arXiv:2301.11497v2 [cs.CV] 1 Jun 2023

D²CSG：利用双重形状补集和丢弃进行紧凑 CSG 树的无监督学习

Fenggen Yu[1] **Qimin Chen**[1] **Maham Tanveer**[1]**Ali Mahdavi Amiri**[1] **Hao Zhang**[1]西蒙弗雷泽大学

摘要

摘要：我们提出了 D²CSG，这是一种由两个双重且互补的网络分支组成的神经模型，并通过丢弃实现对紧凑的构造实体几何（CSG）三维 CAD 形状表示的无监督学习。我们的网络通过按固定顺序组装二次曲面基元来重建三维形状，两个分支均生成基元交集的并集或其逆。D²CSG 与此前所有神经 CSG 模型的一个关键区别在于，它专门设有一个残差分支，用于组装可能复杂的形状补集；该补集会从由覆盖分支建模的整体形状中减去。借助形状补集，我们的网络在理论上具有完备的表达能力，而权重丢弃则通过移除冗余基元进一步提高了 CSG 树的紧凑性。我们通过定量和定性实验均证明，D²CSG 能够生成质量更高、基元更自然的紧凑 CSG 重建，优于所有现有替代方法，尤其是在复杂及高亏格 CAD 形状上。

1 引言

CAD 形状在几何深度学习的发展中发挥着核心作用，迄今为止，大多数神经模型都在 ModelNet [48],、ShapeNet [2], 和 PartNet [29] 等数据集上训练，用于分类、重建和生成任务。这些形状集合都具有明确定义的类别或类标签，而且数据驱动方法的有效性往往取决于能否充分学习特定类别的形状特征。近年来，ABC [19] 和 Fusion360 [45] 等 CAD 零件和装配体数据集的出现，推动了对不依赖类别标签的形状表示学习的需求，使其不再依赖任何类别先验。一个典型例子是，ABC 数据集不提供任何类别标签；与此同时，CAD 形状所呈现的丰富拓扑多样性，也给后续的表示学习带来了挑战。

构造实体几何（CSG）是一种经典的 CAD 表示方法，它将三维形状建模为实体基元（如长方体、圆柱体等）通过布尔运算（包括并集、交集和差集）递归组合而成的结构。值得特别指出的是，差集运算在建模孔洞和高形状亏格方面不可或缺，而这类情况在 CAD 中十分常见。近年来，利用 CSG 进行三维表示学习受到越来越多的关注， [9, 46, 37, 36, 17, 54, 6,3], 旨在提升所学习模型的通用性、紧凑性和重建质量。

就基元数量（这是构造实体几何树紧凑性的直接指标）和重建质量而言，CAPRI-Net [54] 代表了当前最先进的水平。然而，它并非通用神经模型，例如，它无法表示那些装配时需要执行嵌套差集运算的 CAD 形状（即，需要减去一个自身也需要通过基元差集运算构建的部分，例如，参见图 2 中的 CAD 模型）。由于 CAPRI-Net 中（单个）差集运算的两个操作数只能表示交集和并集，其网络无法为具有复杂凹陷和拓扑细节的较复杂 CAD 形状生成自然且紧凑的 CSG 装配体；参见图 1。

预印本。正在评审中。

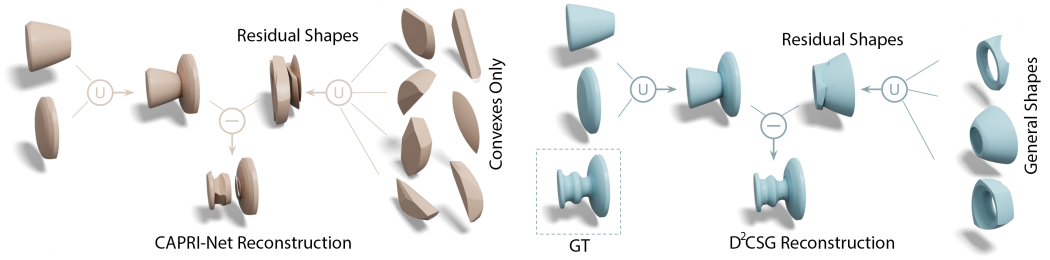


图 1：比较我们提出的网络 D²CSG 与当前最先进方法 CAPRI-Net 的 CSG 树和形状重建结果。要重现真实形状，自然的 CSG 组合需要进行差集运算，涉及复杂的残差形状，而 D²CSG 能够以紧凑的形式（仅三个中间通用形状）和良好的质量预测该形状。CAPRI-Net 只能通过凸体的并集来构建它，因此需要不必要的大量基元，并导致重建效果不佳。

本文提出 D²CSG，这是一种由两个双重且互补分支构成的新型神经网络，用于无监督学习紧凑的3D CAD 形状的 CSG 树表示。如图 2 所示，我们的网络遵循固定顺序的CSG 装配方式，与大多数以往的无监督 CSG 表示学习模型相同 [54, 6, 3]。然而，与所有这些模型的一个关键区别在于残差分支，该分支专门用于装配可能较复杂的补集或残差形状。随后，从由覆盖分支学习得到的整体形状中减去形状补集。 从架构上看，这两个分支完全相同，均由二次曲面基元和基元逆构建交集的并集，但它们由独立的网络参数建模，并作用于不同的基元集合。由于最终差集运算的两个操作数都能够学习通用的 CAD 形状组合，与当前最先进的方法相比，我们的网络能够以更紧凑的方式表示复杂且高亏格的 CAD 形状，并取得更高的重建质量，如图 1 所示。为了进一步提高紧凑性，我们根据丢弃重要性度量，采用针对网络权重的策略来移除冗余基元。重要性度量。

鉴于 CAD 形状之间存在显著的结构多样性，无监督学习 CSG 树颇具挑战，因此我们的网络并非旨在对形状集合学习统一模型。相反，它通过优化由二次曲面基元构成的紧凑 CSG 组合来近似给定的 3D CAD 形状，从而对该形状过拟合 。由于基元的数量、选择和组合方式均未知，导致搜索空间十分复杂，因此这一学习问题仍然具有挑战性。

与 CAPRI-Net 相比，我们的方法具有可证明的通用性，这意味着任何构造实体几何树都可以转换为等价的 D²CSG 表示。我们的双分支网络完全可微，仅需使用神经隐式模型的常规占用损失即可进行端到端训练 [4,3, 54]。我们通过定量和定性实验表明，在 ABC [19] 或 ShapeNet [2], 上训练时，我们的网络生成的构造实体几何重建质量更高、树结构更自然，且质量与紧凑性的权衡优于所有现有替代方法 [3, 6, 54]。

2 相关工作

一般而言，三维形状可以表示为一组组合在一起的基本体或部件。针对点云的基本体拟合已得到广泛研究 [25, 23, 21]。在形状抽象方面，已有研究采用长方体[40][51] 和超二次曲面 [34] ，而 3D 高斯则用于模板拟合 [12]。研究者还利用卷积递归自编码器，从单张图像估计物体部件及其关系，并采用长方体 [30]。此外，研究者也学习了更复杂的子形状，用于形状组合，例如基本 3D 结构 [8], 、隐式凸体 [7], 和神经星形组件 [18], ，以及以可学习参数化面片 [37], 、移动或可变形基本体 [26, 55, 52, 33], 、点云 [24], 或部件感知潜空间 [10]形式表示的部件 。然而，这些技术都没有直接解决为给定的三维形状重建构造实体几何树的问题。

深度 CAD 模型。由于 CAD 模型具有尖锐特征和复杂的拓扑结构，合成和编辑 CAD 模型颇具挑战。基于学习的形状程序通过提供易于使用的编辑工具来执行这些任务 [39, 11, 16, 1]。边界表示（边界表示）常用于 CAD 形状建模，已有研究尝试根据输入网格或点云对这类表示进行逆向工程 [49]。例如，BRepNet [20],、UV-Net [14],

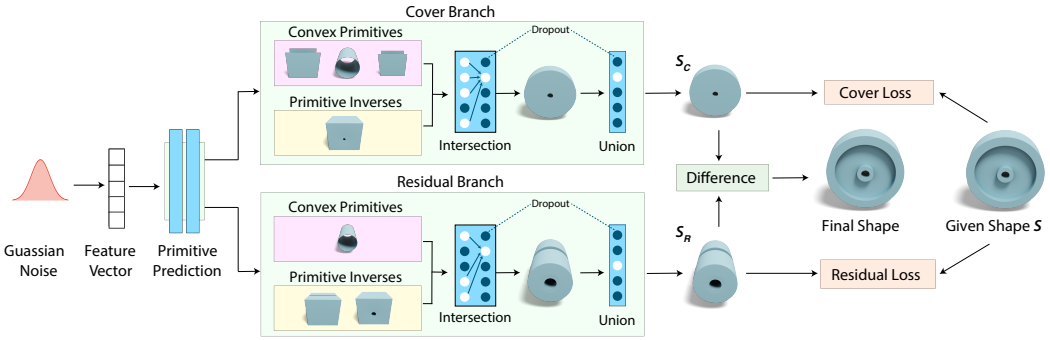


图 2： 对于给定的三维形状 S（显示在最右侧），我们的网络 D^{2}CSG 通过优化占据损失进行训练，同时优化网络参数和特征编码，以重建 S。网络参数定义了由一组二次曲面基元及其逆构成的 CSG 组合。该组合由两个相同的分支构成：覆盖分支（上方）生成形状 S_C，残差分支（下方）生成形状 S_R。通过交集和并集运算得到覆盖形状 S_C 和残差形状 S_R，并优化各自的占据损失后，通过差集运算得到重建形状。此外，还对交集层和并集层中的参数应用丢弃，以移除冗余基元和中间形状。

和 SBGCN [15]提供了可处理边界表示及其拓扑信息的网络，并通过消息传递进行计算。NH-Rep [13] 提出了若干手工设计的规则，用于从边界表示推断构造实体几何树。然而，其构造实体几何树的叶节点并非参数化基本体，而是隐式曲面。基于拉伸的技术 [47, 50, 35] 也已用于基于深度学习的 CAD 形状生成，但拉伸无法提供通用的形状表示，随之采用的监督学习方法 [44, 47] 也仅适用于特定数据集。

CSG学习。 学习 CSG 表示，例如基元组合 [43] 和草图分析 [31, 22], 已成为几何深度学习中的一个新兴课题。尽管大多数方法采用监督学习，例如 CSG-Net [36],SPFN [23],ParseNet [37], DeepCAD [46],和 Point2Cyl [41],，近年来也出现了一些无监督 CSG 树重建的尝试，尤其是在类别无关设定下，采用神经隐式表示 [4, 27, 32, 3]。 InverseCSG [9]是另一种相关方法，它首先使用 RANSAC 检测输入形状表面上的一组基元，然后通过搜索找到能够重建目标形状的 CSG序列。然而，RANSAC 需要细致地调整参数，且最初检测到的基元可能含有噪声。此外，CSG序列的搜索空间十分庞大，导致搜索过程耗时很长。UCSG-Net [17] 通过学习以任意组合顺序重建构造实体几何树。由于顺序灵活，该学习任务难度较高，但将基元限制为仅包含立方体和球体后便可实现。通过学习固定顺序的 组合体，在重建质量和构造实体几何树的紧凑性方面取得了更好的效果，其中包括 D^{2}CSG。

BSP-Net [3] 学习平面基元，并通过交集组合其半空间以获得凸体，随后执行并集操作，从而重建凹形状。CAPRI-Net [54] 通过添加二次曲面基元，并在基元并集之后加入差集运算，对 BSP-Net 进行了扩展。与 D^{2}CSG 类似，CAPRI-Net [54]也尝试通过预测一组基元来执行最终形状差集运算：这些基元按照固定顺序依次进行交集、并集和差集运算，以获得目标形状。然而，它将所有基元限制为凸体，且其序列并不通用，无法支持所有形状。为实现通用性，D^{2}CSG 中的基元集合既包含凸基元，也包含补基元（见第 3.1 节）。这样便能在 CSG 序列的初始阶段执行差集运算。关于 D^{2}CSG 通用性的形式化证明，请参见补充材料。CSG-Stump [6] 同样遵循固定的 CSG 组合顺序，并引入求逆层来建模形状补集。理论上，补集运算有助于使其 CSG 重建具备通用性，但求逆层不可微，而且差集运算只能应用于基本基元，这会严重损害重建的紧凑性和质量。 ExtrudeNet [35] 遵循与 CSG-Stump 相同的 CSG 顺序，并将基本三维基元改为拉伸体，以简化形状编辑，但这牺牲了对类球形状的泛化能力。

过拟合模型。 在压缩 [5], 重建 [42], 和细节层次建模 [38] 等应用中，为单个形状学习过拟合模型并不少见，因为这样可以获得高质量的结果。 NeRF 的底层几何本质上是对形状/场景进行过拟合（即固定），尽管 NeRF 的主要任务是新视角合成 [53, 28]。遵循类似原理，为了复现精细的几何细节，D²CSG 通过优化一个小型神经网络以及随机初始化的特征编码，为给定对象构建构造实体几何树（图 2）。由于 CAD 形状在结构和拓扑方面具有多样性，我们发现学习 CAD 形状的先验并不实用，也难以泛化，因此采用了这种优化/过拟合流程。

3 方法

构造实体几何表示通过对一组基本体（构造实体几何树的叶节点）执行操作，生成中间形状，再将其合并以形成最终形状。在这项工作中，D²CSG 的输入是一个从高斯分布中采样得到的特征向量，该向量与网络参数一起进行优化，以重建给定的三维形状 S；输出是一组经过优化以拟合 S 的占据值（类似于 DeepSDF 中的自动解码器 [32]）。我们不将形状编码为潜在编码，因为这样做无助于提高收敛精度或速度（见补充材料）。我们的设计效率很高，因为 D²CSG 使用轻量级网络，能够快速收敛到每个形状。从特征向量出发，我们将其输入基元预测网络，并生成两个存储基本体参数的矩阵。每个矩阵用于确定一组在形状附近采样的查询点的近似符号距离（ASD）。这两组 ASD 值分别输入覆盖分支和残差分支，每个分支都包含一个交集层和一个并集层。覆盖分支（图 2 上）和残差分支（图 2 下）生成点占据值，以指示一个点是否位于 S_C 内，而 S_R则表示 S_C 内 S_R。二者之差构成最终形状。我们采用 CAPRI-Net 的多阶段训练方法 [54] 以实现精确的构造实体几何重建。此外，由于以往的 CSG 学习方法没有使用显式的紧凑性约束，预测的 CSG 序列中存在许多冗余基本体和中间形状。我们对每个 CSG 层的权重引入了一种新颖的丢弃操作，以迭代地减少冗余并提高紧凑性。

3.1 基元表示

CAPRI-Net 中的隐式二次曲面方程 [54] 可以涵盖常用的 凸 基元，例如平面、球体等。然而，其固定的构造实体几何顺序（交集、并集和差集）并不适用于组合凸体以生成任意形状；证明见补充材料。借助 D²CSG，我们引入了一种更通用的基元形式，还可以涵盖 逆 凸基元，从而使差集运算既可以作为学习到的 CSG序列中的最后一个运算，也可以作为第一个运算。这是我们的方法与 CAPRI-Net 之间的关键区别之一 [54], 后者只能通过交集层生成凸形状。 我们的基元预测网络（一个多层感知机）接收一个大小为 256 的编码，并输出两个 独立的 矩阵 $\mathbf{P}_C \in \mathbb{R}^{p\times 7}$ 和 $\mathbf{P}_R \in \mathbb{R}^{p\times 7}$ （分别用于覆盖分支和残差分支），每个矩阵都包含 p 个基元的参数（见图 2）。下一节将讨论分离 $\mathbf{P}_C$ 和 $\mathbf{P}_R$ 的重要性。 $\mathbf{P}_C$ 和 $\mathbf{P}_R$ 中的一半基元由二次曲面方程表示： $|a|x^2 + |b|y^2 + |c|z^2 + dx + ey + fz + g = 0.$

Com互补基本体。 在 D²CSG 中， $\mathbf{P}_C$ 和 $\mathbf{P}_R$ 中另一半的基本体是反
凸基础体个系数为负：

$$-|a|x^2 - |b|y^2 - |c|z^2 + dx + ey + fz + g = 0. \tag{1}$$

我们避免使用没有任何非负约束的一般二次曲面形式，因为它会产生 CAD 设计中不常见的复杂形状。对于重建，我们在形状表面附近采样 n 个点，并按照与 CAPRI-Net [54]类似的方式计算这些点到所有基本体的 ASD。对于每个点 $\mathbf{q}_j = (x_j, y_j, z_j)$，其 ASD 记录在矩阵 $\mathbf{D} \in \mathbb{R}^{n\times p}$ 中：$\mathbf{D}_C(j,:) = \mathbf{Q}(j,:)\mathbf{P}_C^T$，以及 $\mathbf{D}_R(j,:) = \mathbf{Q}(j,:)\mathbf{P}_R^T$，其中 $\mathbf{Q}(j,:) = (x_j^2, y_j^2, z_j^2, x_j, y_j, z_j, 1)$ 是 Q 的第 j 行。

3.2 双重CSG分支

以往的构造实体几何学习方法采用固定的构造实体几何顺序，并不像 CAPRI-Net [54] 和 D²CSG 那样将差集运算作为最后一步。将差集运算作为最后一步

表1：多阶段训练的设置。

阶段	交集（$\mathbf{T}$）	并集（$\mathbf{W}$）	差集运算（$\mathbf{a}$）	丢弃	Loss
1	浮点数	浮点数	$\mathbf{a}^+$	-	$L^+_{rec} + L_{\mathbf{T}} + L_{\mathbf{W}}$
2	浮点数	-	$\mathbf{a}^*$	-	$L^*_{rec} + L_{\mathbf{T}}$
3	二进制	二进制	$\mathbf{a}^*$	✓	L^*_{rec}

有助于重建出具有复杂凹形的形状 [54]。然而，CAPRI-Net 对执行最终减法运算的两个形状共用一个基元集合。这两个操作数之间的共享会对最终结果产生不利影响。为了通过所有基元反向传播梯度，在训练早期阶段，交集层中的选择矩阵 $\mathbf{T}$ 中的值为浮点数（公式 2 和表1）。因此，许多基元会同时属于覆盖形状和残差形状，即使它们在 $\mathbf{T}$ 中对应的权重可能很小。因此，残差形状所使用的基元可能会影响覆盖形状，导致细节被去除，反之亦然。然而，D^{2}CSG 中的双分支使覆盖形状和残差形状能够使用不同的基元，从而更好地拟合各自的目标形状。

现在，我们简要介绍双分支构造立体几何，并采用与 BSP-Net [3] 类似的符号表示。我们将 ASD矩阵 $\mathbf{D}_C$ 输入覆盖分支，将 $\mathbf{D}_R$ 输入残差分支，并输出向量 $\mathbf{a}_C$ 和$\mathbf{a}_R$，分别表示查询点是否位于覆盖形状 S_C 和残差形状 S_R的内部或外部。每个分支都包含一个从 BSP-Net [3]沿用的交集层和并集层。

训练过程中，交集层的输入是两个 ASD 矩阵 $\mathbf{D}_R \in \mathbb{R}^{n\times p}$ 和 $\mathbf{D}_C \in \mathbb{R}^{n\times p}$。通过两个可学习矩阵 $\mathbf{T}_C \in \mathbb{R}^{p\times c}$和 $\mathbf{T}_R \in \mathbb{R}^{p\times c}$来选择参与构成交叠形状的基本体，其中 c 是中间形状的数量。我们可以通过交集层得到**交集** $\in \mathbb{R}^{n\times c}$，并且仅当 **交集**$(j,i) = 0$时，查询点 $\mathbf{q}^j$ 才位于中间形状 i内部（**交集**$_R$ 相同，仅下标 R不同）：

$$\mathbf{Con}_C = \text{relu}(\mathbf{D}_C)\mathbf{T}_C \quad \begin{cases} 0 & \text{in,} \\ > 0 & \text{out.} \end{cases} \tag{2}$$

然后，通过两个并集层将交集运算得到的所有形状组合起来，从而求得覆盖形状 S_C 和残差形状 S_R。组合形状的内外指示值存储在向量 $\mathbf{a}_R$ 和 $\mathbf{a}_C \in \mathbb{R}^{n\times 1}$ 中，用于指示一个点是否位于覆盖形状和残差形状内部。我们以多阶段方式计算 $\mathbf{a}_C$ 和$\mathbf{a}_R$ （$\mathbf{a}^+$ 和 $\mathbf{a}^*$ 分别对应早期阶段和最终阶段），具体如下：[54]。具体而言，$\mathbf{a}^*_C$ 和 $\mathbf{a}^*_R$ 通过分别求 **交集**$_C$ 和 **交集**$_R$每一行的最小值获得：

$$\mathbf{a}^*_C(j) = \min_{1\le i\le c}(\mathbf{Con}_C(j,i)) \quad \begin{cases} 0 & \text{in,} \\ > 0 & \text{out.} \end{cases} \tag{3}$$

由于在最小值运算中，梯度只能反向传播到最小值，因此我们还在训练早期引入$\mathbf{a}^+_C$，以通过以下公式促进学习：

$$\mathbf{a}^+_C(j) = \quad \mathscr{C}(\sum_{1\le i\le c} \mathbf{W}_C(i)\mathscr{C}(1-\mathbf{Con}_C(j,i))) \begin{cases} 1 & \approx \text{in,} \\ < 1 & \approx \text{out.} \end{cases} \tag{4}$$

$\mathbf{W}_C \in \mathbb{R}^c$ 是一个学习得到的加权向量，$\mathscr{C}$是对 $[0,1]$进行裁剪运算。$\mathbf{a}^+_R$ 的定义类似。

3.3 训练中的损失函数与丢弃

我们的训练方式与 CAPRI-Net [54] 类似，采用分阶段训练，以促进构造实体几何层权重学习中的梯度传播。训练早期阶段（0 - 1）的训练方式和损失函数与 CAPRI-Net 相同，而阶段2由于引入了一种新颖的丢弃机制，是我们训练中的关键部分，因此有所不同。表1概述了训练的各个阶段及其损失函数，更多细节将在下文和补充材料中介绍。

阶段0。在阶段0，**交集层中的 T**和**并集层中的 W**均为浮点数。损失函数为：
$L^+ = L^+_{rec} + L_{\mathbf{T}} + L_{\mathbf{W}}$。 L^+_{rec} 是应用于$\mathbf{a}^+_C$ 和$\mathbf{a}^+_R$ 的重建损失，并会迫使

覆盖形状与残差形状相减后的结果接近输入形状。 $L_{\mathbf{T}}$和 $L_{\mathbf{W}}$ 是应用于交集层和并集层权重的损失，用于使$\mathbf{T}$中的元素保持在 $[0,1]$ 内，并使$\mathbf{W}$ 中的元素接近 1 [3]。请注意，双分支构造立体几何各自拥有独立的网络权重：$\mathbf{T} = [\mathbf{T}_C、\mathbf{T}_R]$ 和$\mathbf{W} = [\mathbf{W}_C、\mathbf{W}_R]$，并且它们分别训练。有关损失的更多细节，请参阅补充材料。

阶段1。此处，向量$\mathbf{a}^+$ 被替换为$\mathbf{a}^*$，以消除对$\mathbf{W}$ （来自阶段0）的使用。这里采用$\min$ 作为覆盖形状和残差形状的并集操作，并限制梯度反向传播仅作用于经过并集操作的形状。损失函数为$L^* = L^*_{rec} + L_T$。 L^*_{rec} 是应用于$\mathbf{a}^*_C$ 和$\mathbf{a}^*_R$， L_T 与阶段0相同。

阶段2。在阶段2中， T 中的条目 t 通过给定阈值 $t_{Binary} = (t > \eta)?1 : 0$ 量化为二值，其中 $\eta = 0.01$。这样，网络将在基本体上执行可解释的交集操作。由于 t_{Binary} 值不可微，损失项 L_T不再使用，因此损失函数简化为仅包含 L^*_{rec} （见表1）。

阶段2中的丢弃。 若没有明确的紧凑性约束或损失，构造实体几何重建中的冗余便不可避免。为缓解这一问题，我们提出两个关键观察：首先，在阶段2学习到的构造实体几何序列完全可解释，因此我们可以追踪生成最终形状所涉及的基本体和 CSG 操作；其次，改变基本体或 CSG 操作会相应地改变最终形状的隐式场。因此，如果移除某些基本体和中间形状不会显著影响最终结果，我们便可以识别出不太重要的基本体和中间形状。因此，我们引入重要性度量 $\Delta\mathbf{S}$，用于衡量移除基本体或中间形状对结果的影响。首先，我们将最终形状的隐式场$\mathbf{s}^*$ 定义如下：

$$\mathbf{s}^*(j) = \max(\mathbf{a}^*_C(j), \alpha - \mathbf{a}^*_R(j)) \begin{cases} 0 & \text{in}, \\ > 0 & \text{out}, \end{cases} \tag{5}$$

其中，$\mathbf{s}^*(j)$ 表示查询点 $\mathbf{q}_j$ 位于重建形状内部还是外部。需要注意的是， α 应当是一个较小的正数（≈ 0.2），因为当点位于形状内部时，$\mathbf{s}^*$ 会趋近于 0。随后，重要性度量 $\Delta\mathbf{S}$ 定义为：

$$\Delta\mathbf{S} = \sum_{j=1}^{n} \Delta\mathbb{1}(\mathbf{s}^*(j) < \frac{\alpha}{2}). \tag{6}$$

此处，$\mathbf{s}^*(j) < \frac{\alpha}{2}$ 将隐式场值$\mathbf{s}^*(j)$从浮点数转换为布尔值，其中内部值为 1，外部值为 0。函数$\mathbb{1}$ 将布尔值映射为整数，以用于求和操作； Δ则捕捉移除基本体或中间形状时的值差异。

请注意，阶段0中， $L_{\mathbf{W}}$ 将并集层参数$\mathbf{W}$ 设为接近 1。阶段2中，每经过 4,000 次迭代，如果从CSG序列中移除中间形状 i 后，$\mathbf{S}$ Δ小于阈值 σ，我们就会将$\mathbf{W}_i = 0$，以此丢弃中间形状 i 。因此，公式（5）将修改为纳入并集层中更新后的权重：

$$\mathbf{a}^*_C(j) = \min_{1 \le i \le c} (\mathbf{Con}_C(j,i) + (1 - \mathbf{W}_i) * \theta) \quad \begin{cases} 0 & \text{in}, \\ > 0 & \text{out}. \end{cases} \tag{7}$$

这样，只要 θ 足够大，被移除的形状 i 就不会影响$\mathbf{a}^*$。在我们的实验中，我们发现 θ 只需大于 α，被移除的形状 i 就不会影响$\mathbf{s}^*$。因此，我们在所有实验中均将 θ 设为 100。此外，如果从 CSG序列中移除基本体 k 后$\mathbf{S}$ Δ低于 σ，我们还会通过设置$T_{k,:} = 0$，对交集层中的参数应用丢弃。我们在实验中将 σ设为 3，并在补充材料中考察 σ 的影响。

4 结果

在本节将介绍的实验中，为了支持复杂形状，我们为每个分支设置$p = 512$个最大基本体数量和 $c = 32$ 个最大交集数量。潜在编码的大小为 256，并使用两层多层感知机根据输入特征编码预测基本体的参数。我们通过优化潜在编码、基元预测网络、交集层和并集层，针对每个形状训练 D^{2}CSG。为了将 D^{2}CSG 与先前方法进行比较（这些方法都需要在测试时额外进行耗时的优化才能达到

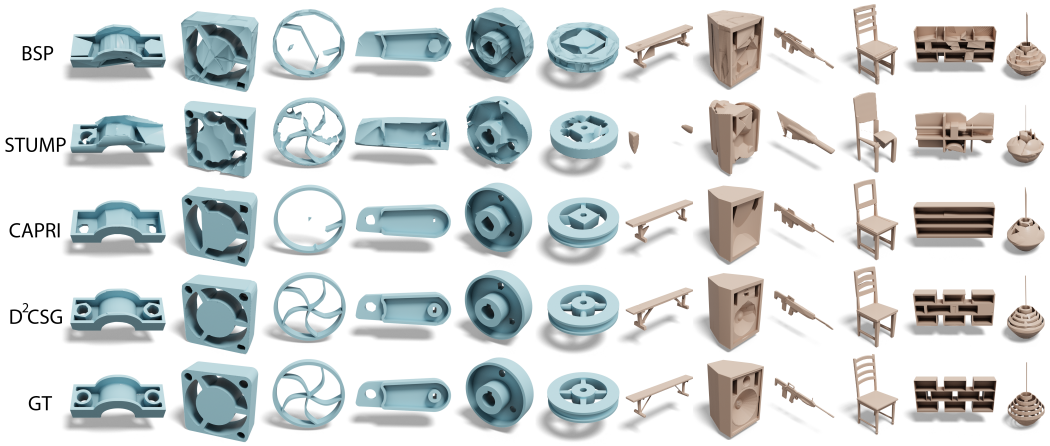


图 3：比较 ABC 数据集（第 1–6 列）和 ShapeNet（第 7–12 列）中基于三维网格的 CSG 表示学习与重建。放大后查看效果最佳。

令人满意的结果（例如，CSG-Stump 每个形状需要 30 分钟），我们随机选取了一组规模适中的形状作为测试集进行评估：从 ABC 中选取 500 个形状，并从 ShapeNet 的 13 个类别中各选取 50 个形状（共 650 个形状）。此外，我们确保从 ABC 选取的形状中有 80% 的亏格大于二且顶点数超过 10K，以涵盖复杂结构。所有实验均使用 Nvidia GeForce RTX 2080 Ti GPU 完成。

4.1 网格到 CSG 表示

给定一个三维形状，任务是为其学习准确且紧凑的 CSG 表示。为此，我们首先在形状表面周围采样 24,576 个点（即距离不超过 $1/64$），并在三维空间中随机采样 4,096 个点。随后将全部 28,672 个点缩放到范围 $[-0.5, 0.5]$，并利用这些点及其占据值优化网络。

我们将 D²CSG 与 InverseCSG [9] （I-CSG）、BSP-Net [3], CSG-Stump [6] 和 CAPRI-Net [54], 进行比较，这些方法均输出结构化的参数化基元。为公平比较，我们使用相同的迭代次数优化所有这些网络。在优化前，BSP-Net、CSG-Stump 和 CAPRI-Net 均在 CAPRI-Net 提供的训练集上进行预训练，以获得更好的初始化。需要注意的是，CSG-Stump 对 ABC 中的形状（存在形状差异）和 ShapeNet 中的形状（不存在形状差异）采用不同的网络设置，因此我们在比较中遵循相同的设置。对于 InverseCSG，我们采用其 CAD 形状数据集中大多数形状所使用的 RANSAC 参数。对于每个形状重建，BSP-Net 约需 15 分钟，I-CSG 约需 17 分钟，CSG-Stump 约需 30 分钟，CAPRI-Net 约需 3 分钟才能收敛。相比之下，D²CSG 的训练过程在每个阶段运行 12,000 次迭代，每次形状重建约需 5 分钟。

评估指标。用于形状重建的定量指标包括对称倒角距离（CD）、法线一致性（NC）和边缘倒角距离 [3] （ECD）。对于 ECD，我们将法线叉积的阈值设为 0.1，以提取靠近边缘的点。倒角距离和 ECD 在表面上的 8K 个采样点上计算，并乘以 1,000。此外，我们通过比较基本体数量 # P 来评估形状的紧凑性，因为所有基于 CSG 的建模方法都会预测一些通过交集操作组合起来的基本体。

评估与比较。我们在图 3 中展示了 ABC 数据集和 ShapeNet 数据集中的代表性示例的视觉比较。由于 InverseCSG 仅接受高质量 CAD 网格作为其基元检测算法的输入，且无法处理 ShapeNet，因此我们仅在 ABC 数据集上对其进行比较；更多视觉结果见补充材料。我们的方法始终能够重建出更准确的形状，并保留几何细节和凹陷。InverseCSG 中基于 RANSAC 的基元检测算法很容易产生带噪声且不准确的基元，导致 CSG 建模后出现许多退化平面。BSP-Net 只是组合凸形状来拟合目标形状，因此在不使用差集运算的情况下，得到的结果紧凑性较差。CSG-Stump 倾向于使用相当多的差集运算来重建形状。这也会导致形状表面被许多冗余基元雕刻（即缺乏紧凑性）。此外，由于

表 2：比较在 ABC 和 ShapeNet 中从三维网格进行构造实体几何表征学习的方法。

	ABC					ShapeNet			
方法	I-CSG	BSP	STUMP	CAPRI	Ours	BSP	STUMP	CAPRI	Ours
CD ↓	0.576	0.115	0.383	0.177	**0.069**	0.164	2.214	0.124	**0.119**
NC ↑	0.877	0.921	0.850	0.903	**0.928**	0.882	0.794	**0.890**	0.887
ECD ↓	6.330	4.047	8.881	3.990	**3.091**	3.899	6.101	2.035	**1.722**
#P ↓	43.62	359.38	83.42	66.26	**28.62**	694.21	228.58	50.94	**43.78**

表 3：对 D^{2}CSG 关键组件的消融研究：互补基元（CP）、双分支（DB）和丢弃（DO）；评估三项质量指标、三项紧凑度指标，以及由构造实体几何树引导的分解所产生的构造实体几何基元数量（#P）、中间形状数量（#IS）和表面片段数量（#Seg）。最佳结果以粗体标示，次佳结果以蓝色标示。

行 ID	CP	DB	DO	CD ↓	法向量一致性 ↑	扩展倒角距离 ↓	#P ↓	#IS ↓	#Seg ↓
1	-	-	-	0.177	0.903	3.99	66	8.3	82
2	✓	-	-	0.073	0.935	3.12	38	5.8	55
3	✓	-	✓	0.088	0.926	3.48	**27**	**5.3**	**40**
4	✓	✓	-	**0.069**	**0.936**	**2.98**	53	6.8	57
5	✓	✓	✓	**0.069**	0.928	3.09	29	5.7	42

表 4：比较从 ABC 和 ShapeNet 的三维点云中学习构造实体几何表征。

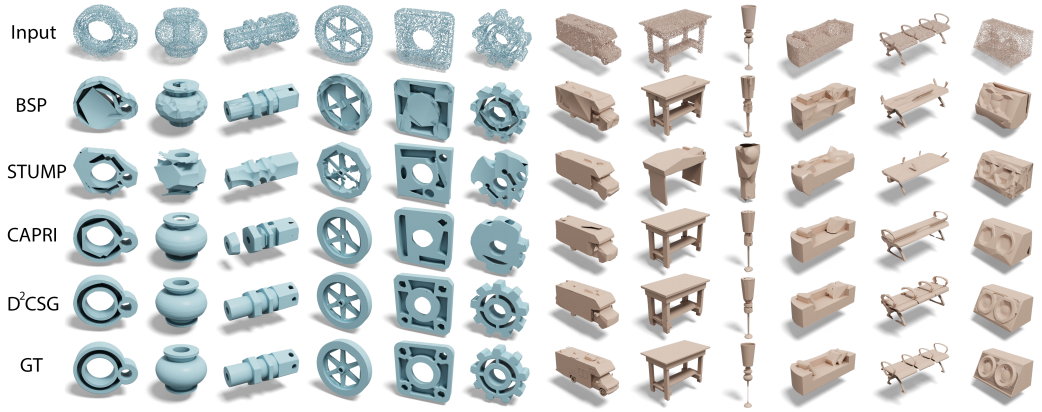
	ABC				ShapeNet			
方法	BSP	STUMP	CAPRI	Ours	BSP	STUMP	CAPRI	Ours
CD ↓	0.133	0.695	0.225	**0.085**	0.268	1.177	0.242	**0.224**
NC ↑	0.919	0.841	0.894	**0.924**	**0.889**	0.840	0.888	0.886
ECD ↓	3.899	7.303	3.308	**3.029**	1.854	3.482	1.971	**1.815**
#P ↓	360.87	67.436	68.57	**35.28**	553.76	211.36	55.21	**51.26**

CSG-Stump 不支持复杂形状之间的差集运算，因此无法重建小孔或复杂的凹陷。由于 CAPRI-Net 固定 CSG序列中被减去的中间形状仅由凸形状的并集构成，因此它无法重现具有复杂凹陷的目标形状。D^{2}与其他方法相比，D^{2}CSG 在 ABC 数据集和 ShapeNet 数据集上的所有指标中均取得了最佳的重建质量和紧凑性，但在 ShapeNet 上的 NC 指标除外，D^{2}CSG 的表现略逊一筹（$\Delta = 0.003$）；详见表 2。

消融研究。 我们进行了消融研究（见表 3），以评估以下三项关键特征的有效性， 以及它们之间的权衡：D^{2}CSG：互补基元、双分支设计和丢弃。 我们已有三个重建质量指标，因此在唯一的紧凑性度量 #P 之外，又补充了两个指标：中间形状的数量，以及由构造实体几何树进行形状分解后得到的表面片段数量；其定义的更多详情见补充材料。 我们观察到，引入互补基元（第 2 行）能更好地泛化到结构复杂的形状，且精度高于 CAPRI-Net [54]基线（第 1 行）。双分支设计进一步提高了重建精度 （第 4 行与第 2 行相比），因为残差形状中的基本体不会抹去覆盖形状的细节，而且每个分支都能更好地拟合其目标形状，而不影响另一个分支。然而，双分支设计确实牺牲了紧凑性。 在最终阶段以丢弃替代双分支（第 3 行与第 4 行相比），使基元数量减少约 50%，并改善了另外两项紧凑度指标，但精度略有下降。显然，结合全部三项组件可实现最佳权衡，如第 5 行所示：加入双分支后，配合丢弃对三项紧凑度指标的影响都相当小。 在补充材料中，我们还考察了采用基本基元以及修改最大基元数量所带来的影响。

4.2 应用

点云到构造实体几何。 我们从点云重建 CAD 形状，每个点云包含 8,192 个点。对于每个输入点，我们沿其法线采样 8 个点，扰动从高斯分布（$\mu = 0$， $\sigma = 1/64$）中采样。如果该点位于法向量的反方向，



Fi图 4：构造实体几何重建从 ABC（第 1–6 列）和 ShapeNet（第 7–12 列）的三维点云进行重建。

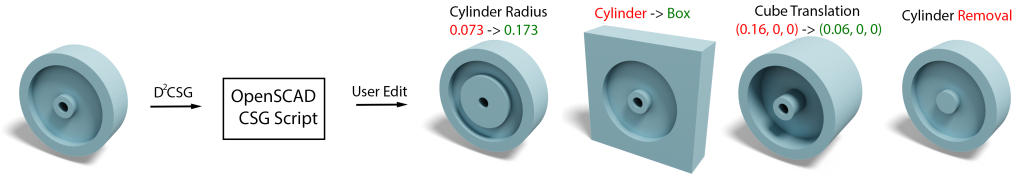


图 5：D²CSG 可针对给定形状学习 OpenSCAD 脚本，并支持编辑。

则占用值为 1，否则为 0。这样，我们为每个形状采样 65，536 个点，以拟合网络，其他设置与网格到 CSG 实验相同。表 4 中的定量比较和图 4 中的视觉比较表明，在 ABC 数据集和 ShapeNet 上，我们的网络在不同的重建相似度和紧凑度指标方面均优于 BSP-Net、CSG-Stump 和 CAPRI-Net。更多结果请参见补充材料。

形状编辑。获得基本体和 CSG 组装操作后，我们可以通过更改基元参数、调整 CSG 序列或变换中间形状来编辑形状。为便于在常用 CAD 软件中编辑形状，我们还将二次曲面基元转换为基本基元（例如立方体、球体、圆柱体等）：具体做法是，针对每个预测出的二次曲面基元，识别与其最匹配的基本基元。随后，我们可以将基元参数和 CSG 序列导出为 OpenSCAD 脚本。图 5 展示了形状编辑结果。

5 讨论、局限性与未来工作

我们提出了 D²CSG，这是一种简单而有效的方法，用于无监督学习三维 CAD 对象的通用且紧凑的构造实体几何树表示。在 ABC 和 ShapeNet 数据集上进行的大量实验表明，我们的网络在重建质量和紧凑性方面均优于最先进的方法。此外，我们还获得了大量视觉证据，表明我们的方法生成的构造实体几何树往往比以往方法生成的更加自然。

我们的网络无法在形状集合上进行泛化；它会针对单个输入形状“过拟合”，本质上是在优化以寻找一个构造实体几何组合。尽管这可能会被视为一种局限，但并非毫无道理，因为我们希望处理的 CAD 形状（即来自 ABC 的形状）似乎并不具备足够的基元组装泛化能力。 另一项局限是，二次曲面基元表示无法表示复杂曲面，只能对其进行近似，例如环面和非均匀有理 B 样条。此外，将可解释的构造实体几何操作纳入网络往往会导致梯度反向传播问题，并限制对细节的重建精度，例如椅腿上的装饰曲线。我们还希望将该方法扩展到基于图像和自由形式草图的结构化 CAD 形状重建。未来工作的另一个有趣方向，是将基元组装优化从 CAD 零件扩展到室内场景。

参考文献

[1] Dan Cascaval、Mira Shalah、Phillip Quinn、Rastislav Bodik、Maneesh Agrawala 和 Adriana Schulz。用于双向编辑的可微 3D CAD 程序。《计算机图形学论坛》，41(2):309–323，2022。[2] Angel X. Chang、Thomas Funkhouser、Leonidas Guibas、Pat Hanrahan、Qixing Huang、Zimo Li、Silvio Savarese、Manolis Savva、Shuran Song、Hao Su、Jianxiong Xiao、Li Yi 和 Fisher Yu。ShapeNet：一个信息丰富的 3D 模型库。arXiv 预印本 arXiv:1512.03012，2015。[3] Zhiqin Chen、Andrea Tagliasacchi 和 Hao Zhang。BSP-Net：通过二叉空间分割生成紧凑网格。发表于计算机视觉与模式识别会议，第 45–54 页，2020。[4] Zhiqin Chen 和 Hao Zhang。学习用于生成式形状建模的隐式场。发表于计算机视觉与模式识别会议，第 5939–5948 页，2019。[5] Thomas Davies、Derek Nowrouzezahrai 和 Alec Jacobson。作为紧凑形状表示的过拟合神经网络。arXiv 预印本 arXiv:2009.09808，2020。[6] Jianmin Zheng Daxuan Ren、Jianfei Cai、Haiyong Jiang Jiatong Li、Zhongang Cai、Junzhe Zhang、Liang Pan、Mingyuan Zhang、Haiyu Zhao 和 Shuai Yi。CSG-Stump：一种便于学习的类 CSG 表示，用于可解释的形状解析。发表于 ICCV，第 12458–12467 页，2021。[7] Boyang Deng、Kyle Genova、Soroosh Yazdani、Sofien Bouaziz、Geoffrey Hinton 和 Andrea Tagliasacchi。Cvxnet：可学习的凸分解。发表于计算机视觉与模式识别会议，第 31–44 页，2020。[8] Theo Deprelle、Thibault Groueix、Matthew Fisher、Vladimir Kim、Bryan Russell 和 Mathieu Aubry。学习用于 3D 形状生成与匹配的基本结构。神经信息处理系统进展，32，2019。[9] Tao Du、Jeevana Priya Inala、Yewen Pu、Andrew Spielberg、Adriana Schulz、Daniela Rus、Armando Solar-Lezama 和 Wojciech Matusik。InverseCSG：将 3D 模型自动转换为构造实体几何树。《ACM 图形学汇刊》(TOG)，37(6):1–16，2018。[10] Anastasia Dubrovina、Fei Xia、Panos Achlioptas、Mira Shalah、Raphaël Groscot 和 Leonidas J Guibas。通过潜在空间分解进行组合形状建模。发表于 ICCV，第 8140–8149 页，2019。[11] Kevin Ellis、Maxwell Nye、Yewen Pu、Felix Sosa、Josh Tenenbaum 和 Armando Solar-Lezama。编写、执行、评估：通过读取-求值-打印循环进行程序合成。神经信息处理系统进展，32，2019。[12] Kyle Genova、Forrester Cole、Daniel Vlasic、Aaron Sarna、William T Freeman 和 Thomas Funkhouser。利用结构化隐式函数学习形状模板。发表于计算机视觉与模式识别会议，第 7154–7164 页，2019。[13] Hao-Xiang Guo、Liu Yang、Hao Pan 和 Baining Guo。NH-Rep：用于边界表示实体隐式转换的神经半空间表示。《ACM 图形学汇刊》(TOG)，2022。[14] Pradeep Kumar Jayaraman、Aditya Sanghi、Joseph G Lambourne、Karl DD Willis、Thomas Davies、Hooman Shayani 和 Nigel Morris。UV-Net：从边界表示中学习。发表于计算机视觉与模式识别会议，第 11703–11712 页，2021。[15] Benjamin Jones、Dalton Hildreth、Duowen Chen、Ilya Baran、Vladimir G Kim 和 Adriana Schulz。Automate：用于 CAD 装配体自动配合的数据集与学习方法。《ACM 图形学汇刊》(TOG)，40(6):1–18，2021。[16] R. Kenny Jones、Theresa Barton、Xianghao Xu、Kai Wang、Ellen Jiang、Paul Guerrero、Niloy Mitra 和 Daniel Ritchie。Shapeassembly：学习生成用于 3D 形状结构合成的程序。《ACM 图形学汇刊》(TOG)，Siggraph Asia 2020，39(6):文章 234，2020。[17] Kacper Kania、Maciej Zieba 和 Tomasz Kajdanowicz。UCSG-Net：无监督发现构造实体几何树。神经信息处理系统进展，33:8776–8786，2020。[18] Yuki Kawana、Yusuke Mukuta 和 Tatsuya Harada。作为基元表示的神经星形域。神经信息处理系统进展，33:7875–7886，2020。[19] Sebastian Koch、Albert Matveev、Zhongshi Jiang、Francis Williams、Alexey Artemov、Evgeny Burnaev、Marc Alexa、Denis Zorin 和 Daniele Panozzo。ABC：面向几何深度学习的大型 CAD 模型数据集。发表于计算机视觉与模式识别会议，第 9601–9611 页，2019 年 6 月。[20] Joseph G Lambourne、Karl DD Willis、Pradeep Kumar Jayaraman、Aditya Sanghi、Peter Meltzer 和 Hooman Shayani。BRepNet：实体模型的拓扑消息传递系统。发表于计算机视觉与模式识别会议，第 12773–12782 页，2021。[21] Eric-Tuan Lê、Minhyuk Sung、Duygu Ceylan、Radomir Mech、Tamy Boubekeur 和 Niloy J Mitra。Cpfn：用于高分辨率点云的级联基元拟合网络。发表于计算机视觉与模式识别会议，第 7457–7466 页，2021。[22] Changjian Li、Hao Pan、Adrien Bousseau 和 Niloy J. Mitra。Free2cad：将徒手绘图解析为 CAD 命令。《ACM 图形学汇刊》(TOG)，41(4)，2022。[23] Lingxiao Li、Minhyuk Sung、Anastasia Dubrovina、Li Yi 和 Leonidas J Guibas。对 3D 点云进行几何基元的监督拟合。发表于计算机视觉与模式识别会议，第 2652–2660 页，2019。[24] Yichen Li、Kaichun Mo、Lin Shao、Minhyuk Sung 和 Leonidas Guibas。从单张图像学习 3D 零件装配。发表于欧洲计算机视觉会议，第 664–682 页。施普林格，2020。[25] Yangyan Li、Xiaokun Wu、Yiorgos Chrysathou、Andrei Sharf、Daniel Cohen-Or 和 Niloy J. Mitra。Globfit：通过发现全局关系一致地拟合基元。《ACM 图形学汇刊》(TOG)，30(4)，2011。[26] Zhijian Liu、William T Freeman、Joshua B Tenenbaum 和 Jiajun Wu。物理基元分解。发表于欧洲计算机视觉会议，第 3–19 页，2018。[27] Lars Mescheder、Michael Oechsle、Michael Niemeyer、Sebastian Nowozin 和 Andreas Geiger。占据网络：在函数空间中学习 3D 重建。发表于计算机视觉与模式识别会议，第 4460–4470 页，2019。

[28] Ben Mildenhall、Pratul P Srinivasan、Matthew Tancik、Jonathan T Barron、Ravi Ramamoorthi 和 Ren Ng。NeRF：将场景表示为用于视图合成的神经辐射场。《ACM 通讯》，65(1)：99–106，2021。[29] Kaichun Mo、Shilin Zhu、Angel X. Chang、Li Yi、Subarna Tripathi、Leonidas J. Guibas 和 Hao Su。PartNet：用于细粒度分层部件级三维物体理解的大规模基准。在计算机视觉与模式识别会议上，页码 90–918，2019。[30] Chengjie Niu、Jun Li 和 Kai Xu。Im2struct：从单张 RGB 图像恢复三维形状结构。在计算机视觉与模式识别会议上，2018。[31] Wamiq Para、Shariq Bhat、Paul Guerrero、Tom Kelly、Niloy Mitra、Leonidas J Guibas 和 Peter Wonka。Sketchgen：生成受约束的 CAD 草图。神经信息处理系统进展，34：5077–5088，2021。[32] Jeong Joon Park、Peter Florence、Julian Straub、Richard Newcombe 和 Steven Lovegrove。DeepSDF：学习用于形状表示的连续有符号距离函数。在计算机视觉与模式识别会议上，页码 165–174，2019。[33] Despoina Paschalidou、Angelos Katharopoulos、Andreas Geiger 和 Sanja Fidler。神经部件：使用可逆神经网络学习表现力丰富的三维形状抽象。在计算机视觉与模式识别会议上，页码 3204–3215，2021。[34] Despoina Paschalidou、Ali Osman Ulusoy 和 Andreas Geiger。重新审视超二次曲面：学习超越长方体的三维形状解析。在计算机视觉与模式识别会议上，页码 10344–10353，2019。[35] Daxuan Ren、Jianmin Zheng、Jianfei Cai、Jiatong Li 和 Junzhe Zhang。ExtrudeNet：用于形状解析的无监督逆向草图与拉伸。在《计算机视觉——ECCV 2022：第 17 届欧洲计算机视觉会议，特拉维夫，以色列，2022年10月23日至27日，论文集，第二部分》中，页码 482–498。施普林格，2022。[36] Gopal Sharma、Rishabh Goyal、Difan Liu、Evangelos Kalogerakis 和 Subhransu Maji。CSGNet：用于构造实体几何的神经形状解析器。在计算机视觉与模式识别会议上，页码 5515–5523，2018。[37] Gopal Sharma、Difan Liu、Subhransu Maji、Evangelos Kalogerakis、Siddhartha Chaudhuri 和 Radomír M˘ech。ParseNet：用于三维点云的参数化曲面拟合网络。在欧洲计算机视觉会议上，页码 261–276，2020。[38] Towaki Takikawa、Joey Litalien、Kangxue Yin、Karsten Kreis、Charles Loop、Derek Nowrouzezahrai、Alec Jacobson、Morgan McGuire 和 Sanja Fidler。神经几何细节层次：使用隐式三维形状进行实时渲染。在计算机视觉与模式识别会议上，页码 11358–11367，2021。[39] Yonglong Tian、Andrew Luo、Xingyuan Sun、Kevin Ellis、William T Freeman、Joshua B Tenenbaum 和 Jiajun Wu。学习推断并执行三维形状程序。arXiv 预印本 arXiv:1901.02875，2019。[40] Shubham Tulsiani、Hao Su、Leonidas J. Guibas、Alexei A. Efros 和 Jitendra Malik。通过组合体积基元学习形状抽象。在计算机视觉与模式识别会议上，页码 2635–2643，2017。[41] Mikaela Angelina Uy、Yen yu Chang、Minhyuk Sung、Purvi Goel、Joseph Lambourne、Tolga Birdal 和 Leonidas Guibas。Point2Cyl：从点云逆向工程三维物体，生成拉伸圆柱体。在计算机视觉与模式识别会议上，页码 11850–11860，2022。[42] Francis Williams、Teseo Schneider、Claudio Silva、Denis Zorin、Joan Bruna 和 Daniele Panozzo。用于曲面重建的深度几何先验。在计算机视觉与模式识别会议上，页码 10130–10139，2019。[43] Karl D.D. Willis、Pradeep Kumar Jayaraman、Hang Chu、Yunsheng Tian、Yifei Li、Daniele Grandi、Aditya Sanghi、Linh Tran、Joseph G. Lambourne、Armando Solar-Lezama 和 Wojciech Matusik。JoinABLe：学习参数化 CAD 接头的自底向上装配。在计算机视觉与模式识别会议上，页码 15849–15860，2022。[44] Karl DD Willis、Yewen Pu、Jieliang Luo、Hang Chu、Tao Du、Joseph G Lambourne、Armando Solar-Lezama 和 Wojciech Matusik。Fusion 360 gallery：一个用于根据人工设计序列进行程序化 CAD 构建的数据集和环境。《ACM 图形学汇刊》（TOG），40(4)：1–24，2021。[45] Karl D. D. Willis、Yewen Pu、Jieliang Luo、Hang Chu、Tao Du、Joseph G. Lambourne、Armando Solar-Lezama 和 Wojciech Matusik。Fusion 360 gallery：一个用于根据人工设计序列进行程序化 CAD 构建的数据集和环境。《ACM 图形学汇刊》（TOG），40(4)，2021。[46] Rundi Wu、Chang Xiao 和 Changxi Zheng。DeepCAD：用于计算机辅助设计模型的深度生成网络。在 ICCV 上，页码 6772–6782，2021。[47] Rundi Wu、Chang Xiao 和 Changxi Zheng。DeepCAD：用于计算机辅助设计模型的深度生成网络。在 IEEE/CVF 国际计算机视觉会议论文集上，页码 6772–6782，2021。[48] Zhirong Wu、Shuran Song、Aditya Khosla、Fisher Yu、Linguang Zhang、Xiaoou Tang 和 Jianxiong Xiao。3D ShapeNets：一种用于体积形状的深度表示。在计算机视觉与模式识别会议上，页码 1912–1920，2015。[49] Xianghao Xu、Wenzhe Peng、Chin-Yi Cheng、Karl DD Willis 和 Daniel Ritchie。使用区域图推断 CAD 建模序列。在计算机视觉与模式识别会议上，页码 6062–6070，2021。[50] Xiang Xu、Karl DD Willis、Joseph G Lambourne、Chin-Yi Cheng、Pradeep Kumar Jayaraman 和 Yasutaka Furukawa。Skexgen：使用解耦码本自回归生成 CAD 构建序列。arXiv 预印本 arXiv:2207.04632，2022。[51] Kaizhi Yang 和 Xuejin Chen。通过点云联合分割进行长方体形状抽象的无监督学习。《ACM 图形学汇刊》（TOG），40(4)：1–11，2021。[52] Chun-Han Yao、Wei-Chih Hung、Varun Jampani 和 Ming-Hsuan Yang。从图像集合中发现三维部件。在 ICCV 上，页码 12981–12990，2021。[53] Lior Yariv、Jiatao Gu、Yoni Kasten 和 Yaron Lipman。神经隐式曲面的体渲染。神经信息处理系统进展，34：4805–4815，2021。[54] Fenggen Yu、Zhiqin Chen、Manyi Li、Aditya Sanghi、Hooman Shayani、Ali Mahdavi-Amiri 和 Hao Zhang。CAPRI-Net：通过自适应基元装配学习紧凑型 CAD 形状。在计算机视觉与模式识别会议上，页码 11768–11778，2022。

[55] Chuhang Zou、Ersin Yumer、Jimei Yang、Duygu Ceylan 和 Derek Hoiem。3d-prnn：使用循环神经网络生成形状基元。发表于*ICCV*，第 900–909 页，2017。