

MeshCNN: 一种基于边的神经网络

RANA HANOCCA, 以色列特拉维夫大学

AMIR HERTZ, 以色列特拉维夫大学

NOA FISH, 以色列特拉维夫大学

RAJA GIRYES, 以色列特拉维夫大学

SHACHAR FLEISHMAN, 亚马逊

DANIEL COHEN-OR, 以色列特拉维夫大学

多边形网格为三维图形提供了一种有效的表示方法。它们可以准确地捕捉图形的表面和拓扑信息【A1】, 并利用不均匀性来表示大平面区域以及图形尖锐、复杂的特征【A2】。然而, 多边形网格这种不均匀性和不规则性也同时妨碍了基于卷积和池化操作相结合的神经网络对网格的分析工作【A3】。在本文中, 我们利用了网格的特有属性设计了一种专门处理三角网格的卷积神经网络 MeshCNN 来直接对三维图形分析处理。与经典的 CNN 类似, MeshCNN 通过利用图形固有的大地线的连接, 将专用的卷积与在网格边上操作的池化层相结合【A3】。卷积被应用在边以及这条边关联三角形的四条边上【A4】, 并通过保留曲面拓扑信息的边缘折叠操作来进行池化【A5】, 进而为后续的卷积生成新的网格连接。MeshCNN 学习要折叠的边, 从而形成了一个任务驱动的过程, 在这个过程中网络会显示和扩展一些重要的特征, 并在同时丢掉那些冗余的特征。我们也通过进行不同的学习任务来证明我们的任务驱动池对三维网格分析的有效性。

ACM 参考格式:

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. Mesh CNN:A Net work with an Edge. *ACM Trans. Graph.*1,1 (February 2019), 12 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnn>

1 引言

三维图形是计算机图形学领域的前沿和核心, 也是计算机视觉、计算几何等相关领域的主要有研究价值的议题。我们身边的图形, 特别是用于描述自然实体的图形, 通常都是由连续的表面构成的。

由于算力限制的原因, 并且为了便于数据处理, 不同的三维图形离散近似表示方法已被提出并且广泛地在一系列基于图形的应用中使用。其中, 最受欢迎的是多边形的网格表示, 简称网格, 通过一组三维空间的二维多边形来近似曲面[Botsch et al.2010]【B1】。网格提供

作者地址: Rana Hanocka, 以色列特拉维夫大学; Amir Hertz, 以色列特拉维夫大学; Noa Fish, 以色列特拉维夫大学; Raja Giryes, 以色列特拉维夫大学; Shachar Fleishman, 亚马逊; Daniel Cohen-Or, 以色列特拉维夫大学。

本作品同意全部或部分的数码拷贝、硬拷贝供个人或课堂免费使用, 前提是复制品的制作或发布不是为了牟利或商业利益, 并且复制品须带有本通知和首页的完整引文。必须尊重除 ACM 以外的其他人拥有本作品的版权。允许信用提取。以其他方式复制或重新发表、在服务器上发布或者重新发布到期刊列表需要实现获得指定许可权并且/或收取费用。许可权从这里进行申请 permissions@acm.org

© 2019 Association for Computing Machinery.

0730-0301/2019/2-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnn>

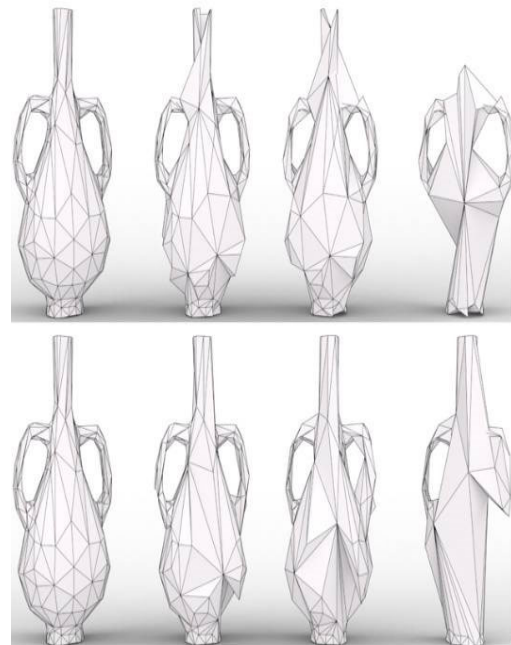


图 1. 在不规则结构上进行网格池化, 并且在空间上适应具体任务。与几何简化不同 (以最小的几何失真删除边), 网格池化寻找哪些边需要被折叠后送入网络。第一排: MashCNN 训练于判断花瓶是否有把手, 第二排: 训练判别花瓶是否有颈部 (颈口)。

有效的, 不均匀的形状表示。一方面, 只需要少量的多边形就可以捕捉到大而简单的曲面【B2】。另一方面, 图形表示的灵活性在需要时可以提供更高的分辨率【B3】, 也支持对几何上错综复杂的主要图形特征进行真实的重建与揣摩【B4】。网格的另一个有益的特性在于它自己本身就提供了图形连接的信息【B5】。这形成了下层表面的综合表示。

与另一种流行的点云表示法相比, 网格的优势也是显而易见的。尽管点云表示方法简单而且与常用的数据采集技术有直接的联系 (扫描)【B6】, 但是在需要更高质量的图形和保留锐利特征的需求下, 它仍然存在不足【B7】。

最近几年, 在图像上使用卷积神经网络 (CNNs) 在分类和语义分割等多种任务中都表现出了优异的性能[2013;2014;2018]【B8】。它们成

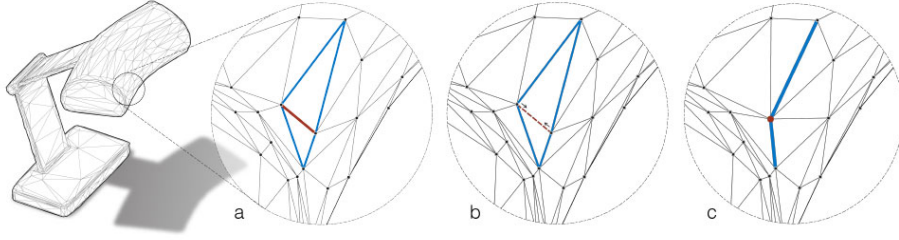


图 2. (a) 特征的计算是在边上进行的，通过应用卷积与两个三角形的四条边组成的邻域计算获得。四个蓝色边与红色边相关联。池化过程如 (b) 和 (c) 所示。在 (b) 中，红色边折叠为一个点，四条边（蓝色）合并为 (c) 中的两条边（蓝色）。注意：在单条边折叠的过程中，五条边将转化成两条。

功的秘诀在于将卷积、非线性和池化层的结合【B9】，从而形成一个对输入的无关紧要的变化保持恒定（稳健）的框架[LeCun 2012; Krizhevsky et al. 2012]【B10】。然而，由于图像是在离散值的规则网格上表示的【B11】，因此将 CNNs 直接扩展到处理不规则结构是困难的事情【B12】。

最初的解决方法避开了使用规则让 CNN 适用于不规则数据的方式【B13】，而是将 3D 图形映射到多个 2D 投影[Su et al. 2015]或 3D 体素网格[Wu et al. 2015]【B14】。虽然这些方法受益于直接使用了易于理解的图像 CNN 操作，但是它们间接表示的转化需要大量的内存【B15】和浪费、冗余的 CNN 计算（例如，在未使用的体素上进行卷积运算）【B16】。

更加有效的方法是直接将 CNN 应用于不规则的稀疏的点云表示[Qi et al. 2017a]【B17】。虽然这些方法得益于紧密的输入表示，但是点云先天无法表达局部曲面【B18】。此外，点云的邻域和连通性的概念定义不清晰，这使得卷积和池化的操作变得困难【B19】。这种概念的模糊性也促进了一批旨在克服这一问题的工作开展[Monti et al. 2017; Wang et al. 2018a; Yi et al. 2017]。

为了挖掘网格表示本身的先天潜力，我们提出了 MeshCNN：一种专门为网格设计，与众所周知的 CNN 模型类似的神经网络。Mesh CNN 直接在不规则的三角形网格上进行卷积和池化操作，这些操作与网格独有的特性相协调。在 MeshCNN 中，网格的边类似于图像中的像素，因为边是应用所有操作的基本单位【B20】。我们之所以选择使用边，是应为每条边正好与两个面（三角形）相关联，这定义了一个由四条边组成的自然固定大小的卷积邻域（如图 2）【B21】。我们利用连续面的法线顺序来应用对称卷积运算，学习与旋转、缩放和平移变换无关的边特征。

MeshCNN 的一个关键特性是它独特的池化操作：mesh-pooling，它在不规则结构上进行操作，并在空间上适应任务。在 CNN 中，池化减少了网络中的特征数量，从而学习消除影响不大的冗余特征。

由于特征位于边上，因此下采样效果的直观感受就像是使用了大名鼎鼎的网格简化技术：边折叠[Hoppe 1997]【B22】。然而，与传统的边缘折叠技术不同，网格池化操作以特定于任务的方式选择将哪些边折叠以后送到网络【B23】，而且被丢弃的边只会引起最小的几何失真【B24】。被清除的边是对目标特征贡献最小的边（参见图 1 和图 8 中的示例）【B25】。

为了增加灵活性并支持各种可用数据，每个池化层将网格简化为预定的固定边数【B25】。此外，当它产生任何指定的输出边数时，MeshCNN 对网格的大小是不可知的，并且能够处理不同的三角剖分。如图 1 所示，我们展示了中间的池化步骤以及最终的输出结果在语义上是可解释的。为了说明我们方法的效果，我们在图形分类和分割任务上进行了大量的实验，并在常规数据集和高度非均匀网格上均得到了优于最新方法的实验结果。

2 相关工作

我们在工作中提出或使用的许多运算操作都是基于经典的网格处理技术[Hoppe 1999; Rusinkiewicz and Levoy 2000; Botsch et al. 2010; Kalogerakis et al. 2010]，或者更具体的说是基于网格简化技术[Hoppe et al. 1993; Garland and Heckbert 1997; Hoppe 1997]。特别的，我们使用边折叠技术[Hoppe 1997]来实现任务驱动的池化操作。传统的网格简化技术旨在以最小的几何失真来减少网格元素的数量[Tarini et al. 2010; Gao et al. 2017]，但在这项工作中，我们使用网格简化技术来降低神经网络环境下特征图的分辨率【C1】。在下文中，我们将回顾利用神经网络进行三维数据分析的相关工作，将这些神经网络按输入的代表类型进行组织。

多视角二维投影。利用现有的技术和架构从二维表示三维是可行的，这需要通过三维图形从不同角度的二维投影来表示三维图形【C2】。这些渲染图像集作为标准 CNN 模型后续处理的输入【C3】。Su 等人[2015]是第一次使用多视角 CNN 进行图形分类，而这种方法（原始

无法进行图形语义的分割【C4】。后来, [Kalogerakis et al.2017]提出了一个更全面的形状分割多视图框架:生成每个三维视图的图像级分割图【C5】,然后使用 CRF (经过训练的端到端)解决标签一致性的问题。Qi 等人[2016]探索了基于视图和体积的方法,并观察到第一种方法与当时可用方法相比的优越性。

体素法。把三维图形转换成二进制的体素形式【C7】,为三维图形的表示提供了一种栅格式的方法,这种方式与二维图像的栅格化表示很类似。因此,在二维栅格上的操作可以直接扩展到三维的栅格上,图像领域的一般方法也可以很容易在形体领域运用。Wu 等人[2015]率先使用该方法,并且展示了一种可以给体素形体做分类和补全的 CNN 模型。在这篇文章之后, Brock 等人 [2016]通过一种基于体素的变量自动编码技术,实现了图形的重建【C8】。[Tchapmi et al. 2017]通过三线插值和条件随机场(CRF)的方法,结合体积网络提升了图形语义分割的效果【C9】。Hanocka 等人[2018]使用体素法训练网络来退化栅格上的扭曲部分,来做图形的对齐,并在原本的网格上使用这种估计后的变形【C10】。

尽管这种表示法相对简单,但是他对算力有很大的要求,需要很大的存储空间。为了减轻内存的消耗,一些为加速策略被提出,他们削弱了体积法表示的稀疏性来为表示减少负担。

图。图结构是由栅格表示法演化而来的一种表示方式,它可以表现不规则性【C11】。为了实现对基于图结构数据(主要有社交网络,通信张量网络、基因数据)的分析,大量的工作中直接使用了神经网络来处理。一种方法提倡使用图表示法的拉普拉斯变换来处理[Bruna et al.2014; Henaff et al. 2015; Defferrard et al. 2016; Kostrikov et al. 2018],因此往往用在谱域上。另一种方法提倡通过直接提取局部相连的部分并且把他们转换成规范的形式来处理图结构,并使用这种规范的形式来被神经网络所处理[Niepert et al. 2016]。Atwood et al. [2016]提出扩散卷积的方法,其中扩散是用来判定每一个结点的相邻结点的。Monti et al. [2017]使用图像的空间域把表面参数化成局部块。Xu et al. [2017]在表面块上使用定向卷积来做语义分割任务。Yi et al. [2017]使用在谱域上使用图卷积来做三维分割任务。Kostrikov et al. [2018]使用拉普拉斯表面网络给三维形体做生成模型。Such et al. [2017]在图结构上引入了顶点过滤的概念,但没有结合池化操作来做特征的聚集。这些方法大多是在图的顶点上操作的。

流形。这是 Masci et al. [2015]所第一个提出的,介绍了在网格上局部特征的深度学习方法(固有的网格描述符和[Kokkinos et al. 2012]类似)【C12】。并且已经证明了如何使用这些学得特征来做对应和

提取。特别地,他们实现了在固有网格上的卷积操作。

通常,流形图形上的局部块是近似欧几里得的。通过把三维流形参数化成二维的,再利用这种特性就可以通过 CNN 模型做流形的分析[Henaff et al. 2015; Boscaini et al. 2016; Sinha et al. 2016; Maron et al. 2017; Ezuz et al. 2017]。Boscaini et al. [2015]通过对顶点频率的分析来学习局部固有的三维形体。另一种方式使用复曲面的拓扑结构来定义形体图像上的卷积。h [Haim et al. 2018; Maron et al. 2017]. Poulenard et al. [2018]定义了一种新的卷积层使得大地线信息可以在网络的各层之间传播。

Verma et al. [2018]提出了一种图神经网络,在这种网络中每个顶点的相邻顶点在卷积操作中是没有被提前定义的,而是会根据他的特征动态计算。Tatarchenko et al. [2018]引入了切线卷积,在这种方法中,每个点周围一个小的区域被用来重建局部函数,卷积就是在局部函数上使用的。不像之前的研究,他们通过在规则三维栅格上局部采样进行池化操作。一些衍生的模型也被提出了。Litany et al. [2018]引入了一种自动编码器来进行形体的补全。Ranjan et al. [2018] 实现了用网格自编码器生成的三维面部。

[Bronstein et al. 2017]对几何深度学习做了全面的调研。我们的方法和过去这些方法的独特点在于我们的网络是专门为了在网格结构上运行而设计的。特别地,我们单独为选择哪个区域简化这一任务实现了一个池化操作算子,它是基于任务的目的来学习的。

据我们所知,这篇文章是目前第一个用来 (i) 在网格的边上做卷积 (ii) 可以适应自己的任务来学习池化操作的研究。在 [Ranjan et al. 2018]中,提出了一种固定的池化操作,用来给网格自动编码。在图神经网络中,经过训练学习的池化操作被提出过 [Ying et al. 2018; Cangea et al. 2018]。然而这些操作没有把三角形网格的特性考虑进去。

一种提取了边的特征的卷积和双图卷积模型在[Monti et al. 2018]中被提出,它扩展了图注意力网络[Velickovic et al. 2018]。但是,在他们的研究中使用的注意力和优化方法和我们的不同;在这份研究中,我们为网格定义了一些操作算子,这些算子利用了网格的独特结构和特性。这使得我们可以定义一个具有恒定网络特征的对称卷积。

点云,几乎是所有方法中最简单的表达方式,点云为三维图形提供了一个不加修饰的近似表示方法。这种表示方法和数据提取关系紧密,并且很容易从别的表示方法转换,这些优势使得点云经常用于数据分析。因此,最近的科研工作会使用神经网络来开发对点云分析的技术。PointNet [Qi et al. 2017a]提出使用 1x1 的卷积和全局最大池化来实现阶的恒定。在它的后续研究中,PointNet++ [Qi et al. 2017b],所有的点被分成几份来更好地捕获局部结构。Wang et al. [2018b] 考

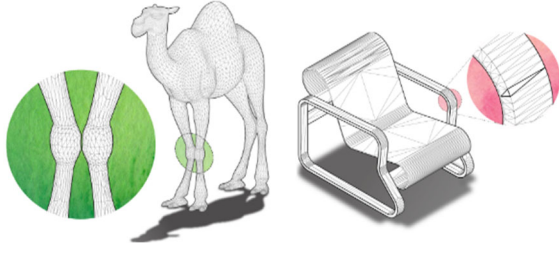


图 3. 多边形网格的优势。左图：对图形结构有精确的描述。网格，不像点云，可以通过测地线分离轻松传达骆驼关节处的独特特征（放大的图可以看到），尽管在欧氏空间上他们很相似。右图：自适应的不均匀表示。大而平的表面只需要很少的大多边形就可以表示了，而细节较多的区域则需要有很多小的多边形表示。

虑了局部点周围的信息，并且会根据点和点之间在特征空间的距离来计算相似性，并根据相似性动态更新这些信息。虽然大部分的基于点的方法关注全局或者中级属性，[Guerrero et al. 2018] 提出了一种可以根据初始点云计算局部形体特征（比如法线、曲率）的网络，而 [Williams et al. 2018] 通过学习几何先验来根据点云实现表面的重建。Atzmon et al [2018] 通过把点云方程映射到体积方程的方式，在点云上定义了一个高效的卷积操作算子。这使得该方法对于点的阶数是不变的，并且在数据存在变形时很鲁棒。最近，Li et al. [2018] 提出了 PointCNN，它将卷积的概念从局部的栅格扩展到了欧氏邻域的点上的 x -卷积。

在这份研究工作中，与之前的工作不同的是，我们依靠网格的边来提供不均匀的、测地线的领域信息，并且拥有数量一致的卷积邻域。不变的特征计算在边上进行，并通过网格简化技术，例如边折叠，以符合形体的几何结构和拓扑结构。

3 概述：在网络上应用 CNN

在计算机图形学领域，三维数据最常见也是最基本的表示方式就是不均匀多边形网格：大而平的表面只需要很少的大多边形就可以表示了，而细节较多的区域则需要有很多小的多边形表示。网格可以很简明地表达图形表面的拓扑结构：很详实地描述了复杂的结构，并且消除了邻近表面的歧义（见图 3）。

为了实现把 CNN 直接应用到三角形网格的目标，我们必须先定义和实现几个和标准 CNN 模型类似的模块：卷积和池化层。图像是由规则栅格中的离散数值来表示的，和图像不同的是，网格分析的主要难点在于网格的不规则性和不均匀性。在这份研究工作中，我们希望能利用这些难点和独特的性质，而不是绕过他们。因此，我们有意设计该网络，使得它可以直接在网格结构上进行卷积和池化操作的，并且避免了它向规则、均匀的表达方式的转化。

不变卷积。在我们的设定里，假设所有的图形都被表示为流形网格【D1】，也可能有边界边。这个假设保证了每一条边最多被两个面（三角形面）共享，因此它要么和两条边邻接、要么和四条边邻接。一个面的顶点按逆时针排序，这样就可以定义出每条邻边的两种表达顺序【D2】。例如，见图 4，边 e 的 1-环邻边可以被排序为 (a,b,c,d) 或者 (c,d,a,b) ，顺序取决于哪个面被定义为第一个邻域。这会模糊卷积的感受域，阻碍了不变特征的形成。我们采取了两种措施来解决这个问题，并且在我们的网络中我们保证了相似变换（旋转，平移，缩放）的不变性。首先，我们仔细设计了边的描述符，使其只包含相对几何特征，这些特征对于相似变换是不变的。其次，我们把四条 1-环边聚合成两对有歧义的边（比如， a 和 c ， b 和 d ），再通过在每对边上使用简单的对称函数（比如， $\text{sum}(a,c)$ ）生成新的特征。卷积是在新的对称特征上进行的，这样就消除了顺序的不确定性。

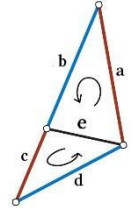
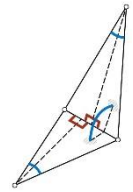


图 4

输入特征。对于每一条边，输入的特征是一个 5 维的向量：二面角、两个内角和两个对于每个面的边长比。边长比指的是边的长度和两个邻接面的垂线（如图）的比值。【D3】我们给每一个基于两个面的特征（内角和边长比）排序，这样就解决了顺序的不确定性并保证了不变性。可以注意到这些特征都是相对的，这使他们对于平移、旋转、和均匀缩放是不变的。



全局排序。边的全局排序指的是一个特定形状的边的数据（输入特征）输入到网络中的顺序。由于卷积是在局部邻域内的操作，该排序对于卷积操作没有影响。通过扩展，全卷积任务，例如分割，不会被它影响。对于需要全局要素聚合的任务，例如分类，我们使用了，Qi 等人[2017a] 在 PointNet 中提出的方法，并且在网络的卷积层和全连接层之间加了一层全局平均池化层。这一层使得初始的排序操作不是特别重要，并且保证了对于变换的不变性。

池化。网格池化是通过边折叠步骤完成的【D4】，边折叠在图 2(b)和(c)中解释了。在 (b) 中，虚线边被折叠成了一个点，带来的后续结果是，四条邻边(蓝色的)合并成了(c)中两条(蓝色的)边。注意，在边折叠步骤中，五条边变成了两条边。算子是按照边的特征（最小法线）来区分优先级的，因此使得网络可以决定选择网格的哪一部分来简化，哪些部份来选择不变。这就形成了一个有任务导向的过程，网络可以根据任务学习决定每个部分的优先级（见图 1）。

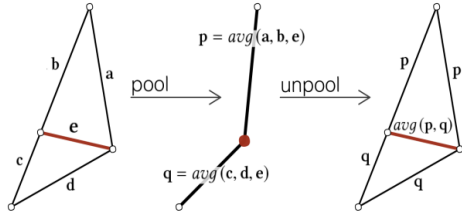


图 5. 在网格池化和逆池化过程中的特征聚合。

我们的简化性质的一个显著优点是，它在池化层到达最终的全连接层之前提供了关于输出维度的灵活性，同时池化还有助于增强初始网格三角剖分的鲁棒性。虽然它没有向三角测量提供平衡，但在实践中，通过持续折叠并简化网格，在初始曲面细分存在差异的情况下，我们仍然可以观察到相似表示的收敛。

4 方法

基于网格的表示可以方便地在单个矩阵中提供空间邻域和特征。但由于不规则网格不符合这种格式，我们必须将特征与连通性分开定义。我们通过在网格的标准构造中工作来实现这一点。

网格由对 (V, F) 定义，其中 $V = \{v_1, v_2, \dots\}$ 是 \mathbb{R}^3 中的一组顶点位置， F 定义了连接（三角形网格的三个顶点）。给定 (V, F) ，也使用边缘 E 一组顶点对地定义网络连接。

所有网格元素 V ， F 和 E 都可以与各种特征（例如法线或颜色）相关联。在这项工作中， E 也拥有一组特征。边缘特征一开始是一组相似不变的几何特征（相当于图像中的 RGB 值），并在通过网络层的过程中开发出更高的抽象特征。

在实验中网格为网络提供了卷积邻域的连接和初始几何输入特征两个属性。一旦提取了输入特征，由于边缘折叠操作后的新顶点位置对分类和 SEG 调解任务没有影响，并且仅用于可视化目的，此时网格顶点就没有任何意义了。

接下来，我们将展开并提供有关网格卷积、网格池和网格解池操作的详细信息。

4.1 网格卷积

我们为边缘定义了一个卷积运算符，其中空间支持端口使用四个入射邻域定义（图 3）。卷积是核 K 和邻域之间的点积，因此，边缘 E 及其四个相邻边的卷积为：

$$e \cdot k_0 + \sum_{j=1}^4 k_j \cdot e^j,$$

其中 e^j 是 e 的第 j 个卷积邻域的特征。如图 4 所示， e 的四个邻域是 (e^1, e^2, e^3, e^4) 即可能是 (a, b, c, d) 也可能是 (c, d, a, b) ，这使得每个滤波器值最多操作两个可能的边缘（例如 k_1 可能在 a 上也可能在 c 上）。为了保证对输入数据的排序的卷积不变性，我们将一组简单的对称函数应用于模糊对。这就产生了一组新的保证不变的卷积邻域。在我们的环境中，边缘 e 的接收领域：

$$(e^1, e^2, e^3, e^4) = (|a - c|, a + c, |b - d|, b + d). \quad (2)$$

这产生使得卷积操作将忽略网格初始元素排序，使得无论如何都会产生相同的输出。一般矩阵 MUL Tiplation(Gemm)通过将图像扩展到列矩阵（如 IM2Col [Jia 2014]）实现具有内核的多通道张量的卷积：。等效地，我们构建未包装的矩阵以有效地执行卷积操作。

在实践中，我们可以使用高度优化的批处理运算(如 conv2D)通过将所有边缘特征聚合到 $n_c \times n_e \times 5$ 的特征张量，其中 n_c 是边的数量， n_e 是特征通道的数量，5 是边和卷积邻域的数量（等式 2）这个矩阵乘以一个使用标准 GEMM 的卷积的权值矩阵。

在卷积操作之后，生成了一个新的批特征张量，其中新的特征的数量等于卷积核的数量（就像在图像中一样）。注意，在每个池阶段之后，新的连接将为下一个卷积定义新的卷积邻域。

4.2 网格池化

我们通过定义三个核心操作将传统的池化操作扩展到不规则数据，同时推广了池化的概念：

- 1) 定义给定邻接的池区域
- 2) 每个池区域中的合并特征
- 3) 重新定义合并特征的邻接

对于池化如图像等常规数据，邻接是固有地隐含的，因此，池化区域直接由所选择的内核大小决定。由于每个区域的特征被合并(如平均池化和最大池化)，从而产生另一个均匀间隔的网格，因此新的邻接再次被固有地定义。在讨论了上面定义的三个通用的池操作之后，传统的池显然是广义过程的一个特殊情况。

网格池化是广义池化的另一种特殊情况，其邻接性是由拓扑结构决定的。与具有自然减少系数的图像不同，例如 4 为 2×2 的池，我们将网格池定义为一系列边缘折叠操作，其中每个这样的边缘折叠都将五条边转换为两条边。因此，我们可以通过添加一个定义池网格中目标边数的超参数来控制每个池运算符后网格的期望分辨率。在运行期间，提取网格邻接信息需要查询不断更新的特殊数据结构（详情参见 [Berg 等人]）

我们根据边缘特征的大小来优先考虑边缘折叠的顺序（使用优先级队列），允许网络选择网格的哪些部分与解决该任务相关。这使得网络能够非均匀地折叠对损失最不重要的某些区域。将两相邻的边重合会删除三条边（如图 2 所示）因为两面都变成了一条边。通过取出每个特征通道上的平均值，每个面中的三个边的每个特征都被合并为一个新的边缘特征。

边折叠操作根据边缘特征的强度进行优先排序，作为它们的 ℓ_2 范数。这些特征被聚合，如图 5 所示，其中有两个合并操作，一个是针对最小边缘特征 e 的每个入射三角形，从而得到两个新的特征向量（表示 p 和 q ）。这两个三角形的信道索引 i 中的边缘特征：

$$p_i = \text{avg}(a_i, b_i, e_i), \text{ and } q_i = \text{avg}(c_i, d_i, e_i) \quad (3)$$

边缘折叠后，为后续边缘折叠更新半边数据结构。

最后，注意：并非每条边都可以折叠。在我们的设置中不允许产生非流形面，因为它违反了四个卷积邻域假设。因此，如果边在 1 环的交点上有三个顶点，或者有两个边界顶点，它被认为是无效的。

4.3 网络逆池化

逆池化是池化操作的（部分）逆。虽然池化层降低了特征激活（编码或压缩信息）的分辨率，但逆池化层增加了特征激活（解码或解压缩信息）的分辨率。池操作记录合并操作的历史记录（例如，最大位置），并使用它们展开功能激活。因此，逆池化操作没有可学习的参数，它通常与卷积相结合，以恢复在池化操作中丢失的原始分辨率，与卷积的结合有效地使逆池化成为一个可学习的操作。

每个网格逆池化层都与网格池化层成对，以对网格拓扑和边缘特征进行上采样。非池化层通过在池化之前存储连接来恢复上采样拓扑结构（在网格池化之前）。注意，连接上采样是可逆的操作（与图像中一样）。对于非池化的边缘特征计算，我们保留一个图，它存储从原始边（池之前）到新边（池之后）的邻接结构。每个未汇集的边缘特征都是汇集的边缘特征的加权组合。逆池化的过程如图 5 所示。

5 实验

MeshCNN 是将 CNN 直接应用于三角网格的通用方法，有很多应用。使用第 4 节中描述的 MeshCNN 构建块，我们可以构建不同的网络配

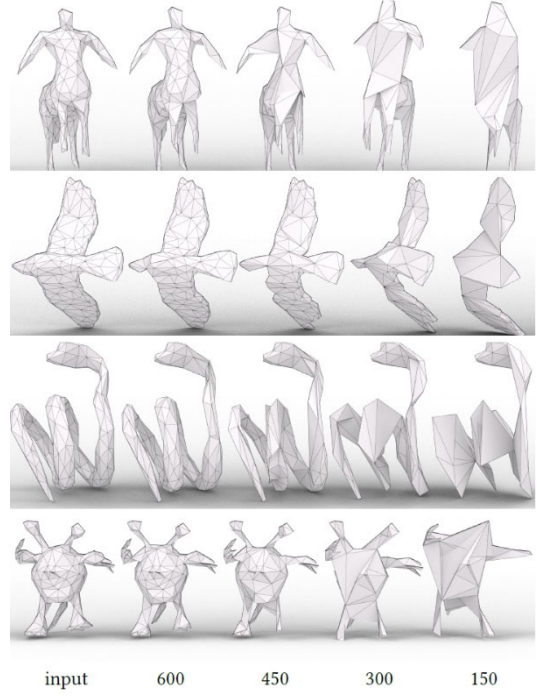


图 6. SHREC11 分类数据集上的中间集合网格。输入网格都被简化为大约 750 条边（500 个面），并按顺序汇集到 600、450、300 和 150 条边。

置以解决不同的任务。与传统的 CNN 一样，这些构建块提供了一个即插即用的框架。为了提高计算效率，在池操作中，我们对每个池操作只聚合一次特征。当边缘排序和折叠按顺序进行时，这种松弛允许在 GPU 上执行特征聚合操作，这提高了计算效率。

接下来，我们演示了 MeshCNN 在分类和分割任务上的性能。关于所使用的网络体系结构的详细信息见附录 A。

5.1 数据处理

在所有集合中，我们将每个网格简化为大致相同的边数。正如之前说的，MeshCNN 并不需要在所有样本中使用相同数量的边数。但如果网格与 CNN 图像的初始大小类似，将使得几何网格抽取有助于降低输入分辨率，并降低训练所需的网络容量。由于分类任务学习了全局形状描述，与分割任(2250 条边)相比，我们通常使用较低的分辨率(750 条边)。

Classification SHREC		
Method	Split 16	Split 10
MeshCNN	98.6	91.0%
GWCNN	96.6%	90.3%
GI	96.6%	88.6%
SN	48.4%	52.7%
SG	70.8%	62.6%

表 1. SHREC 分 30 类(比较自[2017])。16 和 10 是训练的分割, 训练多达 200 轮。

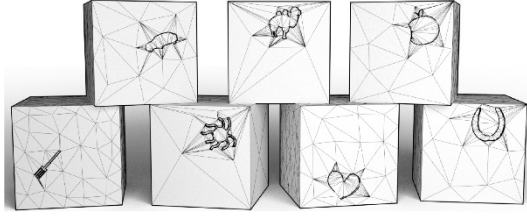


图 7. 雕刻立方体分类数据集。通过压缩 MPEG-7 [Latecki 和 Lakamper 2000]中的贴纸, 我们生成了 23 个不同的类别(例如, 汽车, 心脏, 苹果等), 并将它们放置在随机位置的随机表面上。

数据增强。有几种数据增强的方式可以为神经网络生成更多的数据样本。值得注意的是, 由于我们的输入特征是相似不变的, 在应用旋转、平移以及各向同性缩放(在 x 、 y 、 z 方向等因子)时, 不会生成新的输入特征。然而, 我们可以通过对 x 、 y 和 z 的顶点位置应用各向异性缩放来生成新的特征 $\langle S_x, S_y, S_z \rangle$ (每一个从正态分布中随机抽样, $\mu=1$ 和 $\sigma=0.1$), 或者将顶点移动到网格表面的不同位置以此来改变网络的输入特征。此外, 我们通过执行随机边缘翻转来增加每个对象的曲面细分。由于灵活的输入分辨率, 我们也可以在训练前折叠一个小的随机边缘集。

5.2 网格分类

SHREC 数据集。我们对 SHREC 数据集集中的 30 个类进行分类[Lian et al. 2011], 每个类有 20 个示例。按照[Ezuz et al. 2017]中的设置, 其中 16 和 10 是每个类的训练示例的数量, 训练 200 轮后停止。由于我们没有在[2017]中使用的确切分割, 我们的结果是在 3 个随机生成的 16 和 10 个示例的集合上取平均值。结果如表 1 所示。由[2017]中不难看出, 相比于 SG [Bronstein et al. 2011] (bag-of-features representation), SN [Wu et al. 2015] (volumetric CNN), GI [Sinha et al. 2016] (CNN on fixed geometry images), GWCNN [2017] (learned geometry images)我们的方法的优点是显而易见的。如图 6, 我们可视化了该数据集的网格池化的一些例子。我们观察到网格池化在一致的语义方式下表现出了一致性。(参见图 11)。

Cube Engraving Classification		
method	input res	test acc
MeshCNN	750	92.16%
PointNet++	4096	64.26%

表 2 立方雕刻数据集结果 (如图 7)

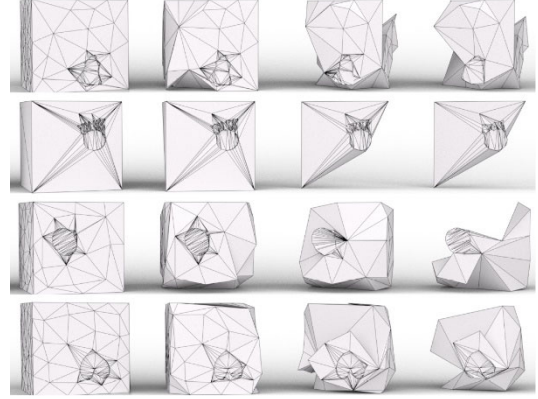


图 8. 训练 MeshCNN 来预测图标雕刻的类别。观察神经网络是如何在分类任务中保留重要边并去除冗余边的。

立方雕刻。为了展现 MeshCNN 的运用, 我们建模了一组带有浅图标雕刻的立方体(见图 7)。通过使用来自 MPEG-7 二进制形状 [Latecki 和 Lakamper 2000]数据集的 23 个类, 每个类大约有 20 个图标。每类留出三个图标用于测试集, 其余的用于训练集。对于每个图标, 我们随机抽取 10 个不同的状态(位置、旋转和立方体面)来插入图标。每个立方体大约有 500 片面, 这意味着简单形状相对于复杂形状来说在平面区域有着更多的三角面。这个集合包含了总共 4600 个形状, 其中 3910 个用于训练, 690 个用于测试。我们计划本文发表后公布该数据集及数据合成代码。

我们训练了 MeshCNN 来进行立方体分类。其定量结果如表 2 所示。为了可视化网格池化的分类任务的影响, 我们提取了每个网格池化后的中间结果(如图 8), 来观察 MeshCNN 如何学会在保留图标内部和边缘同时减少无关于网格划分任务(平面立方体表面)的边缘。

我们还在这个集合上训练了基于点的方法, 结果如表 2 所示。尽管这个例子很可能被认为是人为的, 但它是为了强调 MeshCNN 擅长于那些几何分辨率差异很大的 3D 形状。

COSEG Segmentation			
Method	Vases	Chairs	Telealiens
MeshCNN (<i>UNet</i>)	97.27%	99.63%	97.56%
MeshCNN (<i>rand. pool</i>)	96.64%	99.23%	97.01%
PointNet	91.5%	70.2%	54.4%
PointNet++	94.7%	98.9%	79.1%
PointCNN	96.37%	99.31%	97.40%

表 3. MeshCNN 在 COSEG segmentation 上的评估结果。

5.3 网格分割

MeshCNN 还可以应用在图形的一致性分割上，它是形状分析和合成中许多应用的重要组成部分。我们使用监督学习来训练 MeshCNN 来预测每条边在 COSEG [Wang et al. 2012] 和人体分割 [Maron et al. 2017] 数据集上属于特定片段的概率。由于这两个数据集都提供了每个表面的真值，我们基于原始分辨率的标签在简化的网格上生成边缘的语义标签。

最直接的 MeshCNN 语义分割配置是使用一系列的网格卷积层(以及标准化和非线性激活单元)。然而，合并网格池使 MeshCNN 能够学习分割驱动的边缘折叠。由于池化使得输入网格的分辨率降低且不再与真值的边缘标签一致。为此，我们使用网格逆池化层将分辨率上采样回原始输入大小。

COSEG 数据集。我们评估了 MeshCNN 在 COSEG 数据集上的分割性能，COSEG 数据集包含三个大集：外星人、花瓶和椅子，分别包含 200、300 和 400 个模型。我们将每个形状分成 85%/15% 的训练/测试部分。通过比较 PointNet, PointNet++ 和 PointCNN 三种方法，最后得出的各方法的最佳准确性如表 3 所示。可见 MeshCNN 在该数据集上的表现要优于其他方法。

我们认为我们方法比起其他来说更具优势的原因在于我们的神经网络是为网络结构量身定制的。为了进一步说明这一点，我们还报告了随机池(随机选取折叠边)情况下的结果，并表明这种变化降低了神经网络的性能。此外，如图 9 所示，我们给出了 MeshCNN 语义分割神经网络在给定测试集上使用池化层和非池化层的最终分割预测。这也表现了池化是如何来适应目标的。

人体分割数据集。我们在 [Maron et al. 2017] 提出的人体分割数据集上评估了我们的方法。该数据集由 SCAPE [Angelov et al. 2005]、FAUST [Bogo et al. 2014]、MIT [Vlasic et al. 2008] 和 Adobe Fuse [Adobe 2016] 提供的 370 个训练模型组成，测试集为来自 SHREC07 数

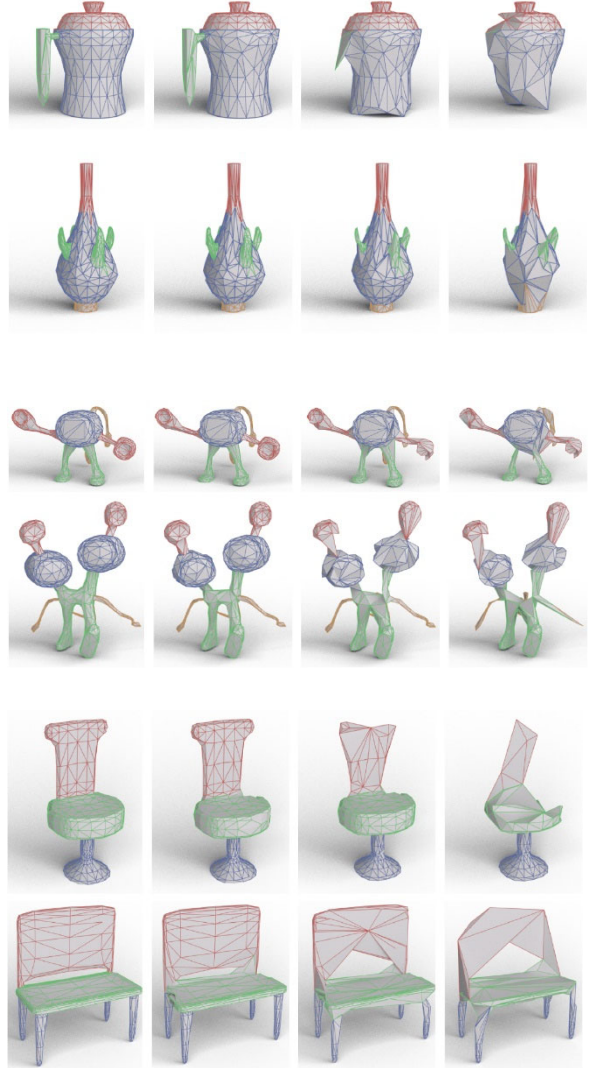


图 9. 语义分割的结果(保持)测试形状。左侧是每条边的分割预测，然后是每个池化层后的中间简化网格。为了可视化的目的，中间网格的边缘用最终的分割预测着色。MeshCNN 使用网格池来学习从相同的部分折叠边缘，这些部分通过网格解池层展开回原来的输入网格分辨率(例如，看最上面一行，整个花瓶把手已经折叠到一个边缘)。

据集的 18 个模型。根据 [Kalogerakis et al. 2010]，将模型人为分割成 8 个标签。最近，[Poulenard and Ovsjanikov 2018] 报告了该数据集的结果，并与 GCNN [Masci et al. 2015]，PointNet++ [Qi et al. 2017b]，Dynamic Graph CNN [Wang et al. 2018b]，和 Toric Cover [Maron et al.

Human Body Segmentation			[2018]
Method	# Features	Accuracy	
MeshCNN	5	92.30%	
SNGC	3	91.02%	
Toric Cover	26	88.00%	
PointNet++	3	90.77%	
DynGraphCNN	3	89.72%	
GCNN	64	86.40%	
MDGCNN	64	89.47%	

表 4. 人体分割结果(引用来自[Poulenard and Ovsjanikov 2018]的比较)。来自[Haim et al. 2018]的最新比较。



图 10 基于[Maron et al.2017]数据集的人体形状分割结果。

2017]进行了比较。在表 4 中我们列出了[Poulenard and Ovsjanikov 2018]的报告,在加入了 Haim 等人[2018]的最新成果后形成了该集合的最前沿的结果。在这种情况下,由于 MeshCNN 对网格结构和手头任务的适应性,比其他方法来说(部分基于图/流形,部分基于点)更有优势。图 10 展示了 MeshCNN 的一些定性结果。

5.4 其他评估

计算时间。对于训练边数规模在 2250/750 大小的分割/分类任务,我们采用的非优化 PyTorch [Paszke et al.2018]实现方法在配置一张 GTX 1080TI 显卡的条件下,平均每个样本只需花费 0.21/0.13 秒。

曲面细分鲁棒性。我们通过在 COSEG 分割数据集上进行定性与定量实验,检验了我们的方法对于三角剖分差异的鲁棒性。为此,我们生成了两个经过处理的数据集。其中一个数据集来源于网格重构程序(使用 Blender 和 MeshLab),另一个数据集中随机打乱 30% 的顶点的位置,实现方式就是将点向它随机的一个环顶点移动。我们可以从这结果(如表 5)的细微差别中看出曲面细分变化的弹性。图 12 中给出了定性的结果。

特征不变性。值得一提的是,使用相对特征可以保证 MeshCNN 在旋转,平移和等比例缩放时保持原来的功能。从本质上来讲,常用的笛卡尔坐标系对于刚体变换是比较敏感的,为了说明这一点,我们在语义分割任务中训练了 MeshCNN: (i) 使用不变的几何特征以及 (ii) 使用边的中点 (x,y,z) 作为输入特征。为了评估学习的泛化

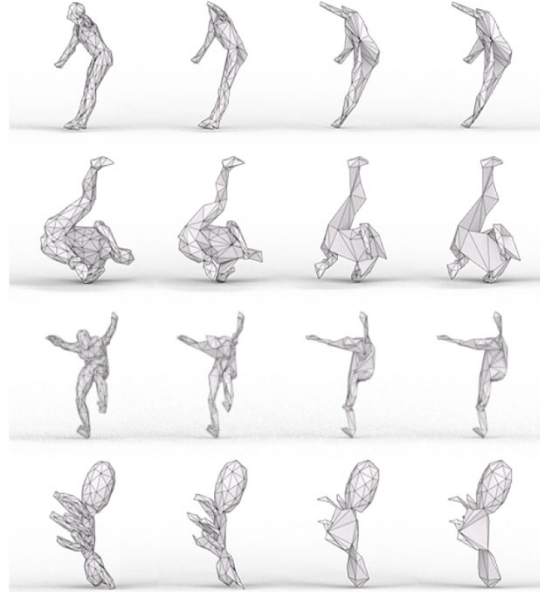


图 11. 自适应池化具有显示类中语义信息的潜能。我们发现,在形状分类任务 (SHREC) 中,相同类别中存在一致的语义池化。例如,人的头部被类似地池化,而其他相似的形状属性分别被池化到其他的类别。(底部)

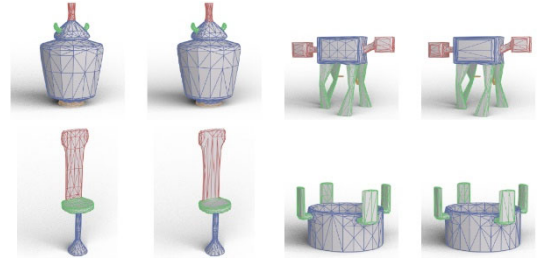


图 12. 不同三角剖分的鲁棒性 (语义分割)。左: 测试形状, 右: 交替三角剖分

Robustness			
Set	Vases	Chairs	Aliens
Reference	97.27	99.63%	97.56%
Remeshing	96.33%	88.98%	97.77%
Perturbations	97.12%	99.60%	97.34%

表 5. 两个鲁棒性集的定量结果。



图 13 差分特征的鲁棒性。我们使用具有几何特征的 MeshCNN 与标准 3D 点（边中点）进行对比，来评估拉伸集（左）上的泛化性。虽然两者在标准测试集上都达到了很高的精度，但是与 3D 点的情况相比，MeshCNN 的几何特征（右）可以很好地推广到拉伸集。

性，我们沿着坐标轴进行非均匀放缩（不对这些增加后的类型进行训练）。使用相对几何特征在标准测试集上达到了 99.63%，同样在新的测试集上达到了 98.44%。相比之下，使用绝对坐标在标准测试集上能达到 99.11%，却在新的测试集上退化到 78.27%。需要注意的是，虽然我们的几何特征对于非均匀放缩是不变的，但是由于对位置的不敏感性，它能够更好地体现其泛化能力。

6 讨论与未来工作

我们已经提出了一种在不规则三角网格上直接使用神经网络的一般方法，MeshCNN。我们工作的主要贡献在于给出了卷积和池化操作在非均匀规则结构下的全新定义以及应用场景。这些操作有利于以网格的原生形态来直接分析形状，因此能够受益于非均匀曲面流形表现形式的独特属性。

不变卷积。将网格边作为网络运行的基本构建块是十分重要的，因为边集为不规则结构的卷积提供了一个简单的定义了局部、固定大小的领域的方法。通过利用三角网格特有的对称性，我们可以消除相邻顺序边的对偶性，从而使变换结果保持稳定。在我们的精心设计下，我们使用相对几何属性而非绝对位置作为边特征输入来实现这一目的。因此，不同于常见的表示方法（例如：基于点），点的笛卡尔坐标被忽略，局部与全局特征脱离与位置关系的挂钩，这将更好地形状特征泛化，并且促进相似性变换的不变性。此外，我们强调我们只使用点的位置来展示网格的演化过程，对任务的完成过程并不产生影响。

空间自适应池化。我们提出了一种基于学习到的边特征，并通过折叠缩来进行的池化操作。从而在网络损失函数的指导下实现任务驱动池化。在未来，我们将添加类似于注意力机制模型的专门独立的功能集来对边折叠进行优先排序。将网络所确定的一系列特征进行可视化是非常重要的，这有助于我们深入理解网络实际学到的内容。同

时，我们发现，与使用绝对坐标不同的是，我们计算得到的微分特征不仅能够提供相似性变换的不变性，还能抑制过拟合。通过在不同对象上展示语义相似池化的能力，网络的泛化性得到了进一步的证明，同时也自然会得到更好的结果。从上面这些，我们可以发现研究这种强大的机制可能有助于更好地理解神经网络的行为。

我们认为空间自适应的不规则任务驱动池化是一个重要的贡献，它也有可能影响许多基于图片的 CNN 任务。例如，高分辨率图片分割通常会产生一个低分辨率分割映射，并且对其进行下采样，可能会使用残差连接。MeshCNN 的池化在语义上简化了具有统一特征的区域，同时保留了复杂特征。因此，在未来，我们致力于将类似的不规则池化应用于图像分割任务来获取高分辨率的分割映射，其中图像中较大的均匀区域将由少量的三角形表示。

目前，我们使用顺序的边折叠方法进行实现。通过对边特征使用并行排序技术[Bozidar and Dobravec 2015]，这个操作可以在一块 GPU 上并行化，并确保只有不相邻的边可以同时被折叠。显然，以这种非顺序化的方法池化得到的特征可能与顺序的到的特征不一样。

尽管我们的方法对于不同的三角剖分具有鲁棒性（正如我们的实验所证明的），MeshCNN 和其他网络一样需要依靠良好的数据集才能实现不错的泛化性。从这个角度上来看，MeshCNN 的性能很容易由于受到对抗网格攻击而降低，就像图像领域中的对抗攻击的影响。因此，对于应付这种对抗攻击的鲁棒性将成为我们未来工作一个十分有趣的方向。

另一个未来研究的方向是通过生成模型，网络上采样和特征合成来修改现有的网格。我们的想法是使用与边折叠相反顺序的边拆分，其中我们需要生成边折叠的顺序列表，这类似于逆池化的生成。因此，在合成新网格的时候，网络决定了拆分哪个点，例如拆分特征值较高的边的相邻点。

最后，在扩展我们的既定目标的时候，我们确定了一个非常具有前景的努力方向，而这个既定目标是为三角网格到一般的图而设计的。基于边折叠的池化和逆池化可以类似于我们提出的 MeshCNN 的方式应用于一般图。对于卷积，我们必须考虑一种适合于一般图的不规则性的合适替代方案。一种有趣的方法可能是使用注意力机制来处理边[Monti et al. 2018]。