arXiv:2601.13603v1 [cs.CG] 20 Jan 2026

DCCVT：可微裁剪质心 Voronoi 镶嵌

Wylliam Cantin Charawi[1] Adrien Gruson[1] Jane Wu[2] Christian Desrosiers[1] Diego Thomas[3]
[1] ´高等技术学院 ´erie [2]加州大学伯克利分校 [3]九州大学

摘要

尽管移动立方体（*MC*）和 *Marching Tetrahedra*（*MTet*）因简单高效而被广泛用于三维重建流程，但其可微变体在网格提取方面仍不够理想。这往往限制了基于学习的框架从点云或图像重建三维网格的质量。相比之下，裁剪质心 *Voronoi* 镶嵌具有更强的理论保证，能够生成质量更高的网格。然而，由于缺少可微分的表述，它们一直无法融入现代机器学习流程。为弥补这一缺口，我们提出 *DCCVT*，这是一种可微算法，利用裁剪质心 *Voronoi* 镶嵌从含噪有符号距离场（*SDF*）中提取高质量三维网格。我们推导出计算裁剪质心 *Voronoi* 镶嵌的完全可微表述，并展示了如何将其与基于深度学习的 *SDF* 估计相结合，从输入点云重建精确的三维网格。基于合成数据的实验表明，在网格质量和重建保真度方面，*DCCVT* 的表现优于最先进的方法。
https://wylliamcantincharawi.dev/DCCVT.github.io/

1. 引言

高质量的三维内容对于众多领域都至关重要，包括增强现实/虚拟现实、数字仿真、医疗保健、游戏和电影制作。在各种三维表示中，三维网格等离散格式因其渲染和操作简单而得到广泛使用。然而，从真实世界数据（如二维图像或不完整、含噪的点云）生成精确的三维内容是一项极具挑战性的任务，通常需要专业知识和大量人工投入。因此，大量研究致力于辅助或完全自动化这一过程。一种常见的方法是使用隐式表示，例如符号距离场（SDFs），根据观测结果对其进行优化，随后通过后处理步骤将其转换为离散的三维网格。这种方法之所以有效，是因为平滑的符号距离场非常适合采用优化技术。

近年来，利用多视图图像对符号距离场进行优化和正则化受到了广泛关注
或三维点云。然而，提取

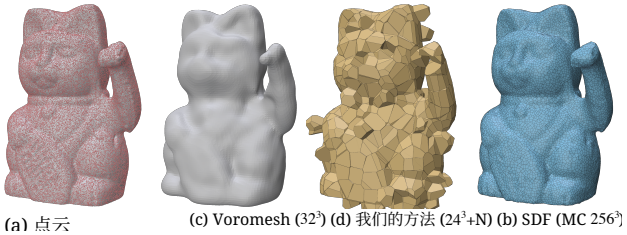


图 1. 无定向点云的三维网格重建对比。(a) 输入点云。(b) 使用移动立方体在 256^3 分辨率下重建得到的输入符号距离函数（SDF）。(c) 使用 32^3 个采样点的 Voromesh 基线。(d) 我们提出的 DC-CVT 方法，采用 24^3 个采样点及近邻采样点，实现了更准确、规则的曲面重建。

三维网格通常通过与优化过程相分离的简单方法进行提取，往往导致重建效果不够理想。为解决这一问题，研究者提出了可微分版本的移动立方体（DMC） [19] 和 Marching Tetrahedra（DMTet） [32] ，使网格提取能够直接融入优化循环。这些方法在以符号距离场或点云作为输入时，能够提升网格重建质量。尽管如此，我们认为，底层算法——移动立方体和 Marching Tetrahedra——本身并不适合高质量网格提取，因此其可微分版本（DMC 和 DMTet）仍存在根本性局限。

移动立方体在固定网格上运行，在网格顶点处采样符号距离函数值，并沿符号距离函数穿过零值的边提取三角网格。尽管这种方法快速高效，但由于网格单元的方向固定，通常会产生较差的镶嵌效果，导致三角形不规则并引入近似误差，尤其是在处理复杂几何形状时更为明显。因此，通常需要进行后处理，以生成更加规则的三维网格。另一方面，Marching Tetrahedra 通过采用德劳内四面体剖分提供了更大的灵活性，使镶嵌能够更好地适应隐式曲面的形状。然而，网格中的不规则性仍然难以完全消除。近期研究表明，质心沃罗诺伊镶嵌（CVT）能够生成规则性显著提高的体积和表面离散化结果 [8]。此外，与 MarchingCubes 和 Marching Tetrahedra 相比，CVT 还能为隐式形状提供经证明更为精确的近似

与 MarchingCubes 和 Marching Tetrahedra 相比，精确近似隐式形状。

在本文中，我们提出了 DCCVT，这是一种用于裁剪质心 Voronoi 镶嵌（CCVT）的可微算法 [38]。与 MC 和 MTet 相比，CCVT 已展现出显著优势；我们也证明，DCCVT 相较于 DMC 和 DMTet 同样具有显著优势。尽管我们的重点是根据输入点云进行三维网格重建，但我们提出的可微裁剪质心 Voronoi 镶嵌这一核心贡献，也广泛适用于其他三维网格重建流程。我们的贡献如下：

- 一种从稀疏点云中提取质心 Voronoi 镶嵌的方法；
- 通过裁剪操作，对 Voronoi 站点位置及其对应的符号距离函数值进行富有表现力的端到端优化；
- 一种迭代上采样方法，可提升我们方法的整体鲁棒性。

2. 相关工作

显式方法。

显式方法直接连接采样点以构建网格。一个典型例子是 Bernardini 等人提出的球旋转算法（BPA）[4]，：当一个用户指定半径的球能够同时接触三个点，且不包围其他点时，算法便生成一个三角形。这种简单的启发式方法能够生成水密网格，在实践中仍被广泛使用。然而，BPA 存在一些明显的局限性，包括对噪声和非均匀采样密度敏感，以及需要调节一个可能并不适用于所有表面区域的单一球半径。相比之下，泊松表面重建 [14, 15] 将网格恢复表述为利用局部基函数求解一个条件良好的稀疏线性系统。它不依赖半径启发式方法，通常对噪声更鲁棒，但需要带方向点云，而原始扫描数据中往往没有这类点云。法线估计算法 [1, 2, 11, 17]可用于估计点样本方向，但现有方法仍会产生带有噪声的预测结果。

此外，人们还提出了用于点集三角剖分的深度学习方法 [22, 25, 31, 35, 44, 45]。DeepDT [22]通过直接从点云及其对应的德劳内三角剖分中预测四面体的内外标签来重建曲面。PointTriNet [31] 提出候选三角形，另一个分类器则判断每个三角形是否应出现在最终三角剖分中。DMNet [45] 以及 Zhang 和 Tao [44] 的后续工作将德劳内三角剖分建模为对偶图，并将局部几何特征嵌入其结构中。

隐式方法。许多基于学习的曲面重建方法，都涉及直接根据非结构化点集[6, 7, 9, 10, 13, 18, 23, 27, 34, 39, 40, 46]预测三维曲面的连续符号距离函数。其中一种较早的相关工作是 DeepSDF [27], ，它引入了一种潜变量条件神经网络，可为整个形状类别学习连续的符号距离函数，从而能够根据部分/含噪数据实现高质量的形状表示、插值和补全。Points2Surf [9]提出了一种基于补丁的框架，通过将精细的局部补丁与粗略的全局有符号距离函数符号预测相结合，在不使用法线的情况下从原始点云学习隐式曲面；这使得重建更加准确，并提高了对未见形状的泛化能力。NeuralTPS [6]利用薄板样条曲面参数化，在训练过程中生成粗略的曲面采样点，无需学习到的形状先验或法线输入，即可根据单个稀疏点云推断符号距离函数。Zhang 等人 [46] 提出了一种两阶段方法，将符号距离函数学习与自适应 Delaunay 网格剖分算法相结合。近期的 Hotspot 方法 [40]可在几分钟内推断出神经符号距离函数。除了符号距离函数表示之外，研究者还提出了许多神经隐式表示，用于从点云进行曲面重建[3, 12, 20, 26, 29, 33, 36, 37, 41, 43]。

虽然隐式表示能够捕捉平滑且精确的曲面，但应用移动立方体或 Marching Tetrahedra 等标准网格提取方法时，细节保留往往会受到影响。因此，当前研究正致力于将网格提取整合到学习过程中，以保真地保留细节，而可微提取则逐渐成为一种关键方法。

可微网格提取。 Liao 等人 [19] 提出了 Deep Marching Cubes，这是一种可微的移动立方体变体，能够实现从点云进行端到端训练。然而，该方法要求使用轴对齐体素网格，因此受到细节分辨率的限制。为解决这一问题，DMTet [32]以四面体离散化取代网格，消除了轴对齐约束，并能够实现更精细的重建。类似地，Wu 等人 [42]提出了一种具有真实梯度的可微 Marching Tetrahedra 变体，用于稀疏视角三维人体重建。最近，Maruani 等人 [24] 提出了 Voromesh，利用沃罗诺伊图的规律性来优化空间镶嵌，从而提升网格质量。然而，大多数现有的可微方法将网格优化和有符号距离函数优化视为彼此独立的阶段，这可能限制最终重建的最优性。

3. 预备知识

一组三维采样点 $\{\mathbf{s}_i\}_{i=1}^N \subset \mathbb{R}^3$ 隐式地表示了三维空间的一种镶嵌，该镶嵌由沃罗诺伊单元 $\mathcal{V}(\mathbf{s}_i)$ 构成，其定义如下：

$$\mathcal{V}(\mathbf{s}_i) = \{\mathbf{p} \in \mathbb{R}^3 \mid \|\mathbf{p} - \mathbf{s}_i\| < \|\mathbf{p} - \mathbf{s}_j\|, \forall j \neq i\}.$$

沃罗诺伊图也是德劳内四面体剖分的对偶 $\mathcal{T} = \{\mathcal{T}_m\}_{m=1}^M$。

每个四面体 $\mathcal{T}_m \subset \mathcal{T}$ 及其对应的采样点$\{\mathbf{s}_a, \mathbf{s}_b, \mathbf{s}_c, \mathbf{s}_d\}$ 定义一个 Voronoi 顶点 $\mathbf{v}_m$，即其外接圆心，计算方式为

$$\mathbf{v}_m = \mathbf{s}_a + \frac{\alpha\,(\mathbf{q}\times\mathbf{r}) + \beta\,(\mathbf{r}\times\mathbf{p}) + \gamma\,(\mathbf{p}\times\mathbf{q})}{2\,\mathbf{p}\cdot(\mathbf{q}\times\mathbf{r})}, \quad (1)$$

其中，$\mathbf{p} = \mathbf{s}_b - \mathbf{s}_a$、$\mathbf{q} = \mathbf{s}_c - \mathbf{s}_a$、$\mathbf{r} = \mathbf{s}_d - \mathbf{s}_a$ 是局部边，α，β，γ是它们的平方范数。分母是标量三重积的 2 倍，即 $12\,V_{\text{signed}}$，其中四面体体积为 $V = |\mathbf{p}\cdot(\mathbf{q}\times\mathbf{r})|/6$。

沃罗诺伊单元质心。 通常，使用随机采样点初始化的沃罗诺伊图会产生不规则的离散化结果，其中采样点通常形成细长四面体和薄片[1],，这可能在三维网格提取过程中引发严重问题。为了提高镶嵌的规则性，最小化质心沃罗诺伊镶嵌（CVT）能量是一种广泛使用且鲁棒的策略。当所有采样点都与其对应沃罗诺伊单元的质心重合时，就称沃罗诺伊图处于 CVT 配置。这种配置可以通过 Lloyd 算法来实现 [21]。

4. 所提方法

我们旨在从无定向点云中重建精确且水密的三维网格（图 2a ）。我们将表面建模为在三维网格上离散化的隐式符号距离函数（图2b），并从中提取网格。不同于以往仅优化符号距离函数或采样站点位置的方法，我们联合优化站点位置$\{\mathbf{s}_i\}_{i=1}^N$ 及其对应的符号距离函数值 $\{\phi_i\}_{i=1}^N$。我们的方法包括：（1）通过鲁棒投影方案将零交叉沃罗诺伊顶点 $\{\mathbf{v}_j\}_{j=1}^M$ 投影到符号距离函数零水平集上（第 4.1 节）；（2）对感知符号距离函数的沃罗诺伊图进行正则化，使其满足质心沃罗诺伊镶嵌（CVT）性质（第 4.2节）；（3）通过误差驱动的采样点插入，自适应地细化离散化（第 4.3节）。该优化过程完全可微，便于与现代机器学习框架集成。图 2 概述了我们提出的方法。为简洁起见，所有示意图均以二维形式呈现。

4.1. 联合优化

我们方法的核心思想是将传统的沃罗诺伊图转化为感知表面的沃罗诺伊图。在这种图中，由与 SDF 零水平集相交的四面体生成的沃罗诺伊顶点，会被定位在通过裁剪沃罗诺伊单元提取的最终三维网格顶点附近。我们以可微的方式同时构建 CVT 正则化和向 SDF 零水平集的顶点投影，从而在端到端框架中联合优化符号距离函数值和离散化，使得损失函数可以直接应用于输出三维网格的顶点。

在本节余下部分中，我们仅关注与定义为 $\mathcal{T}^{(0)}$ 的符号距离函数的零水平集相交的四面体。它们是决定输出三维网格顶点的元素。四面体 $\mathcal{T}_j$ 由四个站点 $\{\mathbf{s}_a, \mathbf{s}_b, \mathbf{s}_c, \mathbf{s}_d\}$ 定义；若其顶点处的符号距离函数值满足以下条件，则将其归类为相交四面体

$$\min(\phi_a,\phi_b,\phi_c,\phi_d) < 0 < \max(\phi_a,\phi_b,\phi_c,\phi_d)\,. \quad (2)$$

对于每个这样的四面体，我们使用公式（$\mathbf{v}j$ 1）计算对应的 Voronoi 顶点，然后将其投影到符号距离函数的零水平集上。同样，确定所有穿越边的中点，并将其投影到零水平集上。随后，将这些投影点与输入目标点之间的距离作为数据损失进行计算。通过将投影过程表示为关于符号距离函数值和采样点位置均可微的形式，我们便能在统一框架下进行联合优化。

零交叉 Voronoi 顶点的投影。

图 2 （c）展示了将 Voronoi 顶点投影到符号距离函数的零水平集上的过程。每个 Voronoi 顶点都定义为其对应四面体的外心。然而，该外心有时可能远离四面体本身，从而使符号距离函数值和梯度的直接重心插值变得不可靠。此外，显式搜索包含给定 Voronoi 顶点的四面体在计算上代价极高。为了解决这些问题，我们引入了一种高效且鲁棒的投影策略，即使在具有挑战性的四面体构型下也能保持有效。具体而言，对于每个四面体，我们根据其四个顶点在零水平集上的投影拟合一个平面，并将相应的 Voronoi 顶点投影到该平面上。

对于每个 $\mathcal{T}_i$ 穿过零水平集的四面体，我们首先将其关联的采样点$\mathbf{s}_i'$ 投影到零水平集上，得到

$$\mathbf{s}_i' = \mathbf{s}_i - \frac{\nabla\phi_i}{\|\nabla\phi_i\| + \epsilon}\,\phi_i\,. \quad (3)$$

然后，我们计算投影点的质心，得到

$$\bar{\mathbf{s}}_j' = \frac{1}{4}\sum_{i\in\mathcal{T}_j}\mathbf{s}_i', \quad (4)$$

我们寻找一个平面，其法向量为 $\mathbf{n}j$，该平面经过站点质心 $\bar{}\mathbf{s}_j'$，且到投影后采样点$\mathbf{s}_i'$ 的平均距离最小。将中心化坐标表示为 $\Delta\mathbf{s}_i' = \mathbf{s}_i' - \bar{}\mathbf{s}_j'$，可以通过优化以下问题求得该平面：

$$\arg\min_{\mathbf{n}_j}\ \frac{1}{4}\sum_{i\in\mathcal{T}_j}\Big(\underbrace{\frac{\mathbf{n}_j}{\|\mathbf{n}_j\|}\cdot\Delta\mathbf{s}_i'}_{\text{distance to plane}}\Big)^2. \quad (5)$$

[1]扁平且细长的 Voronoi 单元

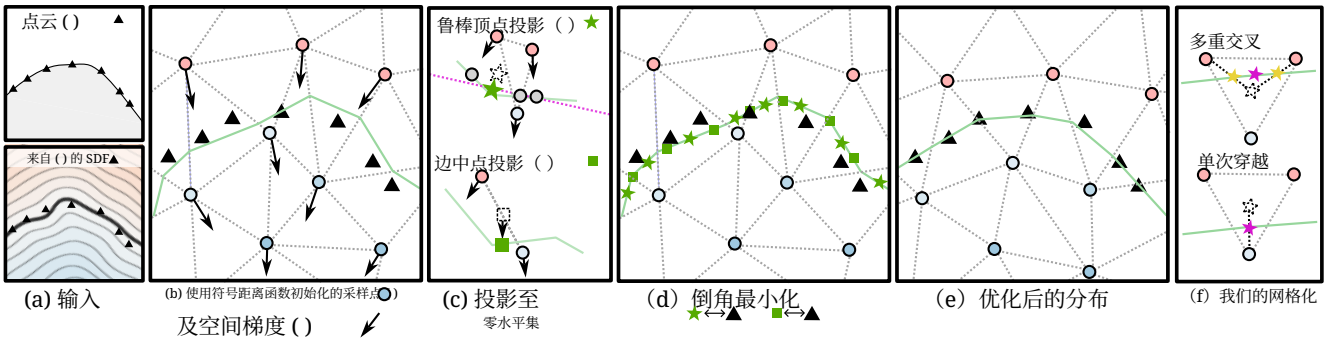


图2. 我们提出的方法概述。从无方向的三维点云以及根据点云推断出的 SDF 估计值（a）出发，我们初始化 Voronoi 站点（b），由此得到离散化的 SDF（绿色表示零水平集）。随后，我们将零交叉 Voronoi 顶点（绿色星形）和边中点（绿色方形）投影到 SDF 零水平集上，并最小化其与输入点云之间的倒角距离（c–d）。这一优化会生成改进后的站点分布和 SDF 表示（e），随后我们使用基于 Voronoi 的网格生成策略提取最终网格（f），其中 Voronoi 顶点以紫色显示。

该法向量 $\mathbf{n}_j$定义了经过质心 $\mathbf{c}_j$的拟合平面的方向，对应于[2] 中心化站点协方差矩阵的第一特征向量。最后，我们将 Voronoi 顶点 $\mathbf{v}_j$ 投影到该拟合平面上，得到

$$\mathbf{v}'_j \;=\; \mathbf{v}_j - \left[(\mathbf{v}_j - \mathbf{c}_j)\cdot \mathbf{n}_j\right]\,\mathbf{n}_j\,. \tag{6}$$

该投影仅利用基于站点的信息，确保 Voronoi 顶点位于 SDF 的零水平集上。完整推导请参见补充材料（第 7 节），其中还包含与朴素牛顿投影的对比实验。

零交叉边中点的投影。 除 Voronoi 顶点外，我们还计算穿过零水平集的四面体边的中点（图2c）。这些边中点提供了额外的约束，确保 Voronoi 面与我们的 SDF 表示和目标点云均保持良好对齐。对于每条对应于一对相邻站点 $(\mathbf{s}_i,\mathbf{s}_j)$ 且穿过零水平集的边，我们近似计算其中点 $\mathbf{b}_{ij}$ 的 SDF 值和梯度向量，如下所示。

$$\phi(\mathbf{b}_{ij}) \approx \tfrac{1}{2}(\phi_i+\phi_j),\;\; \nabla\phi(\mathbf{b}_{ij}) \approx \tfrac{1}{2}(\nabla\phi_i+\nabla\phi_j) \tag{7}$$

最后，我们使用牛顿步法（式（**8**））将边的中点 $\mathbf{b}_{ij}$ 投影到 SDF 的零水平集上。

$$\mathbf{b}'_{ij} \;=\; \mathbf{b}_{ij} - \frac{\nabla\phi(\mathbf{b}_{ij})}{\|\nabla\phi(\mathbf{b}_{ij})\|+\epsilon}\,\phi(\mathbf{b}_{ij})\,, \tag{8}$$

空间 SDF 梯度。 为了计算每个站点处的 SDF 空间梯度，我们假设每个四面体内部的梯度为常数，记为 $\nabla\phi^{(j)}$。使用一阶近似，站点 $i\in\mathcal{T}_j$ 处的 SDF 估计为

$$\hat{\phi}_i \;\approx\; \bar{\phi}_j \;+\; (\mathbf{s}_i-\bar{\mathbf{s}}_j)^{\top}\nabla\phi^{(j)}. \tag{9}$$

[2]对应最小特征值的特征向量。

然后，通过最小化所有采样点的实际符号距离函数值与估计符号距离函数值之间的均方差，得到梯度 $\nabla\phi^{(j)}$：

$$\arg\min_{\nabla\phi^{(j)}} \;\frac{1}{4}\sum_{i\in\mathcal{T}_j}\left(\hat{\phi}_i-\phi_i\right)^2. \tag{10}$$

如补充材料第8 节所述，可以通过求解一个线性系统高效地完成此计算。

然后，站点 $\mathbf{s}_i$ 处的空间梯度可通过其关联四面体的梯度的体积加权平均得到：

$$\nabla\phi_i \;=\; \frac{1}{\sum_{j\in\mathcal{D}_\mathbf{s}} V_j}\sum_{j\in\mathcal{D}_\mathbf{s}} V_j\nabla\phi^{(j)}, \tag{11}$$

其中，$\mathcal{D}_\mathbf{s}$ 是与 $\mathbf{s}$ 相邻的四面体集合，V_j 是四面体 j 的体积，而 $\nabla\phi^{(j)}$ 是其在式（33）中定义的局部梯度。

4.2. 正则化

引导我们进行优化的数据损失是原始点云与重建顶点及边中点集合之间的 Chamfer 距离，定义如下：

$$\mathcal{L}_{\text{CD}} \;=\; \min_{\text{dist}}\left(\{\|\mathbf{p}-\mathbf{v}_i\|^2, \|\mathbf{p}-\mathbf{b}_{ij}\|^2\}\right) \tag{12}$$

其中$\mathbf{v}_i$ 为沃罗诺伊顶点，$\mathbf{b}_{ij}$ 为边中点。该损失鼓励沃罗诺伊顶点和边都靠近输入点云，从而确保提取的网格能够准确表示输入数据。除了这一主要损失外，我们还引入了若干正则化损失，以促进采样点均匀分布并使符号距离函数平滑变化。

CVT正则化。 使用顶点计算一种感知表面的质心沃罗诺伊镶嵌（CVT）损失投影 到零水平集上（公式（6））。不与零水平集相交的四面体中的顶点保持不变。在优化过程中，站点可能偏离

该质心，从而导致沃罗诺伊单元不规则。为鼓励规则性，我们定义一种 CVT 损失，用于惩罚每个站点$\mathbf{s}_i$与其质心$\mathbf{c}_i$之间的欧氏距离，表示为

$$\mathcal{L}_{\mathrm{CVT}} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{s}_i - \mathbf{c}_i\| \,. \tag{13}$$

Eikonal正则化。 符号距离场的一项基本性质是满足Eikonal方程，即要求符号距离函数的梯度在每个点都具有单位范数。这一性质对于确保距离场的一致性至关重要，并广泛应用于我们的多次投影步骤中。为了对 SDF 梯度进行正则化，我们采用 Wu 等人提出的定义 [42] 并将我们的 Eikonal损失定义为

$$\mathcal{L}_{\mathrm{Eik}} = \frac{1}{4M}\sum_{j=1}^{M}\sum_{i=1}^{4} V_j\left(\left\|\nabla\phi_{j,i}\right\|^2 - 1\right)^2, \tag{14}$$

其中， M 是四面体的数量，$\nabla\phi_{j,\ i}$ 是四面体 j顶点 i 处的 SDF 梯度。这会在整个域内鼓励 $\|\nabla\phi\| \approx 1$，以体积加权离散形式保持 SDF 的性质。

平均曲率运动（MbMC）正则化。 如 Wu 等人所示，[42], 可以通过引入平均曲率运动（MbMC）项对 SDF 进行正则化，该项鼓励 SDF 按照其平均曲率演化。具体做法是在损失函数中添加一项，以惩罚偏离平均曲率流的情况。为此，他们定义了一个平滑Heaviside函数H ，如下所示:

$$H(\hat{\phi}) = \begin{cases} 0, & \hat{\phi} < -\epsilon_H \\ \frac{1}{2} + \frac{\hat{\phi}}{2\epsilon_H} + \frac{1}{2\pi}\sin\left(\frac{\pi\hat{\phi}}{\epsilon_H}\right), & -\epsilon_H \le \hat{\phi} \le \epsilon_H \\ 1, & \hat{\phi} > \epsilon_H \end{cases} \tag{15}$$

其中 ϵ_H 等于采样点之间的平均边距离，不包括最长的 5% 边。通过在求解四面体梯度时重用式（$\mathbf{W}_j$ ）中定义的权重矩阵33，我们可以计算每个四面体处的平均曲率:

$$\nabla\mathcal{H}^{(j)} = \mathbf{W}_j \begin{bmatrix} H(\phi_A) - \bar{H}_j \\ H(\phi_B) - \bar{H}_j \\ H(\phi_C) - \bar{H}_j \\ H(\phi_D) - \bar{H}_j \end{bmatrix}, \tag{16}$$

¯其中 H_j 表示四面体 $\mathcal{T}_j$的四个采样点处 Heaviside 函数值的平均值，类似于公式（31）中定义的平均 SDF 值。MbMC损失定义为

$$\mathcal{L}_{\mathrm{H}} = \frac{1}{M}\sum_{j=1}^{M} V_j \left\|\nabla\mathcal{H}^{(j)}\right\|, \tag{17}$$

其中， M 表示四面体的数量， V_j 表示四面体 $\mathcal{T}_j$ 的体积。

损失。我们的优化的总损失为

$$L = \mathcal{L}_{CD} + \lambda_{\mathrm{CVT}}\mathcal{L}_{CVT} + \lambda_{\mathrm{Eik}}\mathcal{L}_{Eik} + \lambda_{\mathrm{H}}\mathcal{L}_{H}\,, \tag{18}$$

其中 $\lambda_{\mathrm{CVT}} = 0.1$、 $\lambda_{\mathrm{Eik}} = 0.02$和 $\lambda_{\mathrm{H}} = 0.1$ 分别是相应损失的权重。

4.3. 上采样

我们方法的一个关键特点是能够用数量有限的沃罗诺伊站点表示复杂形状。不过，与以往方法类似，[24, 32], 当有更多活跃站点可用时（即有四面体穿过符号距离函数的零水平集的站点），所提出的方法能够获得更高的网格质量。为此，我们采用一种自适应上采样策略，根据局部符号距离函数特征插入新站点，以细化重建域。是否插入额外站点由基于局部间距和曲率的评分函数决定。

局部特征计算。我们为每个 Voronoi 站点$\mathbf{s}_i$ 计算局部特征，以引导上采样过程。第一个特征是局部间距 ρ_i，它衡量到最近邻站点的距离，其定义如下:

$$\rho_i = \min_{j\in\mathcal{N}(i)}\|\mathbf{s}_i - \mathbf{s}_j\|\,, \tag{19}$$

其中 $\mathcal{N}(i)$ 是站点 i 的一环邻域。该特征可以识别实际密度较低的区域。

第二个特征是曲率代理 κ_i，它衡量站点 i 邻域内 SDF 梯度的平均变化。计算方式如下:

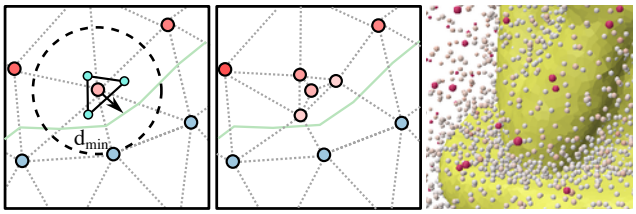
$$\kappa_i = \frac{\alpha_\kappa}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\left\|\hat{\nabla}\phi_i - \hat{\nabla}\phi_j\right\|^2 + (1-\alpha_\kappa), \tag{20}$$

ˆ其中 $\nabla\phi = \nabla\phi/(\|\nabla\phi\| + \epsilon)$ 是单位 SDF 梯度，而 $\alpha_\kappa = 0.8$ 用于使平坦区域也能获得非零分数。该特征捕捉 SDF 的局部曲率，这对于识别需要更多站点的区域至关重要。

分数计算与候选选择。我们根据局部间距和曲率特征，为每个 Voronoi 站点$\mathbf{s}_i$ 计算分数，具体如下:

$$\mathcal{S}_i = \left(\frac{\rho_i}{\tilde{\rho}}\right)\left(\frac{\kappa_i}{\tilde{\kappa}}\right), \quad i \in \mathcal{T}^{(0)}, \tag{21}$$

其中， $\mathcal{T}^{(0)}$ 表示零交叉四面体的集合， $\tilde{\rho}$和 $\tilde{\kappa}$ 分别表示所有活跃站点的局部间距和曲率中位数。该评分旨在平衡站点间距的均匀性与曲率。其他生成的单元未穿过零水平集的站点，其评分为零。



(a) 四面体插入 (b) 新的站点配置 (c) 可视化站点分布

图 3. 选定一个站点后，我们计算最小距离 d_{min}，并插入一个与 SDF 梯度对齐的四面体 (a)。这保留了局部站点连通性 (b)。我们的上采样策略产生了非均匀站点分布 (c)。

我们根据站点候选项的评分，采用与评分成比例的重要性采样来选择站点候选项。每个候选站点在每次上采样迭代中最多只能被选中一次。

调度。 我们以固定间隔执行上采样，直到优化轮次达到 80%；每一步相对于当前采样点数量增加 10% 的采样点。在实验中，我们最多执行 10 次上采样。为确保与非渐进式上采样方法进行公平比较，我们对采样点总数设定上限。

四面体插入。插入新站点的一种方法是：以候选站点为中心，沿 SDF 的空间梯度方向，在半球上随机采样一个点，并依据其 SDF 距离进行采样。然而，添加单个点可能会扰乱沃罗诺伊站点之间的局部连通性。因此，我们提出在每个选定的候选站点周围插入一个小型四面体结构。此方法使我们能够维持先前沃罗诺伊镶嵌的局部连通性和规则性。如图3所示，对于每个选定的站点 $\mathbf{s}_i$，我们生成 4 个新站点，使其构成一个与空间 SDF 梯度方向对齐的正四面体。令 $\{\mathbf{d}_k\}_{k=1}^4$ 为规范四面体方向，且 $\mathbf{T}_i \in \mathbb{R}^{3\times3}$ 为 $\mathbf{s}_i$处的局部坐标系，则：

$$\mathbf{s}'_{i,k} = \mathbf{s}_i + \frac{\rho_i}{4}\mathbf{T}_i\mathbf{d}_k, \quad k = 1, \dots, 4, \tag{22}$$

其中， ρ_i 是式（19）中定义的局部间距，$\mathbf{s}'_{i,\ k}$ 是待插入的新站点。我们将正四面体缩放至局部间距的 1/4，以避免添加的新站点跨越零水平集。对于每个新站点，我们使用原始站点$\mathbf{s}_i$ 处的 SDF 值和空间梯度 $\nabla\phi_i$，按如下方式计算其 SDF：

$$\phi(\mathbf{s}'_{i,k}) \approx \phi(\mathbf{s}_i) + \nabla\phi_i^\top(\mathbf{s}'_{i,k} - \mathbf{s}_i). \tag{23}$$

4.4. 网格提取

每个与零水平集相交的四面体中的顶点，其最终顶点位置都可以根据优化后的站点位置和符号距离函数值，通过公式（6）或公式（8）计算得出。然而，由于我们的符号距离函数模型依赖线性近似，沃罗诺伊顶点无法保证恰好位于零水平集上。为确保网格一致且水密，我们采用了一种受 Marching Tetrahedra 启发的简化插值过程 [32]。

在该方法中，我们首先识别穿过零水平集的活跃四面体。对于每个活跃四面体，我们通过重心插值计算其对应的（未投影）顶点及其符号距离函数值。随后，我们识别由活跃顶点及其相关站点形成、且穿过零水平集的边。最后，在符号距离函数的零水平集处沿该边对顶点位置进行插值。如果一个顶点由多条相交边产生，我们将所有插值得到的位置取平均，以确定其最终位置（图 2f）。

5. 结果

我们使用公开的 Thingi32 数据集（Thingi10K 数据集的子集）评估该方法从无定向点云重建详细三维网格的有效性 [47] ，该数据集包含大量高分辨率真实网格。

5.1. 实验设置

在所有实验中，我们从每个真实三维网格中均匀采样 9.6k 个点，以生成输入点云。沃罗诺伊站点被初始化在不同分辨率的网格上，并在其位置上添加 0.005 的小幅随机扰动，以避免共面性。随后，将输入点云和初始站点位置归一化，使其位于立方体 $[-1, 1]^3$ 内。

SDF 初始化。 对于符号距离函数表示，我们采用最先进的神经网络 Hotspot [40],，其中每个模型都针对输入点云进行过拟合。我们使用训练模型的两个版本：(1) 训练 10k 次迭代后已收敛的模型，以及 (2) 训练 500 次迭代后未收敛的模型。前者用于初始化精确的符号距离函数表示，后者则用于评估我们的方法在符号距离函数表示不精确时的性能。请注意，我们在图 6 中单独研究形状 ID 398259，因为 HotStop 未能为该形状生成可靠的符号距离函数表示。

基线方法。 我们将我们的方法与 Voromesh [24]和 Marching Tetrahedra (MTet) [32] 进行比较，将它们作为基线方法。MTet 仅由根据提取的顶点计算的倒角距离驱动。Voromesh 最小化一种基于平面的距离函数，并且只优化采样点的位置。因此，图 1 (c) 中可以看到小型凸起。相比之下，MTet 和我们的方法都会联合优化采样点位置及其对应的符号距离函数值。所有方法均使用 PyTorch 重新实现 [28]，并采用相同的采样点进行初始化。所有方法均使用 Adam 优化器 [16]，学习率为 5×10^{-4}，且 β 系数设为 (0。8，0。99)。所有实验均固定执行 1k 次迭代。

与 Voromesh 不同，MTet 和我们的方法都需要

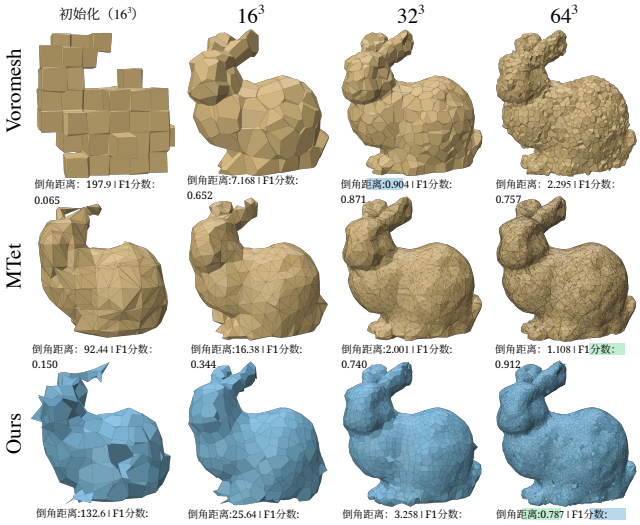


图 4. 具有精确符号距离函数表示的不同采样点分辨率下的优化结果。

对 Voronoi 站点进行四面体表示，该表示通过德劳内三角剖分获得。我们使用 gDel3D [5] 在图形处理器上加速这一计算。每次迭代都会从头重新计算三角剖分。随着 Voronoi 站点不断演变，探索对三角剖分进行增量更新仍是未来值得研究的方向。

质心近似。 计算精确质心非常耗时。为加快这一过程，我们采用近似的质心计算方法。我们不计算每个沃罗诺伊单元的精确质心，而是直接对与该单元关联的投影顶点和未投影顶点的位置取平均值。在实践中，我们没有观察到最终结果存在任何明显差异。

评估指标。 我们使用三个指标评估网格质量：平方倒角距离（CD）、F1 分数和法线一致性（NC）。为计算这些指标，我们采样 1M 个点，并使用 FCPW [30] 执行面到点投影。倒角距离衡量重建网格与输入点云的对齐程度，而 F1 分数评估网格捕捉原始形状的能力。我们将 F1 阈值设为 0.003，以确保对细节的敏感性。为便于阅读，所有报告的倒角距离均缩放 10^5倍。在表格和图中，我们使用绿色突出显示最佳指标，使用蓝色表示次优指标。

5.2. 分析

已收敛的符号距离函数示例。图4 展示了在不同初始站点网格分辨率下，且未进行上采样时所有方法获得的结果。除 Voromesh 外，所有方法在不同初始站点分布网格分辨率下均保持稳定。在较高分辨率下，Voromesh 为每个面提供的投影目标点过少，

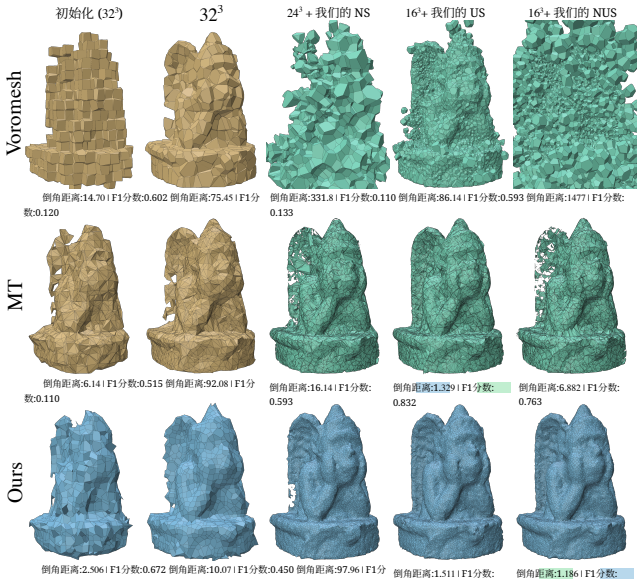


图 5. 不准确的 SDF 表示下不同上采样方法的结果。每次优化使用相同数量的站点。

这会导致面朝向不稳定。尽管我们的 64^3配置引入了一些表面噪声，但得益于 CVT 正则化，提取出的网格整体上仍保持均匀。

未收敛的 SDF 示例。图5 展示了一个更具挑战性的场景，其中 SDF 表示不准确。在这种情况下，由于 SDF 误差，初始的未优化表示存在大面积缺失区域。如图5的第二列所示，我们的方法和 MTet [32] 都能够在该 32^3 分辨率下从不准确的 SDF 中恢复。相比之下，Voromesh 无法恢复，因为它在最终提取时需要准确的占据信息。

非均匀站点分布与上采样。图5的其他列展示了三种点位布置策略的结果：(a) 一种称为近邻采样的朴素方法，将一定比例的站点布置在均匀下采样的输入点云附近；(b) 我们的四面体插入上采样策略（第4.3节）；以及 (c) 这两种策略的组合。在组合策略中，我们从一个均匀网格开始， 16^3, 添加 16^3 个近邻采样点，然后应用我们的迭代上采样策略。在所有情况下，这些替代布置策略最终都得到 32^3个站点。

近邻采样策略仅在我们的方法中有效，因为我们可以在优化过程中高效地移动站点。相比之下，MTet 会受到这种非均匀站点分布的影响，而 Voromesh 则因目标点投影不稳定而无法处理这种情况。我们的上采样策略通过自适应地放置采样点，提升了性能

方法	已收敛			未收敛		
	CD ↓	F1 ↑	NC ↑	CD ↓	F1 ↑	NC ↑
Voro 32^3	10.96	0.628	0.922	14.70	0.602	0.902
MTet 32^3	6.032	0.516	0.898	6.144	0.515	0.894
MTet U	2.539	0.682	0.916	2.443	0.682	0.917
我们的方法 U	2.521	0.677	0.928	2.601	0.672	0.925
MTet N	7.400	0.636	0.873	16.14	0.593	0.838
我们的方法 N	2.346	0.674	0.935	2.573	0.673	0.932
MTet 近邻采样与上采样	5.051	0.660	0.893	10.02	0.619	0.864
我们的方法 近采样上采样	2.326		0.930	2.276	0.680	0.930

表 1. Thingi32 数据集中的 31 个网格在 MTet 和 VoroMesh 32^3 基线以及我们的方法上的平均得分。为完整起见，我们列出了 MTet 的近邻采样（N）、上采样（U）及其组合（NU）方法，以及我们的方法。

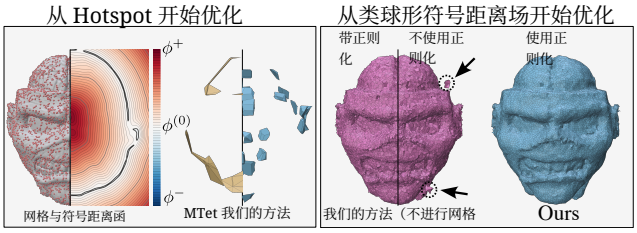


图 6. SDF 的内部/外部问题会导致 MTet 和我们的方法在网格提取时出现问题。从类球形符号距离场开始进行优化则效果良好。

于重建误差代理值最高的区域（公式（21））。总体而言，将近邻采样与我们的上采样策略结合，可获得最佳重建结果。

Thingi32 上的平均性能。表1 报告了所有方法在 Thingi32 数据集上的平均性能；由于网格 ID 398259 的 SDF 表示不准确且带有 HotSpot，因此将其排除。我们的方法始终优于 Voromesh 和采用固定分辨率的 MTet，展现了点位布置策略和优化过程的优势。如图 5所示，我们的上采样策略也能提升 MTet 的性能；得益于 CVT 公式及其与投影顶点的联系，我们的方法对所有自适应点位布置策略均保持鲁棒。总体而言，近邻采样与上采样相结合可取得最佳综合效果。在效率方面，当前实现使用 5个采样点时，优化需要约 32^3 分钟，运行于 RTX 3090 GPU 上——比 MTet（9 分钟）更快，但比 Voromesh（1 分钟）更慢。采用近似方法或更高效的优化策略，有望大幅降低这一计算开销。

简单的 SDF 初始化。图6显示，网格 ID 398259 的 HotSpot SDF 表示未能正确捕捉形状的内外部关系，这是由于采样点过于稀疏所致。这种糟糕的初始化会产生极薄的网格体积，导

方法	CD ↓	F1 ↑	NC ↑
我们的方法 32^3	9.496	0.459	0.878
不使用 CVT 正则化	11.68	0.433	0.835
不使用网格化	13.23	0.257	0.860
不使用中点	11.63	0.441	0.851

表 2. 使用 32^3 网格分辨率对 Thingi32 的 31 个形状进行的方法消融实验。

致所有仅依赖零交叉信息的方法都失效。相比之下，由于我们的方法也会优化 SDF 表示，因此可以从更简单的解析 SDF（例如球体）进行初始化，仍能恢复目标网格。不过我们注意到，在这个具有挑战性的案例中，如果不对 SDF 进行正则化，可能会出现漂浮伪影，如紫色网格所示。

5.3. 消融实验

表2 展示了在使用 32^3 站点分辨率和已收敛的 SDF 初始化时，针对不同设计选择进行的消融实验。我们观察到，CVT 正则化（第 4.2 节）对于获得高质量的重建至关重要，这一点也与先前结果一致。与直接使用公式（4.4）中的投影点相比，我们的网格化策略（第 3 式）能带来更好的性能。此外，投影中点（公式（7））对于改善面方向规整性也很重要。

6. 结论与未来工作

我们提出了 DCCVT，这是一种用于从点云重建三维网格的可微裁剪质心 Voronoi 镶嵌。通过将顶点投影到零水平集，同时保持 CVT 配置，我们的方法能够对优化后的形状进行规整离散化。此外，我们的框架支持不同的点位布置策略，从而实现更准确的形状表示。最后，我们的方法能够从各种 SDF 表示中重建网格，包括未收敛或解析初始化的表示。

一个很有前景的未来研究方向是将 DC-CVT 与深度学习框架相结合，以提升处理存在缺失区域的形状时的鲁棒性。具体而言，集成视觉基础模型可以提供强大的形状先验，使系统能够合理地填补输入点云中的空缺并修复拓扑孔洞。

致谢 本工作获得了九州大学日本学术振兴会/科研费 JP23H03439 和日本医疗研究开发机构 JP24wm0625404 的支持，以及 NSERC发现基金 RGPIN-2022-03182 的支持。J. W. 获得了美国国家科学基金会和加州大学校长博士后奖学金的支持。

参考文献

[1] Yizhak Ben-Shabat 和 Stephen Gould。《Deepfit：通过神经网络加权最小二乘进行 3D 曲面拟合》。载于《欧洲计算机视觉会议》，第 20–34 页。Springer，2020 年。2[2] Yizhak Ben-Shabat、Michael Lindenbaum 和 Anath Fischer。《Nesti-net：使用卷积神经网络对非结构化 3D 点云进行法线估计》。载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，第 10112–10120 页，2019 年。2[3] Yizhak Ben-Shabat、Chamin Hewa Koneputugodage 和 Stephen Gould。《Digs：用于无定向点云的散度引导形状隐式神经表示》。载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，第 19323–19332 页，2022 年。2[4] Fausto Bernardini、Joshua Mittleman、Holly Rushmeier、Cl´audio Silva 和 Gabriel Taubin。《用于曲面重建的球旋转算法》。《IEEE 可视化与计算机图形学汇刊》，5(4)：349–359，2002 年。2[5] Thanh-Tung Cao、Ashwin Nanjappa、Mingcen Gao 和 Tiow-Seng Tan。《一种 GPU 加速的 3D Delaunay 三角剖分算法》。载于《ACM 交互式 3D 图形与游戏研讨会》，2014 年。7，2[6] Chao Chen、Yu-Shen Liu 和 Zhizhong Han。《NeuralTPS：无需先验知识、从单个稀疏点云中学习有符号距离函数》。《IEEE 模式分析与机器智能汇刊》，2024 年。2[7] Gene Chou、Ilya Chugunov 和 Felix Heide。《GenSDF：可泛化有符号距离函数的两阶段学习》。《神经信息处理系统进展》，35：24905–24919，2022 年。2[8] Qiang Du 和 Desheng Wang。《基于质心沃罗诺伊镶嵌的四面体网格生成与优化》。《International Journal for Numerical Methods in Engineering》，56：1355–1373，2003 年。1[9] Philipp Erler、Paul Guerrero、Stefan Ohrhallinger、Niloy J Mitra 和 Michael Wimmer。《Points2Surf：从点云学习隐式曲面》。载于《欧洲计算机视觉会议》，第 108–124 页。Springer，2020 年。2[10] Amos Gropp、Lior Yariv、Niv Haim、Matan Atzmon 和 Yaron Lipman。《用于学习形状的隐式几何正则化》。arXiv 预印本 arXiv:2002.10099，2020 年。2[11] Paul Guerrero、Yanir Kleiman、Maks Ovsjanikov 和 Niloy J Mitra。《PCPNet：从原始点云学习局部形状属性》。载于《计算机图形论坛》，第 75–85 页。Wiley 在线图书馆，2018 年。2[12] Chiyu Jiang、Avneesh Sud、Ameesh Makadia、Jingwei Huang、Matthias Nießner、Thomas Funkhouser 等。《用于 3D 场景的局部隐式网格表示》。载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，第 6001–6010 页，2020 年。2[13] Yue Jiang、Dantong Ji、Zhizhong Han 和 Matthias Zwicker。《Sdfdiff：用于 3D 形状优化的有符号距离场可微渲染》。载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，第 1251–1261 页，2020 年。2

[14] Michael Kazhdan 和 Hugues Hoppe。屏蔽泊松曲面重建。ACM 图形学汇刊（ToG），32(3):1–13，2013。2[15] Michael Kazhdan、Matthew Bolitho 和 Hugues Hoppe。泊松表面重建。载于第四届欧洲计算机图形学协会几何处理研讨会论文集，2006。2[16] Diederik P. Kingma 和 Jimmy Ba。Adam：一种随机优化方法。载于国际学习表征会议（ICLR），2015。6[17] Jan Eric Lenssen、Christian Osendorfer 和 Jonathan Masci。深度迭代曲面法线估计。载于 IEEE/CVF 计算机视觉与模式识别会议论文集，第 11247–11256 页，2020。2[18] Shengtao Li、Ge Gao、Yudong Liu、Ming Gu 和 Yu-Shen Liu。利用隐式滤波从三维点云中学习神经符号距离函数。载于欧洲计算机视觉会议，第 234–251 页。Springer，2024。2[19] Yiyi Liao、Simon Donne 和 Andreas Geiger。深度行进立方体：学习显式曲面表示。载于 CVPR，2018。1，2[20] Yaron Lipman。相变、距离函数与隐式神经表示。arXiv 预印本 arXiv:2106.07689，2021。2[21] Stuart Lloyd。脉冲编码调制中的最小二乘量化。IEEE 信息论汇刊，28(2):129–137，1982。3[22] Yiming Luo、Zhenxing Mi 和 Wenbing Tao。DeepDT：从德劳内三角剖分中学习几何以进行曲面重建。载于 AAAI 人工智能会议论文集，第 2277–2285 页，2021。2[23] Baorui Ma、Zhizhong Han、Yu-Shen Liu 和 Matthias Zwicker。Neural-Pull：通过学习将空间拉向曲面，从点云中学习有符号距离函数。arXiv 预印本 arXiv:2011.13495，2020。2[24] Nissim Maruani、Roman Klokov、Maks Ovsjanikov、Pierre Alliez 和 Mathieu Desbrun。Voromesh：利用沃罗诺伊图学习水密曲面网格。载于国际计算机视觉大会，2023。2，5，6[25] Nissim Maruani、Maks Ovsjanikov、Pierre Alliez 和 Mathieu Desbrun。PONQ：基于神经 QEM 的网格表示。载于 IEEE/CVF 计算机视觉与模式识别会议论文集，第 3647–3657 页，2024。2[26] Lars Mescheder、Michael Oechsle、Michael Niemeyer、Sebastian Nowozin 和 Andreas Geiger。占据网络：在函数空间中学习三维重建。载于 IEEE/CVF 计算机视觉与模式识别会议论文集，第 4460–4470 页，2019。2[27] 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。2[28] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga 等。PyTorch：一款命令式风格、高性能深度学习库。NeurIPS，32，2019。6

[29] Songyou Peng、Michael Niemeyer、Lars Mescheder、Marc Pollefeys 和 Andreas Geiger。卷积占据网络。载于欧洲计算机视觉会议，第 523–540 页。Springer，2020。2[30] Rohan Sawhney。FCPW：西部最快的最近点算法，2021。7[31] Nicholas Sharp 和 Maks Ovsjanikov。PointTriNet：三维点集的学习型三角剖分。载于欧洲计算机视觉会议，第 762–778 页。Springer，2020。2[32] Tianchang Shen、Jun Gao、Kangxue Yin、Ming-Yu Liu 和 Sanja Fidler。深度行进四面体：一种用于高分辨率三维形状合成的混合表示。NeurIPS，2021。1，2，5，6，7[33] Tianchang Shen、Zhaoshuo Li、Marc Law、Matan Atzmon、Sanja Fidler、James Lucas、Jun Gao 和 Nicholas Sharp。Spacemesh：用于学习流形曲面网格的连续表示。载于 SIGGRAPH Asia 2024 会议论文集，第 1–11 页，2024。2[34] Vincent Sitzmann、Julien Martel、Alexander Bergman、David Lindell 和 Gordon Wetzstein。具有周期激活函数的隐式神经表示。神经信息处理系统进展，33:7462–7473，2020。2[35] Sanghyun Son、Matheus Gadelha、Yang Zhou、Zexiang Xu、Ming C Lin 和 Yi Zhou。Dmesh：通用网格的可微表示。CoRR，2024。2[36] Matthew Tancik、Pratul Srinivasan、Ben Mildenhall、Sara Fridovich-Keil、Nithin Raghavan、Utkarsh Singhal、Ravi Ramamoorthi、Jonathan Barron 和 Ren Ng。傅里叶特征使网络能够学习低维域中的高频函数。神经信息处理系统进展，33:7537–7547，2020。2[37] Maxim Tatarchenko、Alexey Dosovitskiy 和 Thomas Brox。八叉树生成网络：用于高分辨率三维输出的高效卷积架构。载于 IEEE 国际计算机视觉会议论文集，第 2088–2096 页，2017。2[38] Li Wang、Franck H´etroy-Wheeler 和 Edmond Boyer。关于从隐式形式进行体积形状重建。载于欧洲计算机视觉会议，2016。2[39] Zixiong Wang、Pengfei Wang、Peng-Shuai Wang、Qiujie Dong、Junjie Gao、Shuangmin Chen、Shiqing Xin、Changhe Tu 和 Wenping Wang。Neural-imls：用于曲面重建的自监督隐式移动最小二乘网络。IEEE 可视化与计算机图形学汇刊，30(8):5018–5033，2023。2[40] Zimo Wang、Cheng Wang、Taiki Yoshino、Sirui Tao、Ziyang Fu 和 Tzu-Mao Li。Hotspot：具有渐近充分条件的符号距离函数优化。载于 CVPR，2025。2，6[41] Francis Williams、Matthew Trager、Joan Bruna 和 Denis Zorin。神经样条：使用无限宽神经网络拟合三维曲面。载于 IEEE/CVF 计算机视觉与模式识别会议论文集，第 9949–9958 页，2021。2[42] Jane Wu、Diego Thomas 和 Ronald Fedkiw。通过法线图进行着装人体的稀疏视角三维重建。载于 IEEE/CVF 冬季计算机视觉应用会议（WACV），2025。2，5[43] Wang Yifan、Lukas Rahmann 和 Olga Sorkine-Hornung。利用隐式位移场实现几何一致的神经形状表示。*arXiv* 预印本 *arXiv:2106 05187*，20212。。[44] 张晨和 Wenbing Tao。通过德洛内三角剖分学习网格化，实现三维形状表示。国际计算机视觉期刊，133(6):3413–3436，2025。2[45] 张晨、袁甘章勤和 Wenbing Tao。Dm-net：用于三维形状表示的德洛内网格网络。载于*2023 IEEE/CVF* 国际计算机视觉大会（*ICCV*），第14372–14382页。IEEE计算机学会，2023。2[46] 张晨、王文涛、李熙萌、廖欣瑶、Wan- juan Su 和 Wenbing Tao。根据点云进行高保真轻量级网格重建。载于计算机视觉与模式识别会议论文集，第11739–11748页，2025。2[47] 周庆南和 Alec Jacobson。Thingi10k：包含10,000个3D打印模型的数据集。*arXiv* 预印本 *arXiv:1605.04797*，2016。6

DCCVT：可微裁剪质心 Voronoi 镶嵌

补充材料

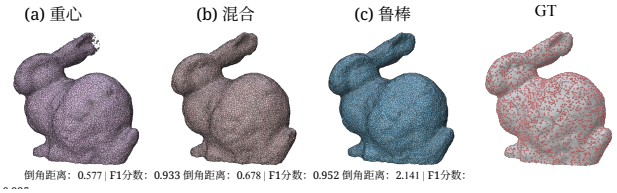


图 7. 重心投影方法 (a)、混合投影方法 (b) 和鲁棒投影方法 (c) 与真实值（GT）的视觉对比。

方法	CD ↓	F1 ↑	NC ↑
重心	3.524	0.675	0.920
混合	2.418	0.686	0.926
鲁棒（我们的方法）	2.326	0.684	0.930

表 3。对比我们的鲁棒投影、混合插值和重心插值的消融实验。报告的数值是使用我们的近邻采样/上采样方法，并基于未收敛的 SDF 表示，在 Thingi32 数据集上得到的平均误差。

本补充文档提供了以下方面的更多详细信息：

- 我们的鲁棒平面拟合方法及其对结果的影响（第 7节）；
- 公式（10）的详细推导（第8节）；
- 对更密集的点云如何影响我们的优化过程的分析（第 9节）；
- 针对 Thingi32 数据集开展的 SDF 正则化效果消融实验（第 10节）；
- 补充信息和结果（第11节）；

7. 鲁棒平面拟合

公式（5）定义了我们的鲁棒平面拟合最小化问题，该问题用于顶点投影步骤。本节将更详细地介绍如何执行此优化。我们还将展示一项消融实验，以证明我们的鲁棒投影的有效性。

公式（31）的解。 该问题的解对应于协方差矩阵的最小特征值所对应的特征向量

$$\mathbf{C}_j = \frac{1}{4}\sum_{i\in\mathcal{T}_j} \Delta\mathbf{s}'_i\,\Delta\mathbf{s}'^{\top}_i. \tag{24}$$

并定义中心化坐标 $\Delta\mathbf{s}'_i = \mathbf{s}'_i - \bar{\mathbf{s}}'_j$。基于这些坐标，我们计算投影采样点的协方差矩阵$\mathbf{C}_j$：

$$\mathbf{C}_j = \frac{1}{4}\sum_{i\in\mathcal{T}_j} \Delta\mathbf{s}'_i\,\Delta\mathbf{s}'^{\top}_i. \tag{25}$$

平面法向量$\mathbf{n}_j$ 由$\mathbf{C}_j$的最小特征值所对应的特征向量给出，并归一化为$\mathbf{n}_j = \mathbf{e}_{\min}/\|\mathbf{e}_{\min}\|$，其中$\mathbf{e}_{\min}$ 是$\mathbf{C}_j$中对应于最小特征值的特征向量。

证明。 为避免对$\mathbf{n}_j$ 进行显式归一化，我们首先将问题重新表述为

$$\arg\min_{\mathbf{n}_j} \frac{1}{4}\sum_{i\in\mathcal{T}_j}\left(\mathbf{n}_j\cdot\Delta\mathbf{s}'_i\right)^2,\ \text{subject to}\ \|\mathbf{n}_j\|^2 = 1. \tag{26}$$

接着，我们通过拉格朗日形式对新增约束进行建模：

$$\begin{aligned}
\mathcal{L}(\mathbf{n}_j,\lambda) &= \frac{1}{4}\sum_{i\in\mathcal{T}_j}\left(\mathbf{n}_j\cdot\Delta\mathbf{s}'_i\right)^2 - \lambda\left(\|\mathbf{n}_j\|^2-1\right)\\
&= \frac{1}{4}\sum_{i\in\mathcal{T}_j}\left(\mathbf{n}_j^{\top}\Delta\mathbf{s}'_i\right)\left(\Delta\mathbf{s}'^{\top}_i\mathbf{n}_j\right) - \lambda\left(\mathbf{n}_j^{\top}\mathbf{n}_j-1\right)\\
&= \mathbf{n}_j^{\top}\left(\frac{1}{4}\sum_{i\in\mathcal{T}_j}\Delta\mathbf{s}'_i\,\Delta\mathbf{s}'^{\top}_i\right)\mathbf{n}_j - \lambda\left(\mathbf{n}_j^{\top}\mathbf{n}_j-1\right)\\
&= \mathbf{n}_j^{\top}\mathbf{C}_j\mathbf{n}_j - \lambda\left(\mathbf{n}_j^{\top}\mathbf{n}_j-1\right).
\end{aligned} \tag{27}$$

通过 $\mathbf{n}_j$ 推导该方程，并将结果设为 0，可得

$$\mathbf{C}_j\mathbf{n}_j = \lambda\mathbf{n}_j, \tag{28}$$

因此，$\mathbf{n}_j$ 是与特征值 λ相关联的$\mathbf{C}_j$ 的特征向量。到该平面的平均距离对应于$\mathbf{n}_j^{\top}\mathbf{C}_j\mathbf{n}_j = \lambda\,(\mathbf{n}_j^{\top}\mathbf{n}_j) = \lambda$。因此，我们通过寻找具有最小特征值的特征向量来最小化该问题。 □

我们鲁棒投影的消融实验。图7和表3 展示了在近邻采样/上采样设置下，我们的鲁棒投影（c）与重心法（a）和混合方法（b）的性能对比。重心法是指利用重心坐标获得插值有符号距离场和梯度值来投影顶点，然后根据公式（8）将其投影到零水平集上。混合方法仅在顶点位于其四面体内部时采用这种重心投影；否则，则改用鲁棒投影。我们观察到，重心法产生的误差最大，因为它无法应对

外推导致的不稳定性。混合方法缓解了这一问题，并显著降低了误差，但仍可能出现异常的空间梯度，导致优化过程中出现轻微的不稳定性。在所有情况下始终采用我们的鲁棒投影，可获得最稳定、最准确的结果。

8. 式（10）的详细推导

证明。 定义四面体 $\mathcal{T}_j$ 的质心和平均 SDF 值如下

$$\bar{\mathbf{s}}_j = \tfrac{1}{4}\sum_{i\in\mathcal{T}_j}\mathbf{s}_i, \quad \bar{\phi}_j = \tfrac{1}{4}\sum_{i\in\mathcal{T}_j}\phi_i. \tag{29}$$

我们将以下问题最小化

$$\min_{\nabla\phi^{(j)}} \frac{1}{4}\sum_{i\in\mathcal{T}_j}\left((\bar{\phi}_j + (\mathbf{s}_i-\bar{\mathbf{s}}_j)^\top\nabla\phi^{(j)}) - \phi_i\right)^2. \tag{30}$$

为了更紧凑地表述这个问题，我们定义相对于质心和平均值的中心化坐标向量和符号距离函数值向量：

$$\Delta\mathbf{S}_j = \begin{bmatrix}\mathbf{s}_A-\bar{\mathbf{s}}_j\\ \mathbf{s}_B-\bar{\mathbf{s}}_j\\ \mathbf{s}_C-\bar{\mathbf{s}}_j\\ \mathbf{s}_D-\bar{\mathbf{s}}_j\end{bmatrix}, \quad \Delta\Phi_j = \begin{bmatrix}\phi_A-\bar{\phi}_j\\ \phi_B-\bar{\phi}_j\\ \phi_C-\bar{\phi}_j\\ \phi_D-\bar{\phi}_j\end{bmatrix}. \tag{31}$$

式（30）中的问题于是变为

$$\arg\min_{\nabla\phi^{(j)}} \frac{1}{4}\left\|\Delta\mathbf{S}_j\nabla\phi^{(j)} - \Delta\Phi_j\right\|^2 =$$
$$\arg\min_{\nabla\phi^{(j)}} \frac{1}{4}\left((\nabla\phi^{(j)})^\top\mathbf{G}_j\nabla\phi^{(j)} - 2(\nabla\phi^{(j)})^\top\Delta\mathbf{S}_j^\top\Delta\Phi_j\right) \tag{32}$$

其中 $\mathbf{G}_j = \Delta\mathbf{S}_j^\top\Delta\mathbf{S}j$ 是局部格拉姆矩阵。对上述函数关于 $\nabla\phi^{(j)}$ 求导并令结果为 0，最终得到

$$\nabla\phi^{(j)} = \mathbf{G}_j^{-1}\Delta\mathbf{S}_j^\top\Delta\Phi_j = \mathbf{W}_j\Delta\Phi_j. \tag{33}$$

□

9. 提高目标点云密度

图10 比较了 MTet [32] 与我们的方法在未收敛的符号距离函数下、采用不同目标点云密度时的表现。提高目标点云中的点密度对两种方法都有帮助。不过，与 MTet 基线相比，我们的方法结合近邻采样/上采样始终取得最佳结果。

10. SDF 正则化消融实验

如图 6所示，SDF 正则化对于移除优化网格周围细小的悬浮零交叉区域至关重要。尽管这些伪影通常很小，但它们代表着拓扑噪声，会降低最终重建结果的整洁度。

方法	CD ↓	F1 ↑	NC ↑	HD ↓
$\lambda_{\text{Eik}}=0.00,\ \lambda_{\text{H}}=0.00$	2.241	0.690	0.929	464.3
$\lambda_{\text{Eik}}=0.02,\ \lambda_{\text{H}}=0.10$	2.276	0.680	0.930	369.8
$\lambda_{\text{Eik}}=2.00,\ \lambda_{\text{H}}=10.0$	5.940	0.634	0.904	499.7

表 4. SDF 正则化消融研究。我们报告了 Thingi32 数据集在不同正则化权重值 λ_{sdf} 下的平均性能。

定量而言，这种权衡体现在表4中。我们发现，施加轻度 SDF 正则化（λ_{Eik} =0.02）能显著改善豪斯多夫距离（HD），这证实了此举能够抑制这些虚假离群点。虽然这种平滑约束会导致表面精度略有下降——表现为倒角距离略微增加，且 F1 分数低于未正则化基线，但它提供了必要的平衡。它能确保生成的符号距离函数在拓扑上保持一致，避免高正则化水平（$\lambda_{\text{Eik}}=2.00$）造成的严重性能退化。

11. 补充信息与结果

为确保比较严谨且公平，MTet 和 DCCVT 均采用相同的初始采样点进行初始化，并使用相同的超参数（$LR=5\times10^{-4}$, $\beta=$(0.8,0.99)）通过 Adam 优化器进行优化，迭代 1000 次。两者的一个关键区别在于优化过程：MTet 仅由提取顶点的倒角距离驱动，而 DCCVT 则引入 CVT 正则化以保持离散化稳定性。实验表明，尽管两种方法在规则网格上取得了相近的结果，但当采用非均匀站点分布（例如近邻采样）初始化时，DCCVT 表现出更强的鲁棒性，而 MTet 往往难以应对。这种鲁棒性表明，DCCVT 的性能提升源自其数学表述，而非特定的初始化优势。

图 11 展示了我们的方法在 SDF 已收敛时的更多结果。对于 Voromesh 的结果，我们加入了一个非常小的 CVT 正则化项，以防止网格提取过程中出现共面性。这会增加 Voromesh 的运行时间，因为优化过程的每次迭代都必须重新计算德劳内三角剖分。我们发现这一项是必要的，尤其是在 SDF 表示尚未收敛时；在这种情况下，Voromesh 最终生成的站点分布经常出现共面配置，而 gDel3D 不支持这种配置 [5]。另一种做法是使用 CGAL 为 Voromesh 执行网格提取。

图 12 展示了我们方法的更多对比结果

在近邻采样和上采样设置的最佳情况下，针对已收敛和未收敛的符号距离函数，将我们的方法与 MTet 进行了比较。结果突显了我们方法的自适应性和表面正则化效果，能够显著改善表面离散化。

图 9显示，尽管重建质量通常在不同 SDF 状态下保持稳定，网格 ID 252119（图 8）在已收敛状态下却遭遇了灾难性故障。这与网格 ID 398259（图 6）的问题如出一辙：完全收敛的 SDF 提供了较差的初始化，使优化陷入局部极小值。较粗糙、未收敛的 SDF 则避免了这一问题，这表明较少精细的初始化有时能为优化循环提供更好的正则化。

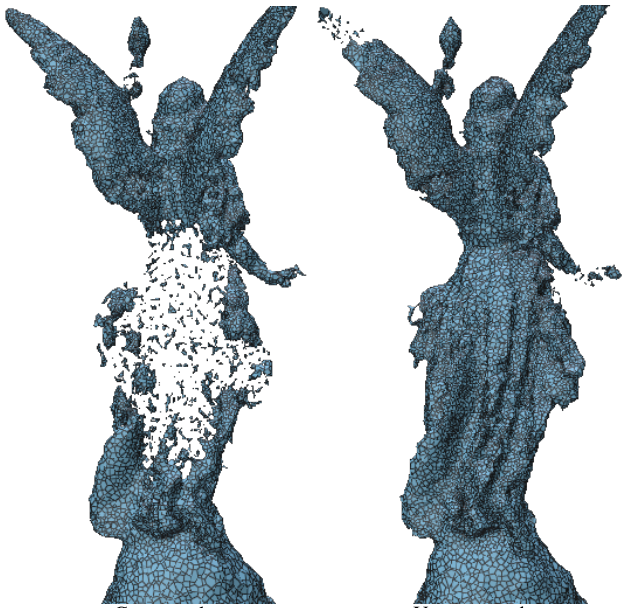


图 8。Hotspot 收敛后生成的 SDF 很难恢复，与图 6中的网格 ID 398259 类似。

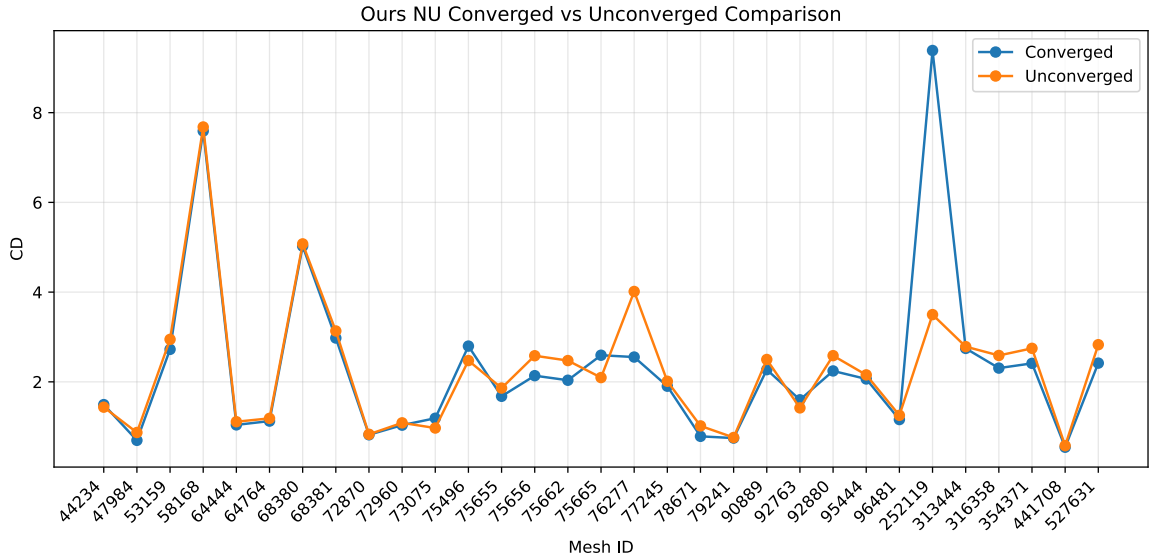


图 9　。已收敛与未收敛 Hotspot 符号距离场状态下每个网格的倒角距离指标对比

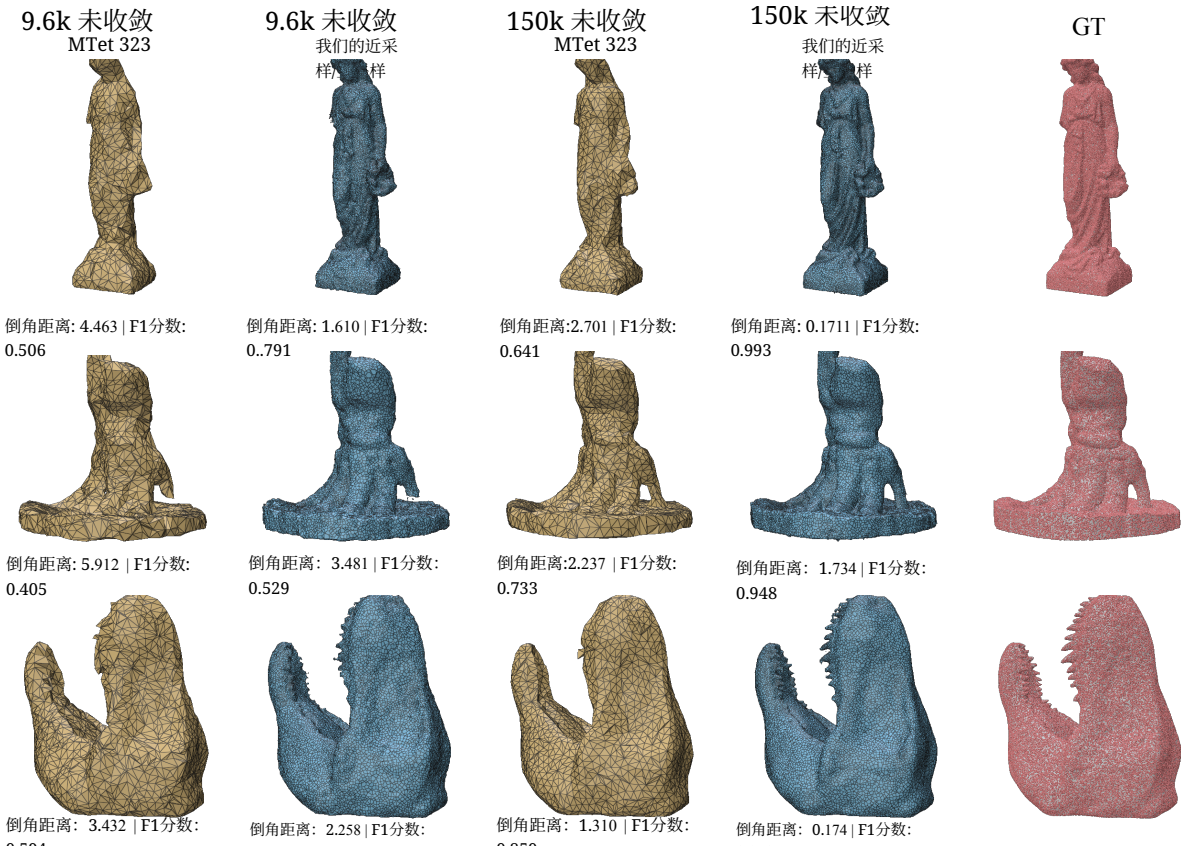


图 10。不同目标点云密度下 MTet 与我们的方法（采用近邻采样与上采样，NU）的对比。展示了使用未收敛符号距离函数的 $9.6k$ 和 $150k$ 个目标点时的结果。真实值（GT）使用 $150k$ 个采样点展示。

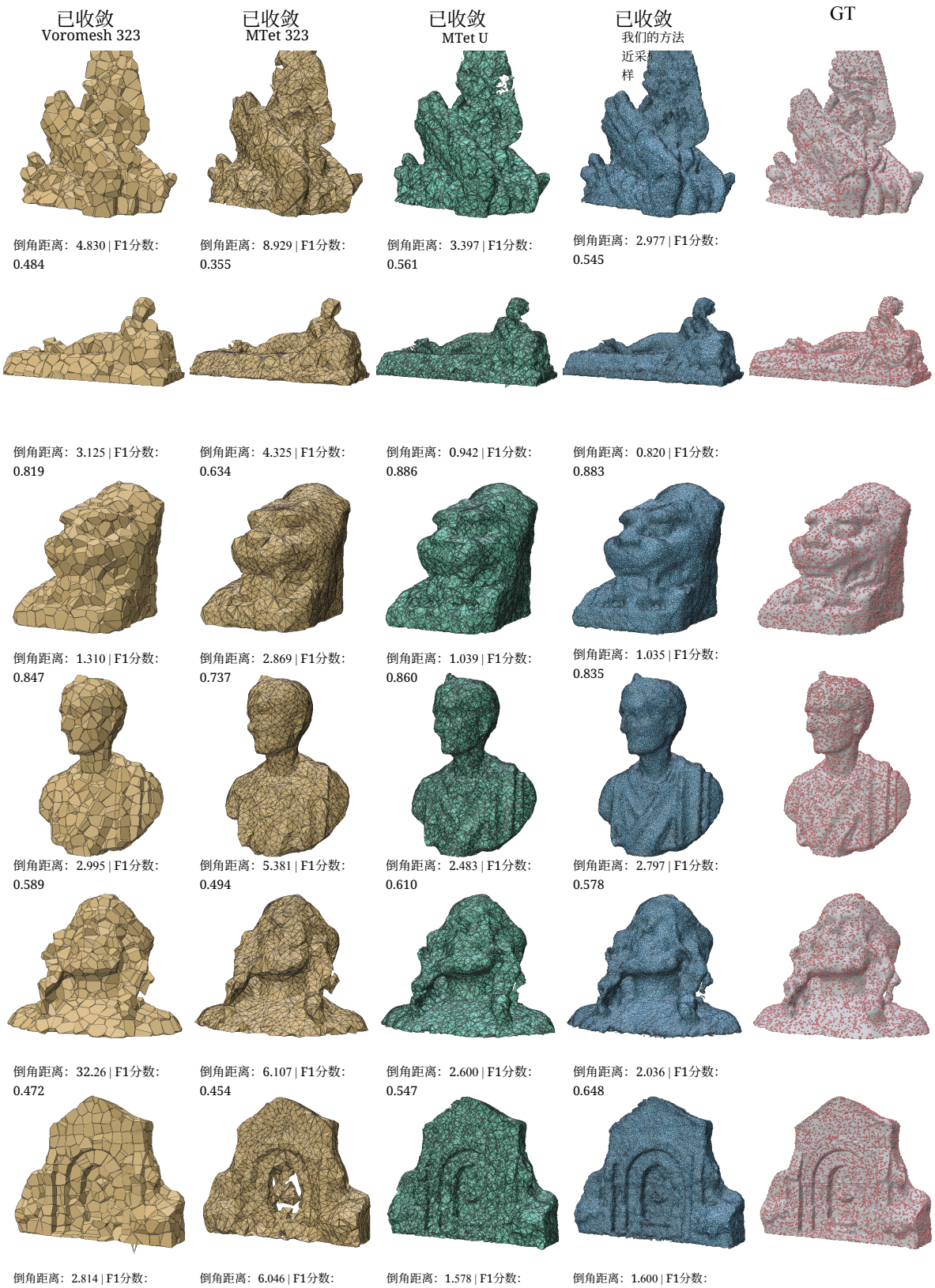


图 11. 在收敛的符号距离函数上，对比 Voromesh、MTet（使用和不使用我们的上采样）以及采用我们的近邻采样/上采样的方法。真实值（GT）在最后一列中显示，其中包含 9600 个采样点。

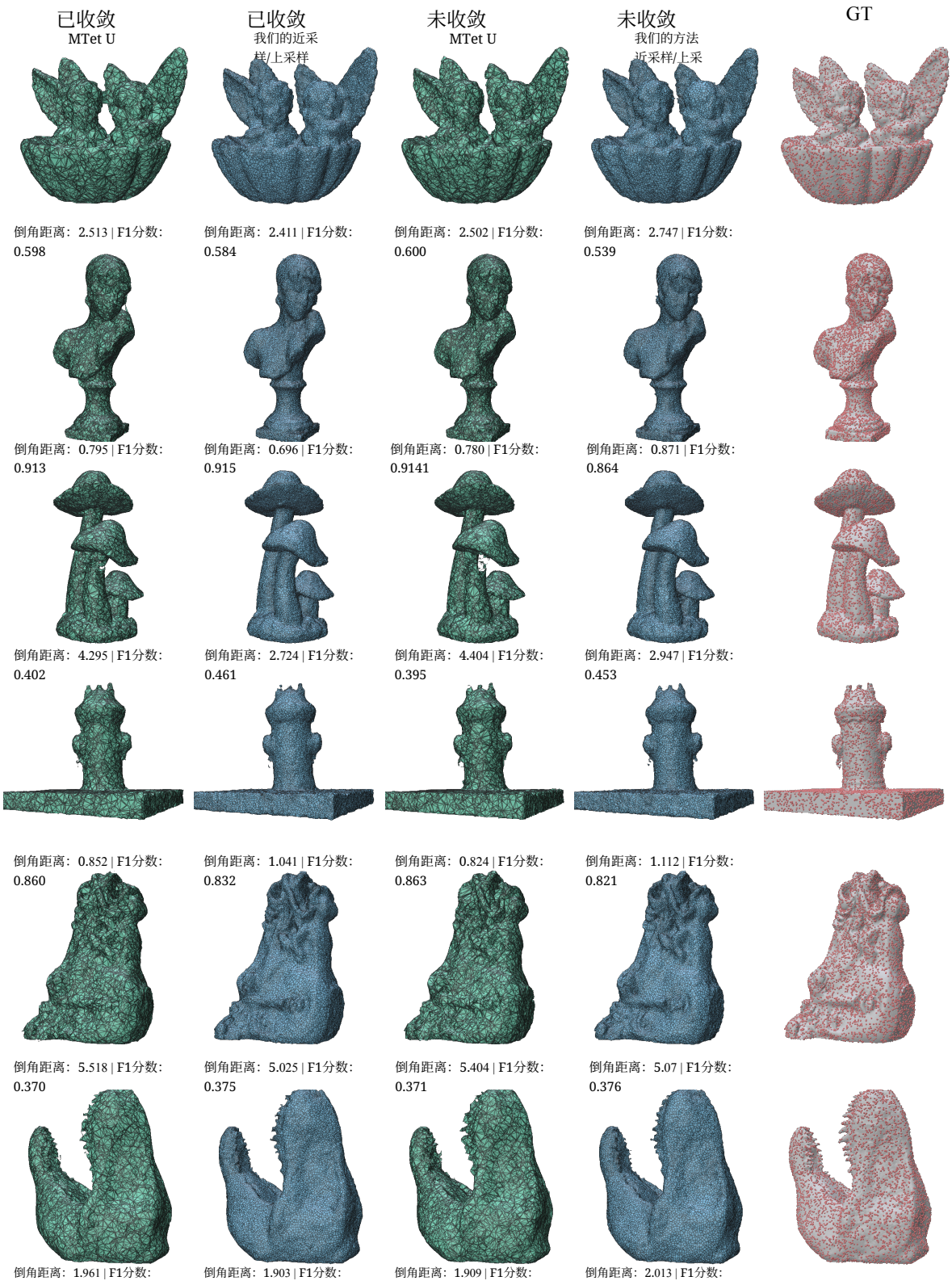


图 12. 在已收敛和未收敛的符号距离函数情况下，对比 MTet（采用我们的上采样）与我们的方法（采用我们的近邻采样/上采样）。