arXiv:2305.04406v1 [math.OC] 8 May 2023

POLYTO：使用凸多边形进行结构拓扑优化

Aaditya Chandrasekhar机械工程系 威斯康星大学麦迪逊分校 麦迪逊，威斯康星州 53706
cs.aaditya@gmail.com

2023年5月9日

摘要

本文提出了一种拓扑优化（TO）框架，其中设计由一组凸多边形参数化。通过扩展拓扑优化中的特征映射方法，该表示方式能够直接提取几何形状。此外，该方法允许直接在多边形上施加几何约束，例如特征尺寸控制，而这些约束在基于密度或水平集的方法中难以实现。与杆件、板件或圆形等较简单的基本几何元素相比，采用多边形能够提供更加丰富多样的形状。多边形被定义为一组半空间的可行集。通过改变半空间的参数，可以获得多种不同构型的多边形。此外，半空间通过可微映射映射到背景网格上，从而实现分析和梯度驱动的优化。通过大量二维结构柔顺度最小化拓扑优化示例，展示了所提框架的应用。最后，还总结了该框架的一些主要局限性及未来研究方向。

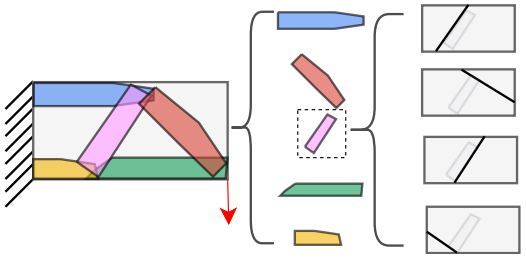


图 1：由一组凸多边形构成的优化设计。每个多边形又由一组半空间的可行集定义。

关键词 拓扑优化 · 特征映射 · 凸多边形

1 引言

拓扑优化（TO）是一种在设计域内优化分配材料的策略 [1]，旨在优化所需目标函数，同时满足一个或多个约束。拓扑优化已发展成熟，现已应用于商业领域 [2]，用于探索新颖且最优的设计。虽然已有多种拓扑优化方法被提出，但基于密度 [3] 和水平集 [4] 的方法已得到广泛采用。这些方法能够在广泛的应用领域中生成复杂、仿生的设计。然而，生成的设计往往无法直接制造，需要大量后处理 [5]，或在优化过程中施加相应的制造约束 [6]。对拓扑优化设计进行后处理可能会显著降低其性能；在诸如面向新型

应用的设计中，例如柔顺机构 [7, 8]，甚至可能使设计完全丧失功能。这使拓扑优化更多地被视为一种“概念工具”，限制了其在工程工作流程中的全部潜力。虽然施加制造约束可以提高可制造性，但制定这些约束需要大量研究投入，也是当前许多研究工作的重点 [9, 10, 11]。最后，尽管增材制造（AM）有望直接制造拓扑优化生成的自由形状设计，但仍需满足增材制造特有的制造约束 [12],，许多问题尚未解决[13]，增材制造也尚未得到更广泛的应用。

与此同时，人们也开发了拓扑优化技术，并认识到如今制造的组件中，有相当一部分是通过组合几何形状较为简单的原材料制成的。这类方法统称为特征映射方法 [14],，包括移动可变形空洞方法 [15], 、移动可变形组件方法 [16], 、自适应气泡法 [17], 以及几何投影方法 [18] 等拓扑优化技术。这些技术的一个显著特点是，设计由高层次几何描述参数化，其特征会映射到网格上以进行分析。这与基于密度和水平集的方法形成对比；在后者中，设计通过像素/体素进行冗长的参数化。有关这些方法的详细综述，请参阅 [14]。

通过高层次几何描述对设计进行参数化，可以简化对特征存在与否及其尺寸相关规则的实施。此外，尺寸标注与公差、设计变异量化与控制等各种生产考量，都需要对几何形状进行高层次描述。最后，通过简单的对偶变换 [19]，它们可以轻松转换为多边形网格。 [20] 这与基于像素的表示法形成对比；后者需要执行计算量很大的等值面提取操作（例如行进立方体）来提取轮廓。

1.1 贡献

本文提出了对特征映射方法的扩展，采用凸多边形对设计进行参数化。与杆件和板件等较简单的几何基元相比， [21], 这种参数化方式在保留几何基元可恢复性的同时，提供了更大的设计灵活性。基元固有的凸性保证进一步提升了可制造性。与 [22], 不同，我们展示了如何利用所提参数化方法轻松推导并施加特征尺寸约束。此外，有符号距离场采用隐式定义，计算起来也很直接。

本文其余部分安排如下。第2.2节介绍所提出的设计参数化方法。第2.3节介绍有符号距离场（SDF）的计算，随后将SDF投影到密度场中进行分析。第2.5节和第2.6节分别介绍优化问题的建模和灵敏度分析。我们在第2.7节总结完整算法。第3节介绍若干展示所提方法的参数化研究。最后，我们在第4节讨论一些局限性及未来研究方向。

2 所提方法

2.1 概述

我们假设设计域、载荷和约束均已给定。此外，我们假设超参数——多边形的数量（K）以及每个多边形的最大边数（S）——均已指定。因此，目标是找到一种多边形构型，使其形成最优设计。为简便起见，我们将讨论范围限定为体积约束下的最小柔度设计优化。

2.2 设计参数化

假设我们的设计将使用 $K(\geq 1)$ 个凸多边形来定义。此外，假设每个多边形最多有 $S(\geq 3)$ 条边。注意，凸多边形可以定义为一组半空间 [23] 所构成的可行集。请记住，虽然由一组半空间定义的可行集必定是凸的，但无法保证它有界或非空。为避免出现无界或空区域，我们采用一种保守的参数化方式，而不是通过半空间的法向量及其到原点的距离来参数化几何形状。具体而言，每个多边形 p_i 的参数化如下：

$$p_i = \{c_x^{(i)}, c_y^{(i)}, \alpha^{(i)}, d_1^{(i)}, \ldots, d_S^{(i)}\} \quad , \; i = 1, \ldots, K \tag{1}$$

其中 $c_x^{(i)}$， $c_y^{(i)}$分别是该 i^{th} 多边形的 x 和 y 中心坐标， $d_j^{(i)} \in \mathrm{R}^+$ 是该 i^{th} 多边形的j^{th} 半空间到其中心的距离。我们将 α_i 定义为该 i^{th} 多边形的整体角偏移，其 j^{th} 半空间的方向定义为 $\theta_j^{(i)} = \alpha_i + {}^{2\pi}{}_{(j-1)\,S}$ （图 2）。注意，虽然这种方向定义限制了设计空间，但它确保多边形有界且非空。

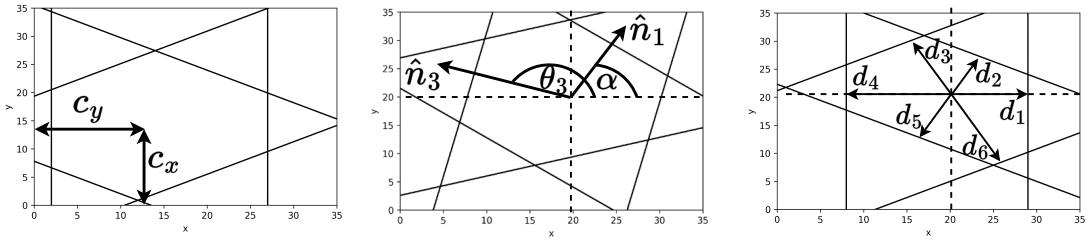


图2：该（六边形）多边形由中心坐标（c_x, c_y）、角偏移（α）以及各半空间到中心的距离（$d_1, \ldots, d_6$）。

2.3 几何投影

几何投影的核心思想，是将由多边形参数（第2.2节）描述的设计映射到定义在网格 [19] 上的密度场。密度函数 $\rho(x)$ $x \in \mathbb{R}^2$ 的定义方式为： $\rho(x) = 0$ 定义空域，而 $\rho(x) = 1$ 定义实体区域。当一个点位于至少一个多边形的可行区域内时，该点即被视为实体。这种映射使得能够在网格上进行后续的结构分析，而这正是优化所需的。此外，该映射具有可微性，因此可以定义目标函数和约束的灵敏度，从而实现基于梯度的优化。

我们首先计算域中一点 $(x,\ y)$ 处 i^{th} 多边形的 j^{th} 半空间的有符号距离场（SDF）（ $\hat{}\phi_j^{(i)}(\cdot)$），如下所示（式2）：

$$\hat{\phi}_j^{(i)}(x, y; c_x^{(i)}, c_y^{(i)}, \theta_j^{(i)}, d_j^{(i)}) = (x - c_x^{(i)})\cos(\theta_j^{(i)}) + (y - c_y^{(i)})\sin(\theta_j^{(i)}) - d_j^{(i)} \tag{2}$$

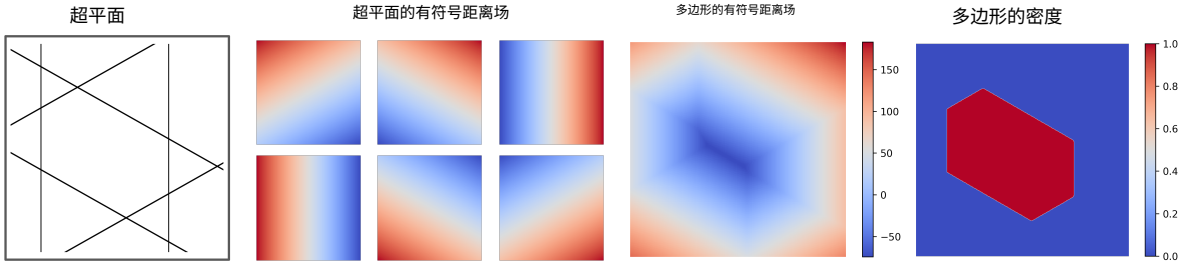


图3：从半空间到密度

定义了半空间的有符号距离场后，可以通过对各平面的有符号距离场取交集（最大值）来计算凸多边形的近似有符号距离场。为了便于基于梯度的学习，我们使用平滑最大值函数对数和指数（LSE）代替最大值。因此， i^{th} 多边形的近似有符号距离场（$\phi(\cdot)$）可表示为（式 3）：

$$\phi^{(i)}(x, y; c_x^{(i)}, c_y^{(i)}, \alpha^{(i)}, d_1^{(i)}, \ldots, d_S^{(i)}) = \mathrm{LSE}(\hat{\phi}_1^{(i)}(x, y), \ldots, \hat{\phi}_S^{(i)}(x, y)) \tag{3}$$

回顾一下，有符号距离场（SDF）定义了域中任意点到物体边界的最短距离。对于多边形内部的任意点，SDF 为负；在边界上为零；对于

多边形外部的点，SDF 为正。此外，请注意，虽然精确的 SDF 满足 $||\nabla\phi(\boldsymbol{x})|| = 1$， $\forall \boldsymbol{x}$ 但近似 SDF [19] 未必满足这一条件。接下来，我们使用 Sigmoid函数，根据 $\phi(\cdot)$ 计算多边形的密度场 $\hat{}\rho(\cdot)$：

$$\hat{\rho}^{(i)}(x, y; \phi^{(i)}) = \frac{1}{1 + e^{-\beta\phi^{(i)}(x,y)}} \tag{4}$$

注意，有符号距离场的负值会被映射为密度 1，正值则映射为 0。接近零的有符号距离场值会被映射为介于两者之间的密度值 $\rho \in (0, 1)$，过渡的锐度由超参数 β 控制。最后，计算出每个多边形 $\{\hat{}\rho^{(1)}, ..., \hat{}\rho^{(K)}\}$ 的密度场后，即可通过 a q^{th} 范数公式推导出设计的密度场，如下所示（公式 5，见图 4）：

$$\rho(x, y; \hat{\rho}^{(1)}, \ldots, \hat{\rho}^{(K)}) = \left(\sum_{k=1}^{K} \left(\hat{\rho}^{(k)}(x, y) \right)^{q} \right)^{\frac{1}{q}} \tag{5}$$

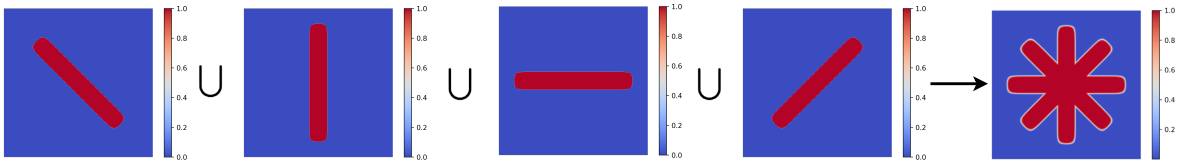


图 4：设计的密度场由多边形的密度场计算得出。

2.4 有限元分析

获得设计的密度场后，我们使用固体各向同性材料惩罚（SIMP）方案来确定杨氏模量 E。令 ρ_e 为在所有 N_e 个单元中，单元 e 中心坐标处计算得到的密度，则可得到对应的 E_e：

$$E_e(\rho_e) = E_{min} + (E_0 - E_{min})\rho_e^p \tag{6}$$

其中 E_0 为基体材料的杨氏模量， E_{min} 为防止全局刚度矩阵奇异而添加的小常数， p 为SIMP惩罚因子。

对于有限元分析，我们使用由双线性四边形单元构成的结构化网格求解线弹性问题。单元刚度矩阵可计算为

$$[K_e] = E_e \int_{\Omega_e} [B]^T [D_0][B] d\Omega_e \tag{7}$$

其中， $[B]$ 为应变矩阵， $[D_0]$ 为本构矩阵，其杨氏模量为1，泊松比取假设值（见表1）。随后组装全局刚度矩阵 $\boldsymbol{K}$。最后，给定节点力向量 $\boldsymbol{f}$，求解状态方程以获得节点位移 $\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{f}$。

2.5 优化问题

本文研究结构部件的拓扑优化。下面将讨论优化公式的各个组成部分。

优化器： 我们采用移动渐近线法（MMA） [24] 来执行设计更新。具体而言，我们使用 Python 实现 [25] ，其中所有默认参数均对应于[26]中介绍的 MMA 版本。关于 MMA移动限值、步长容差和 Karush-Kuhn-Tucker（KKT）容差的取值 [27]，请参见表1。

优化变量： 设计由多边形的中心坐标、角偏移和平面距离定义，共有 K个（$S+3$）自由参数（第2.2节）。进行优化时，我们定义一个增广归一化设计向量$\boldsymbol{z}=[\boldsymbol{z}_{c_x},\boldsymbol{z}_{c_y},\boldsymbol{z}_{\alpha},\boldsymbol{z}_{d},]$，其中各项均位于 $[0,1]$内。对于 c_x、c_y、α、d，它们分别是 x中心、y中心、角偏移和平面距离的下界；对于 c_x、c_y、α、d，它们分别是上述参数的上界。因此，我们可以将未归一化的 x中心还原为$\boldsymbol{c_x} \leftarrow c_x + (c_x - c_x)\ \boldsymbol{z_{c_x}}$，其他参数也依此类推。

目标函数： 我们采用一个简单的柔顺度最小化问题。根据节点位移 $\boldsymbol{u}$（第2.4节）以及施加载荷 $\boldsymbol{f}$，计算柔顺度 $J=\boldsymbol{f}^T\boldsymbol{u}$。

体积约束： 设计需满足总体积约束（式8），其中 v_f^*为允许的最大体积分数，ρ_e 为单元密度，v_e 为单元面积，Ω_0 为域的体积

$$g_v \equiv \frac{\sum_{e=1}^{N_e}\rho_e v_e}{v_f^*|\Omega_0|} - 1 \leq 0 \tag{8}$$

最小长度约束： 为提升可制造性，也可以对多边形的特征尺寸施加约束。具体而言，我们考虑对多边形的边长施加一个最小下限。请看图5，其中展示了一个边长为 l_1、l_2、$\ldots l_6$ 的六边形。一般而言，根据我们的参数化方式（第2.2节），可通过考虑两个包围四边形来推导这些长度（式9）：

$$l_i^{(j)} = \frac{1}{\sin\gamma}[d_{i+1}^{(j)} + d_{i-1}^{(j)} - 2d_i^{(j)}\cos\gamma] \tag{9}$$

其中，$\gamma=\frac{2\pi}{S}$，而 $d_i^{(j)}$、$d_{i-1}^{(j)}$、$d_{i+1}^{(j)}$ 分别是候选 i 和多边形的循环相邻垂直距离。计算出所有多边形所有边的长度后，最短边的近似长度可微分地表示为：

$$l_{min} = \min_{\forall i,\forall j} l_i^{(j)} \approx -LSE_{\forall i,\forall j}(-l_i^{(j)}) \tag{10}$$

最后，随着 l^* 被 根据施加的最小边长，可以表示最小边长约束 a s:

$$g(l) \equiv 1 - \frac{l_{min}}{l^*} \leq 0 \tag{11}$$

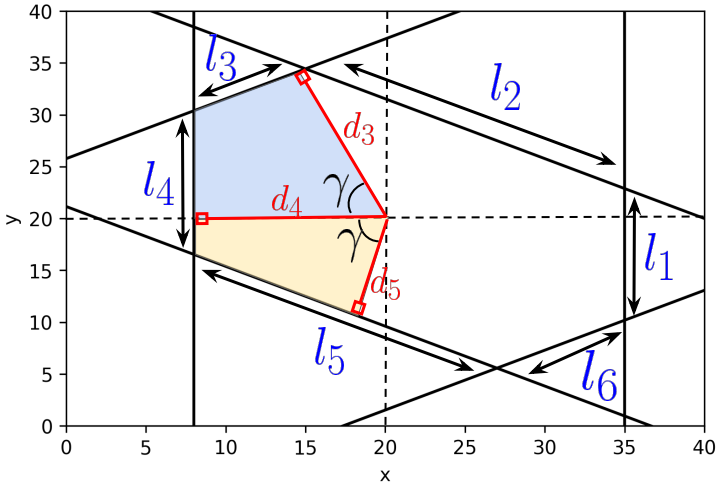


图5：多边形各边的长度（$l_1, l_2, \ldots, l_6$）需受最小界限约束。任意一条边（此处为 l_4）的长度都可以通过分析两个包围四边形（蓝色和黄色所示）得出。

可以用类似的论证构造多边形最大特征尺寸的约束，但本文不作进一步探讨。

综合目标函数、状态方程、体积、最小长度和盒约束，优化问题可写为（方程（12））：

$$
\begin{aligned}
&\underset{\boldsymbol{z}}{\text{最小化}} && J = \boldsymbol{f}^T \boldsymbol{u}(\boldsymbol{z}) && \text{(12a)}\\
&\text{约束条件为} && \boldsymbol{K}(\boldsymbol{z})\boldsymbol{u} = \boldsymbol{f} && \text{(12b)}\\
& && g_v(\boldsymbol{z}) \leq 0 && \text{(12c)}\\
& && g_l(\boldsymbol{z}) \leq 0 && \text{(12d)}\\
& && 0 \leq z_i \leq 1 \quad \forall i && \text{(12e)}
\end{aligned}
$$

2.6 灵敏度分析

基于梯度的优化更新方案中，一个关键要素是目标函数和约束相对于优化参数的灵敏度，即导数。通常，灵敏度分析需要手动进行。这可能既费时又容易出错，尤其是对于复杂的目标函数。在这里，我们使用 JAX [28], 表达包括有符号距离场、投影、有限元分析、目标函数和约束在内的所有计算，利用其自动微分（AD）功能将这一过程完全自动化 [29, 30]。随后，我们可以分别用公式 13、14 和 15 表示目标函数、体积和最小长度尺度约束相对于优化变量的导数。

$$
\frac{\partial J}{\partial \boldsymbol{z}} = \frac{\partial J}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{K}} \frac{\partial \boldsymbol{K}}{\partial \boldsymbol{E}} \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \hat{\boldsymbol{\rho}}} \frac{\partial \hat{\boldsymbol{\rho}}}{\partial \boldsymbol{\phi}} \frac{\partial \boldsymbol{\phi}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{z}} \tag{13}
$$

$$
\frac{\partial g_v}{\partial \boldsymbol{z}} = \frac{\partial g_v}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \hat{\boldsymbol{\rho}}} \frac{\partial \hat{\boldsymbol{\rho}}}{\partial \boldsymbol{\phi}} \frac{\partial \boldsymbol{\phi}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{z}} \tag{14}
$$

$$
\frac{\partial g_l}{\partial \boldsymbol{z}} = \frac{\partial g_l}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{z}} \tag{15}
$$

2.7 算法

完整算法　　所提框架的算法总结于算法1，并在图6中以示意图的形式展示　　.

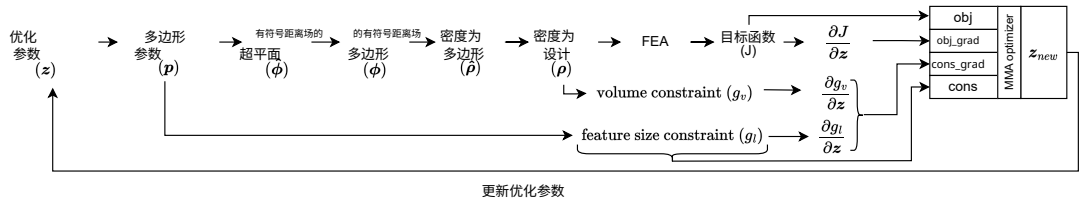


图6：所提框架的优化循环。

3 数值结果

本节通过若干实验展示所提框架。实验中的默认参数汇总于表1。除第3.5节中的实验外，均未使用最小长度尺度约束。所有实验均在 MacBook M1 Pro 上使用 JAX 库 [28] 在 Python 中进行。

3.1 验证

我们考虑 [31] 中展示的示例，其中研究了采用 $S = 16$ 的单个多边形对中部悬臂梁（图 7）进行设计，以实现 $l_x = l_y = 2$ 的 $v_f^* = 0.5$。我们报告其演变过程以及

算法1POLYTO：使用凸多边形进行拓扑优化

$i \leftarrow 0$ ▷ 迭代计数器
$\boldsymbol{z}^{(0)} \leftarrow \boldsymbol{p}^{(0)}$ ▷ 初始化优化变量
收敛 = False
当收敛 == False **执行**
  $\boldsymbol{p}^{(i)} \leftarrow \boldsymbol{z}^{(i)}$ ▷ 从优化变量中获取设计
  计算最小长度约束 g_l ， 基于 $\boldsymbol{p}^{(i)}$ ▷ 公式 9、10、11
  对于 $k = 1, \ldots, K$ **执行** ▷ 每个多边形
    对于 $s = 1, \ldots, S$ **执行** ▷ 每个半空间
      对于 $e = 1, \ldots, N_e$ **执行** ▷ 每个单元
        计算从 $\boldsymbol{x}_e$ 到多边形 k 的边 s 的有符号距离 $\hat{}\phi_s^{(k)}(\boldsymbol{x}_e)$ ▷式 2
      结束循环
    结束循环
    计算多边形 $\phi^{(k)}(\boldsymbol{x}_e)$的有符号距离 ▷ 公式 3
    计算多边形的投影密度 $\hat{}\rho^{(k)}(\boldsymbol{x}_e)$ ▷ 公式 4
  结束循环
  用于 $e = 1, \ldots, N_e$ **do**
    计算密度 ρ_e的并集 ▷ 式 5
    计算弹性模量 E_e ▷ 公式 6
    计算单元刚度矩阵 $\boldsymbol{K}_e$ ▷ 公式 7
  结束循环
  计算组装刚度矩阵 $\boldsymbol{K}$ 并求解 $\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{f}$ ▷ 第2.4节
  计算目标函数 J 和体积约束 g_v ▷ 第 2.5 节
  计算 $\nabla_{\boldsymbol{z}} J, \nabla_{\boldsymbol{z}} g_v, \nabla_{\boldsymbol{z}} g_l$ ▷ 公式 13、14、15
  $\boldsymbol{z}^{(i+1)} \leftarrow$ 移动渐近线法$(J, g_v, g_l, \nabla_{\boldsymbol{z}} J, \nabla_{\boldsymbol{z}} g_v, \nabla_{\boldsymbol{z}} g_l)$ ▷ 更新设计
  $i \leftarrow i + 1$ ▷ 递增计数器
  $\Delta_s \leftarrow ||\boldsymbol{z}^{(i)} - \boldsymbol{z}^{(i-1)}||$ ▷ 计算步长容差
  计算 KKT 条件的范数 Δ_{kkt}
  如果 $k \geq$ maxIter 或 $\Delta_s \leq$ stepTol 或 $\Delta_{kkt} \leq$ kktTol **则** ▷ 收敛标准
    已收敛 = True
  结束 if
结束 while
循环

表1：默认参数

参数	描述和默认值
E, ν	具有杨氏模量的各向同性材料 $E = 1$和泊松比 $\nu = 0.3$
nelx，nely	（100，50）方向的有限元分析网格单元数量
l_x，l_y	分别沿 x 和 y 沿 x 和 y方向的域长度（60，30）
$(\underline{c}_x, \bar{c}_x)$	分别 多边形 x中心范围 (0，l_x)
$(\underline{c}_y, \bar{c}_y)$	多边形 y中心范围 (0，l_y)
$\underline{d}, \bar{d}$	多边形面偏移范围 (0，$0.5l_x$)
p	SIMP惩罚因子参数 = 3
q	密度并集惩罚参数 = 8
β	密度投影中的锐度参数 = 10
最大迭代次数	最大优化迭代次数 = 250
移动限值	MMA 步长 $= 5E-2$
KKT 容差	MMA 收敛准则 $= 1E-6$
步长容差	MMA 收敛准则 $= 1E-6$

最终拓扑结构见图 8。图中也展示了参考文献中的拓扑结构。我们观察到，优化拓扑结构和最终柔顺度与参考文献一致，这验证了我们的方法与已有研究结果相符。请注意，与 [31], 中一样，图中仅显示多边形位于设计域内的部分，因为域外的设计不会影响

目标函数和约束。将多边形完全限制在设计域 [32] 内将作为未来的研究内容。

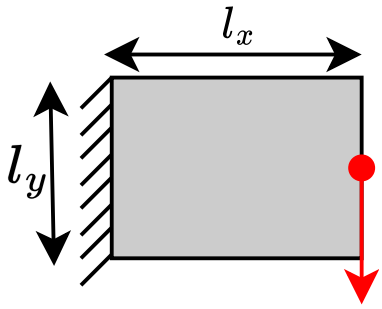


图7：中部悬臂梁

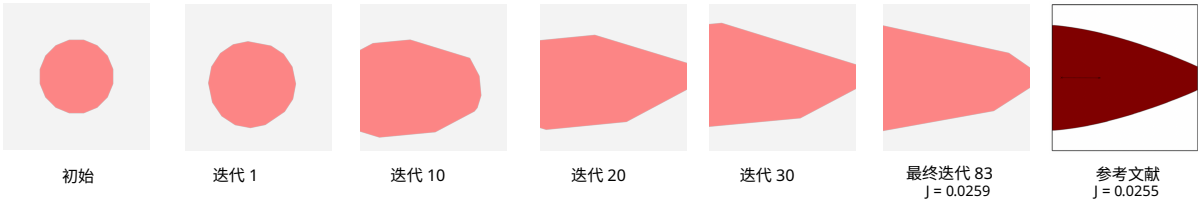


图8：具有单个多边形的中部悬臂梁的设计演化。最终结果与参考文献 [31] 中的结果进行了比较

接下来，我们将所得结果与基于密度的方法得到的结果进行比较。图9展示了使用所提框架（$K = 12$ 和 $S = 6$）获得的设计与使用 [33] 获得的设计之间的对比，研究对象为采用默认参数和 $v_f^* = 0.5$ 的中部悬臂梁。我们观察到，优化后的设计具有相似的性能。请注意，多边形的透明度仅用于示意。

我们还展示了图10所示问题中柔顺度和体积约束的收敛过程。此外，还展示了优化过程中不同阶段的设计快照。所有其他示例也观察到了类似的收敛行为。

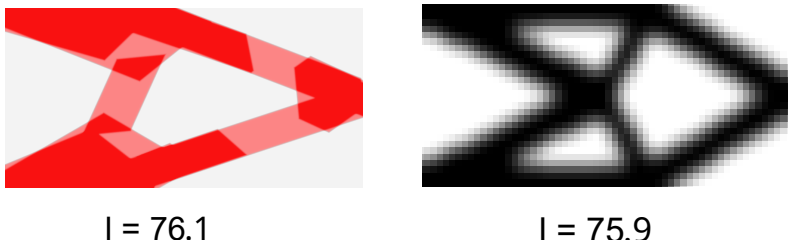


图9：与 [34] 的拓扑结构和柔顺度比较。

3.2 帕累托权衡

设计阶段的一个重要考虑因素是探索帕累托前沿，评估各种设计选择之间的权衡。以图 11 中的 MBB梁问题为例。通过 $K = 6$和 $S = 12$，我们探索结构的柔顺度与体积分数之间的权衡。图 12 展示了得到的帕累托最优前沿以及不同情况下的拓扑结构。不出所料，随着允许的体积分数降低，柔顺度会增加。

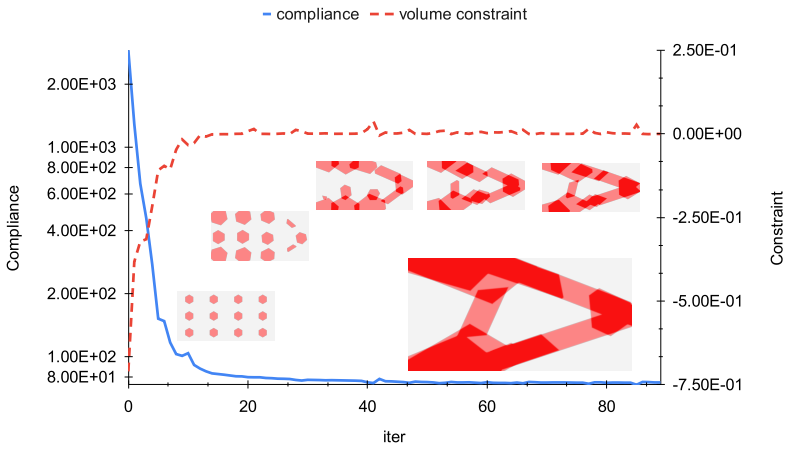


图10：中部悬臂梁的柔顺度（目标函数）与体积约束的收敛情况。展示了第0、5、10、30、50次迭代以及最终迭代的设计结果。

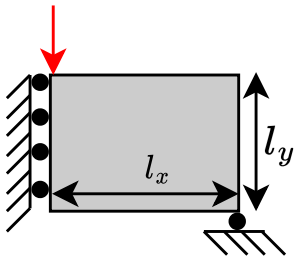


图 11：MBB梁。

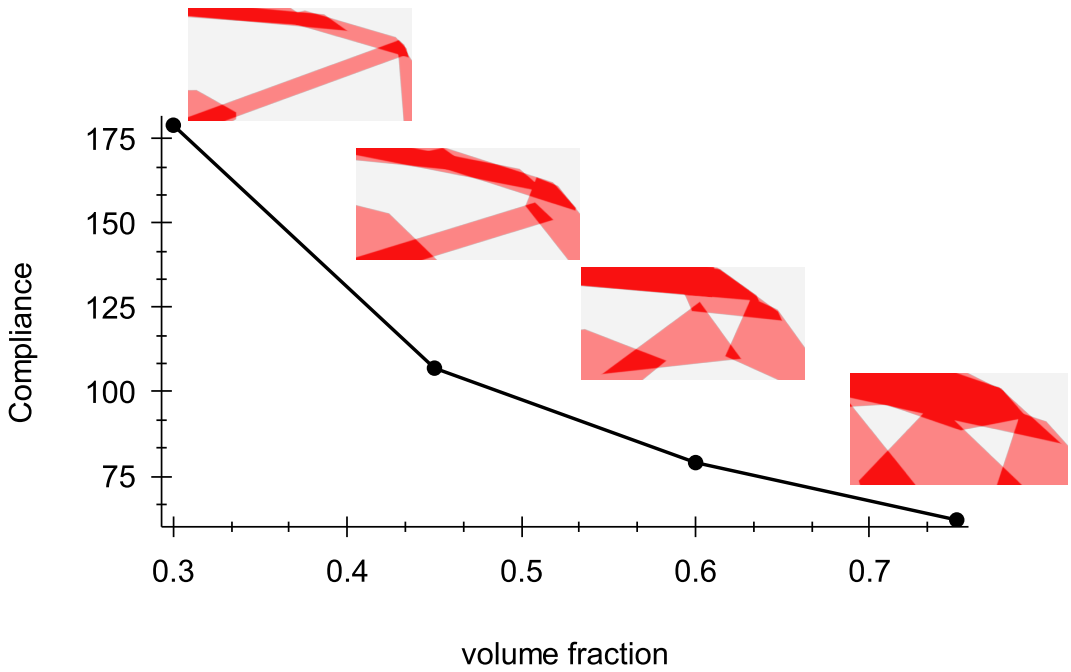
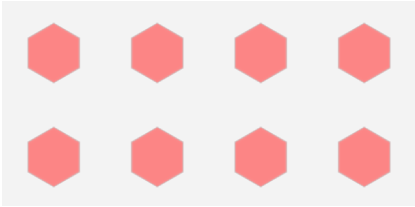


图 12：柔顺度与体积之间的权衡。



(a) 由（K =）个8六边形（$S = 6$）组成的初始设计，位于 4×2 网格上。



(b) 不同初始化下的MBB梁设计。

图13：初始化对所得设计和 的影响，以及 柔顺度。

3.3 初始化的影响

在本实验中，我们考察初始设计对优化结果的影响。由于我们的优化问题具有非凸、非线性的特性，我们预计不同的初始化会导致不同的局部最优解。通常，我们从由等间距、等尺寸多边形组成的网格开始优化。例如，图13a展示了在 4×2 网格上初始化的8个六边形。在此示例中，我们使用超参数 $K = 8$、 $S = 6$ 和 $v_f^* = 0.5$对MBB梁（图11）进行设计。图13b比较了不同初始构型所得到的拓扑结构和性能。我们发现，尽管得到的拓扑结构各不相同，但性能相近。

3.4 多边形参数的变化

接下来，我们考察两个超参数的影响，即多边形的数量（K）和每个多边形中的超平面数量（S）。通过改变多边形的数量以及用于参数化多边形的半空间数量，可以控制设计的复杂度。再次考虑图 11 所示的 MBB梁。在 $v_f^* = 0.5$ 条件下，我们考察图 2 中改变 K 和 S 的影响。对于我们所研究的问题，以及一般的结构构件柔顺度最小化问题，我们没有观察到性能或拓扑结构存在显著差异；不过，对于其他目标函数和物理学问题，这些参数可能会很重要。

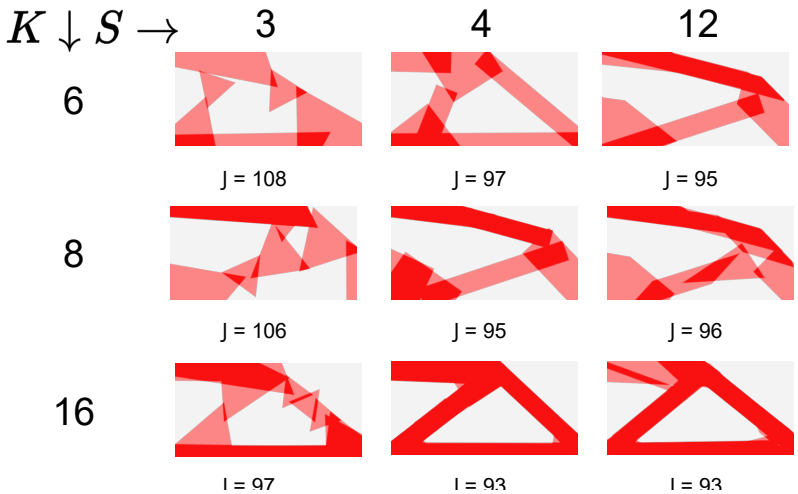
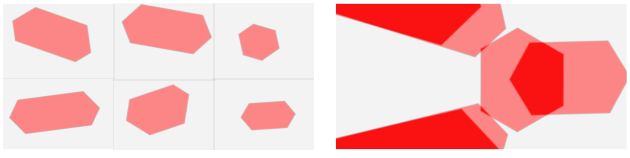
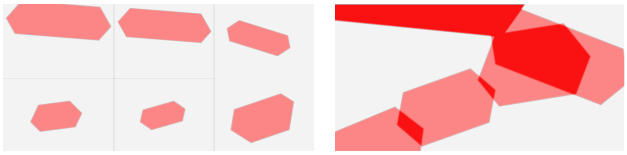


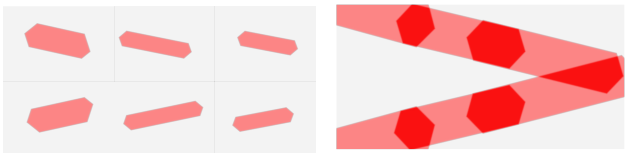
图14：多边形数量（K）和边数（S）对所得拓扑结构和性能的影响。



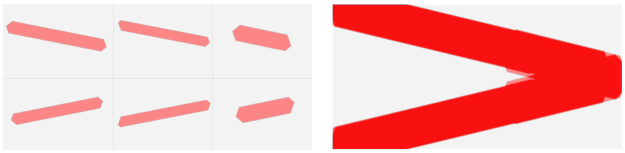
(a) 允许的最小长度 $l^* = 8$



(b) $l^* = 6$的允许的最小长度



(c) $l^* = 4$的允许的最小长度



(d) $l^* = 2$允许的最小长度

图 15：不同允许的最小长度下的多边形（左）和所得设计（右）

3.5 最小长度尺度约束

接下来，我们考察最小长度尺度约束（式 11）对所得拓扑结构和性能的影响。再次考虑图 7 中的中部悬臂梁。图 15 展示了在 $K = S = 6$ 和 $v_f^* = 0.5$ 条件下，允许的最小长度为 l^* 时的拓扑结构及填充多边形。请注意，允许的最小长度越大，约束就越严格。正如预期，随着我们施加更严格的约束，多边形变大，性能则有所下降（图 16）。

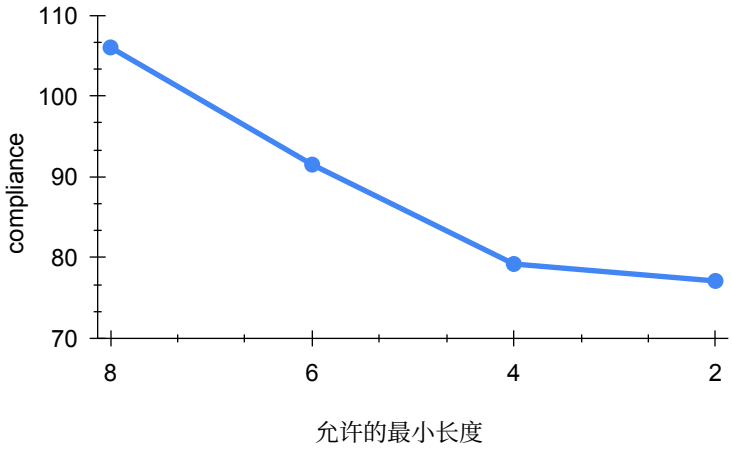


图 16：允许的最小长度对性能的影响。

4 结论

本文展示了一种通过凸多边形参数化设计来实现拓扑优化的方法。作为拓扑优化中特征映射方法的一种自然扩展，多边形能够容纳比杆件和板件等简单基元更多样的形状。这使设计自由度得以提高，从而获得更优的解决方案，同时保留特征映射方法的优势。几何形状的复杂程度可通过多边形的数量以及参数化每个多边形的半空间数量来控制。我们在结构优化的背景下介绍了该框架，以获得满足体积约束的高刚度设计。我们还解析地推导并约束了特征尺寸。数值示例表明，与基于像素的参数化方法相比，该方法能够生成性能相当的设计。

我们假设边的数量和多边形的数量是优化前给定的超参数。将这些数量纳入优化参数，可能有助于获得更优且自动化程度更高的设计。此外，为确保始终得到有界且非空的多边形，我们固定了半空间之间的夹角。未来的工作将探索角度自由参数化的技术。虽然本文仅讨论了二维情形，但该技术可轻松扩展至三维。尽管我们探讨了对单个基元进行特征尺寸控制，但基元重叠可能产生大于指定最小尺寸的特征。除最小特征尺寸约束外，还可施加重叠约束来解决这一问题 [35]。为进一步提高可制造性，未来将探讨悬垂约束 [36] 等其他约束。还可以通过简单地改变半空间的距离来侵蚀或膨胀多边形，进而将其纳入稳健分析框架[37]。最后需要指出的是，计算有符号距离场的成本可能较高。采用基于机器学习的方法 [38] 来获取有符号距离场，或可解决这一问题。

柔顺度与伦理标准

作者声明不存在利益冲突。

结果复现

与本文相关的 Python 代码可在 github.com/aadityacs/PolyTO 找到

参考文献

[1] Ole Sigmund 和 Kurt Maute。拓扑优化方法：比较综述。《结构与多学科优化》，48(6):1031–1055，2013年。[2] Liang Meng、Weihong Zhang、Dongliang Quan、Guanghui Shi、Lei Tang、Yuliang Hou、Piotr Breitkopf、Jihong Zhu 和 Tong Gao。从拓扑优化设计到增材制造：今日的成果与明日的路线图。*Archives of Computational Methodsin Engineering*，27:805–830，2020年。[3] Martin Philip Bendsoe 和 Ole Sigmund。拓扑优化：理论、方法与应用。Springer Science & Business Media，2003年。[4] Michael Yu Wang、Xiaoming Wang 和 Dongming Guo。用于结构拓扑优化的水平集方法。*Computer methods in applied mechanicsand engineering*，192(1-2):227–246，2003年。[5] Subodh C Subedi、Chaman Singh Verma 和 Krishnan Suresh。拓扑优化模型几何后处理方法综述。*Journal of Computing and InformationScience in Engineering*，20(6)，2020年。[6] Sandro L Vatanabe、Tiago N Lippi、Cícero R de Lima、Glaucio H Paulino 和 Emílio CN Silva。带制造约束的拓扑优化：一种统一的基于投影的方法。*Advances in EngineeringSoftware*，100:97–112，2016年。[7] Benliang Zhu、Xianmin Zhang、Hongchuan Zhang、Junwen Liang、Haoyan Zang、Hai Li 和 Rixin Wang。采用基于连续体的拓扑优化设计柔顺机构：综述。*Mechanismand Machine Theory*，143:103622，2020年。[8] Benliang Zhu、Xianmin Zhang 和 Nianfeng Wang。采用水平集方法优化具有多个输出的无铰链柔顺机构的拓扑。《结构与多学科优化》，47:659–672，2013年。[9] Martin F Schubert、Alfred KC Cheung、Ian AD Williamson、Aleksandra Spyra 和 David H Alexander。在严格代工厂制造约束下对光子器件进行逆向设计。*ACS Photonics*，9(7):2327–2336，2022年。

[10] Alec M Hammond、Ardavan Oskooi、Steven G Johnson 和 Stephen E Ralph。在半导体代工厂设计规则约束下进行光子拓扑优化。Optics Express，29(15):23916–23938，2021年。[11] Alok Sutradhar、Jaejong Park、Payam Haghighi、Jacob Kresslein、Duane Detwiler 和 Jami J Shah。在拓扑优化方法中纳入制造约束：综述。载于International Design Engineer- ing Technical Conferences and Computers and Information in Engineering Conference，第58110卷，第V001T02A073页。美国机械工程师学会，2017年。[12] Mary Kathryn Thompson、Giovanni Moroni、Tom Vaneker、Georges Fadel、R Ian Campbell、Ian Gibson、Alain Bernard、Joachim Schulz、Patricia Graf、Bhrigu Ahuja等。增材制造设计：趋势、机遇、考量与约束。CIRP annals，65(2):737–760，2016年。[13] Jikai Liu、Andrew T Gaynor、Shikui Chen、Zhan Kang、Krishnan Suresh、Akihiro Takezawa、Lei Li、Junji Kato、Jinyuan Tang、Charlie CL Wang等。增材制造拓扑优化的当前与未来趋势。《结构与多学科优化》，57:2457–2483，2018年。[14] Fabian Wein、Peter D Dunning 和 Julián A Norato。结构优化中特征映射方法综述。《结构与多学科优化》，62:1597–1638，2020年。[15] Weisheng Zhang、Jishun Chen、Xuefeng Zhu、Jianhua Zhou、Dingchuan Xue、Xin Lei 和 Xu Guo。通过移动可变形空洞（mmv）方法实现显式三维拓扑优化。Computer Methods in Applied Mechanics and Engineering，322:590–614，2017年。[16] Xu Guo、Weisheng Zhang 和 Wenliang Zhong。显式且几何化地进行拓扑优化——一种基于移动可变形组件的新框架。Journal of Applied Mechanics，81(8)，2014年。[17] Shouyu Cai 和 Weihong Zhang。用于结构形状与拓扑优化的自适应气泡法。Computer Methods in Applied Mechanics and Engineering，360:112778，2020年。[18] JA Norato、BK Bell 和 Daniel A Tortorelli。用于基于连续体的拓扑优化（含离散单元）的几何投影方法。Computer Methods in Applied Mechanics and Engineering，293:306–327，2015年。[19] Boyang Deng、Kyle Genova、Soroosh Yazdani、Sofien Bouaziz、Geoffrey Hinton 和 Andrea Tagliasacchi。Cvxnet：可学习的凸分解。载于IEEE/CVF计算机视觉与模式识别会议论文集，第31–44页，2020年。[20] William E Lorensen 和 Harvey E Cline。行进立方体：一种高分辨率三维表面构造算法。ACM SIGGRAPH Computer Graphics，21(4):163–169，1987年。[21] Shanglong Zhang、Julián A Norato、Arun L Gain 和 Naesung Lyu。用于板结构拓扑优化的几何投影方法。《结构与多学科优化》，54:1173–1190，2016年。[22] Julián A Norato。使用超形状进行拓扑优化。《结构与多学科优化》，58(2):415–434，2018年。[23] Stephen Boyd、Stephen P Boyd 和 Lieven Vandenberghe。凸优化。剑桥大学出版社，2004年。[24] Krister Svanberg。移动渐近线法——一种新的结构优化方法。《国际工程数值方法杂志》，24(2):359–373，1987年。[25] Aaditya Chandrasekhar、Alfred K.C. Cheung 和 Ian A.D. Williamson。mmapy：移动渐近线法（MMA）的 Python 实现，2022年。[26] Krister Svanberg。MMA 和 GCMMA：两种非线性优化方法。第1卷：1–15，2007年。[27] Jorge Nocedal 和 Stephen J Wright。数值优化。Springer，1999年。[28] James Bradbury、Roy Frostig、Peter Hawkins、Matthew James Johnson、Chris Leary、Dougal Maclaurin、George Necula、Adam Paszke、Jake VanderPlas、Skye Wanderman-Milne 和 Qiao Zhang。JAX：可组合的 Python 变换+NumPy 程序，2018年。[29] Aaditya Chandrasekhar、Saketh Sridhara 和 Krishnan Suresh。Auto：用于拓扑优化中自动微分的框架。《结构与多学科优化》，64(6):4355–4365，2021年。[30] Momchil Minkov、Ian AD Williamson、Lucio C Andreani、Dario Gerace、Beicheng Lou、Alex Y Song、Tyler W Hughes 和 Shanhui Fan。通过自动微分对光子晶体进行逆向设计。ACS Photonics，7(7):1729–1741，2020年。[31] Julián A Norato。使用超形状进行拓扑优化。《结构与多学科优化》，58(2):415–434，2018年。

[32] Zhan Kang、Yaguang Wang 和 Yiqiang Wang。基于水平集方法、通过最小距离控制多相嵌入组件的结构拓扑优化。*Computer Methods in Applied Mechanics andEngineering*，306：299–318，2016年。[33] Erik Andreassen、Anders Clausen、Mattias Schevenels、Boyan S Lazarov 和 Ole Sigmund。使用88行代码在 MATLAB 中实现高效拓扑优化。《结构与多学科优化》，43：1–16，2011年。[34] Ole Sigmund。用 MATLAB 编写的 99行拓扑优化代码。《结构与多学科优化》，21：120–127，2001年。[35] Zhan Kang 和 Yiqiang Wang。基于材料密度与水平集组合描述的含嵌入式可移动孔洞的集成拓扑优化。*Computer methodsin applied mechanics and engineering*，255：1–13，2013年。[36] Xiaoping Qian。拓扑优化中的欠切与悬垂角控制：一种基于密度梯度的积分方法。《国际工程数值方法杂志》，111(3)：247–272，2017年。[37] Fengwen Wang、Boyan Stefanov Lazarov 和 Ole Sigmund。关于拓扑优化中的投影方法、收敛与鲁棒形式。《结构与多学科优化》，43：767–784，2011年。[38] 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。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第165–174页，2019年。