特别令人兴奋，但这需要对照明和阴影以及背景进行明确的推理。第三，我们使用非常基本的解码器架构来生成点云，我们相信更高级的架构 [26] 可以提高该方法的效率和准确性。最后，损失是在投影上明确计算的这一事实（与例如 Tulsiani 等人 [20] 形成对比）允许直接将 2D 领域的先进技术（如感知损失和 GAN）应用于学习 3D 表示。

## 总结

**论文解决的问题**：解决了从一组未标记类别的特定图像中学习精确的三维形状和相机姿态的问题。

**论文的方法概述**：论文训练了一个卷积网络，通过最小化重投影误差来预测单个图像的形状和姿态：给定一个物体的多个视角，预测的形状到预测相机姿态的投影应该与提供的视图相匹配。

**论文克服了三个问题**：

1. 估计形状和姿势是一个鸡和蛋的问题：没有一个好的形状估计就不可能学习准确的姿势，反之亦然，准确的姿态估计是学习形状的必要条件。这个问题**通过形状和姿势预测器的联合优化解决**
2. **位姿估计容易出现由模糊性引起的局部极小值**：一个目标可能从两个角度看起来很相似，如果网络在所有情况下都收敛到只预测其中一个角度，那么它将无法学习预测另一个角度。这个问题会导致姿态估计中的严重误差。为了解决这个问题，作者**训练了不同的姿态预测器集合并将其提炼为单个学生模型（student model）**
3. **在没有明确的3D监督的情况下学习点云信息**。论文提出了一个可微投影算子，该算子在给定点集和相机姿态的情况下，生成一个二维投影（轮廓、彩色图像或深度图），作者将该算子称为“可微点云”

**现有方法的不足**：现有方法需要相机位姿的精确数据作为label，且预测形状由于基于voxel，其分辨率非常低。

**本文的贡献**：

1. 本文提出了一种不需要标签数据、利用点云表示的方法来学习仅给出二维投影的三维物体的姿态和形状。
2. 实验表明，点云在效率和准确性方面优于 voxel-based 的表示，可以生成高质量的形状模型
3. 相比于使用真值训练的相机位姿估计网络，性能只有很小的下降。
4. 重投影误差：该方法不需要准确的相机姿势 GT，因此可以应用于从真实数据中学习
5. 使用了位姿估计器集合来解决位姿歧义问题

## 相关文章

1. <https://blog.csdn.net/phy12321/article/details/106035033>