VOROFACET: JUNCTION-AWARE MESH EXTRACTION FROM NEURAL UNSIGNED DISTANCE FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsigned distance fields (UDFs) can represent surfaces of arbitrary topology, but recovering mesh connectivity from neural UDFs remains challenging, especially at non-manifold junctions where multiple sheets meet. To this end, we propose **VoroFacet**, a novel pipeline that extracts single-layer meshes from neural UDFs and reconstructs junction connectivity accurately through Voronoi facets. Specifically, we first use a near-surface double-layered proxy derived from the UDF as a local spatial partition for facet selection, where the facets belong to the Voronoi diagram of the sites generated by an adaptive ball cover of the proxy. We then estimate junctions from the mesh of this first extraction and place balls along them, so that in a re-extraction process, the resulting cospherical sites make incident sheets share Voronoi vertices and edges at these junctions. Finally, we propose a three-stage optimization to refine the geometry of the extracted mesh. Experiments demonstrate superior reconstruction quality over state-of-the-art methods, particularly at non-manifold junctions. The source code will be publicly available.

1 INTRODUCTION

With the rapid development of 3D vision, neural implicit fields have attracted increasing attention as a representation of geometry. The most widely used are binary occupancy fields (BOFs) (Mescheder et al., 2019) and signed distance fields (SDFs) (Park et al., 2019), but both rely on an inside/outside classification and can therefore represent only watertight surfaces. Unsigned distance fields (UDFs) (Chibane et al., 2020) avoid this by assigning to each spatial location only its distance to the nearest surface point, and can therefore represent surfaces of arbitrary topology, including open boundaries, non-orientable surfaces, and non-manifold junctions. However, converting neural UDFs into meshes that faithfully represent the underlying surface remains difficult, which limits their use in downstream applications that require explicit geometry.

The first difficulty is that a UDF is non-negative, so field values alone cannot indicate surface crossings; classical sign-based isosurfacing methods (Lorensen & Cline, 1998; Ju et al., 2002) do not apply. The second is that the ideal UDF is not differentiable on the surface, nor on the medial axis, where the nearest point on the surface is not unique. Since neural UDFs are typically smooth, they poorly approximate the ideal field in the narrow regions around them, where the distance values are biased and the gradients are substantially less reliable (Zhang et al., 2023).

Despite these difficulties, existing methods extract meshes from neural UDFs with remarkable geometric accuracy, but their connectivity is often incorrect at non-manifold junctions, where three or more sheets meet. Since the field is unsigned, most methods (Guillard et al., 2022; Zhou et al., 2024; Ren et al., 2023; Stella et al., 2025; Chen et al., 2026; Zhang et al., 2023; Chen et al., 2022; Bao et al., 2026) infer connectivity from local surface evidence, such as gradients, using handcrafted rules or learned networks. At such a junction, these sheets create additional medial axes that degrade this evidence, so these methods tend to leave holes or spurious connections. By contrast, offset-based methods (Hou et al., 2023; Chen et al., 2025) deform the more reliable positive level set of the neural UDF toward the surface, but the near-surface double-layered approximation they produce leaves a gap at a junction instead of a shared connection.

Different from existing methods, we introduce **VoroFacet**, which reconstructs the target surface as a selected set of facets in an ambient Voronoi diagram. Our starting observation is that the near-surface double-layered approximation of DCUDF2 (Chen et al., 2025) can serve as a useful proxy

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

even when its connectivity is unsuitable for the desired output. Instead of transferring the proxy's vertices or faces, we use it only to guide the selection of facets from the Voronoi diagram, so that the output connectivity is constructed anew rather than inherited. Since Voronoi cells partition the space without gaps or overlaps, the selected facets meet along shared Voronoi edges: a smooth sheet is extracted as a single layer, and sheets meeting at a junction are joined along common edges rather than left with holes, gaps, or spurious connections. Moreover, as connectivity follows from this spatial partition rather than from gradient cues or global inside/outside labels, it is less sensitive to the approximation errors of neural UDFs and imposes no restriction on the topology of the target, handling open boundaries, non-orientable surfaces, and non-manifold junctions within a single pipeline.

Our main contributions are summarized as follows.

- We propose VoroFacet, which formulates mesh extraction from neural UDFs as facet selection in a Voronoi diagram guided by a near-surface double-layered proxy, producing single-layer meshes for surfaces of arbitrary topology.
- We design a junction-aware re-extraction strategy that estimates non-manifold junctions and enforces shared connectivity among the sheets meeting there, recovering junctions that existing methods leave as holes, spurious connections, or gaps.
- We develop an algorithm to extract junction curves from CAD meshes to quantitatively evaluate junction reconstruction. Extensive experiments on diverse datasets show that VoroFacet achieves state-of-the-art junction reconstruction while maintaining competitive surface accuracy.

2 Related Work

Existing UDF mesh extraction methods differ mainly in how they determine mesh connectivity, that is, how surface points are connected into faces.

Connectivity from Inferred Pseudo-signs. Pseudo-sign methods infer from local values and gradients, by handcrafted rules or learned classifiers, which corners of each grid cell lie on opposite sides of the surface, and let a Marching Cubes lookup table determine the connectivity within the cell (Guillard et al., 2022; Zhou et al., 2024; Ren et al., 2023; Stella et al., 2024; 2025). MIND (Chen et al., 2026) assembles such local two-sided fields into a global multi-label field to extract non-manifold junctions. However, at a junction the corners of a cell cannot be split into two sides, so the local two-sided field is ill-defined and the gradients behind it are unreliable, leaving holes or spurious connections; a global labeling is further undefined for non-orientable surfaces.

Connectivity from Cell Adjacency. Dual methods follow Dual Contouring (Ju et al., 2002). They place one representative point in each cell that contains the surface, and cell adjacency proposes candidate faces between neighboring cells, which handcrafted rules or learned networks accept or reject using local surface evidence such as values or gradients (Zhang et al., 2023; Chen et al., 2022; Bao et al., 2026). Since such decisions are heuristic and rely on local surface evidence, these methods carry no topological guarantee and can be wrong where that evidence is unreliable, as at a junction, so the output may contain holes or spurious connections. Other variants instead base the decision on an explicit construction from distances or gradients of an ideal UDF (Kohlbrenner & Alexa, 2026; Wang et al., 2026a;b), which a neural UDF only approximates.

Connectivity from Offset Surfaces. Offset-based methods decide connectivity away from the zero level set: they extract a small positive level set as a manifold double cover of the target, deform it toward the surface, and cut a single layer out of the cover (Hou et al., 2023; Chen et al., 2025). Gradients thus guide a deformation rather than decide connectivity, so these methods are less sensitive to gradient errors. Their result, however, inherits the connectivity of this cover: for non-manifold or non-orientable targets no valid single layer exists, so the deformed cover is returned as duplicated geometry. The layers bend around a junction and leave a gap instead of a shared connection.

Our Positioning. In summary, when applied to neural UDFs, existing methods either derive connectivity from surface evidence that degrades at a junction, or inherit connectivity from an offset surface whose layers leave a gap instead of a shared connection. OffsetAxis (Huang et al., 2026), the most recent of the latter, reconstructs the medial axis of the offset volume but connects it heuristically, without any topological guarantee. Our VoroFacet keeps the robust double cover but selects facets from an ambient Voronoi diagram with the proxy, yielding a single layer in which sheets meeting at a junction share Voronoi edges by construction rather than by heuristic connection.

3 PRELIMINARIES

VoroFacet selects facets from the Voronoi diagram of ball-induced sites, which we recall below.

Voronoi Diagrams. Let $\mathcal{P} \subset \mathbb{R}^3$ be a finite set of sites. For a site $\mathbf{p}_a \in \mathcal{P}$, its Voronoi cell is

$$V_a = \left\{ \mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x} - \mathbf{p}_a\| \leq \|\mathbf{x} - \mathbf{p}_b\|, \quad \forall \mathbf{p}_b \in \mathcal{P} \setminus \{\mathbf{p}_a\} \right\}. \tag{1}$$

Each inequality defines the half-space of points closer to $\mathbf{p}_a$ than to $\mathbf{p}_b$, bounded by the bisector plane of $\mathbf{p}_a$ and $\mathbf{p}_b$; thus V_a is the intersection of these half-spaces, hence convex. The collection of all such cells is the Voronoi diagram of $\mathcal{P}$. A Voronoi facet shared by V_a and V_b is the portion of the bisector plane of $\mathbf{p}_a$ and $\mathbf{p}_b$ clipped by the other half-spaces; we denote it by F_{ab}.

Ball-induced Voronoi Sites. VoroCrust (Abdelkader et al., 2020) can convert *mesh surface* into a volumetric Voronoi mesh, as shown in Fig. 1. Since arbitrarily placing the sites near the surface does not guarantee that the boundary of the mesh is aligned with the input surface (Fig. 1, middle), it generates sites from a ball cover with deliberate cosphericity (Fig. 1, right). The surface is first partitioned into smooth patches separated by sharp features and non-manifold junctions. Then, surface balls are centered within patches and kept disjoint from those in other patches, whereas feature balls are centered on patch boundaries and connect the surface balls of adjacent patches.

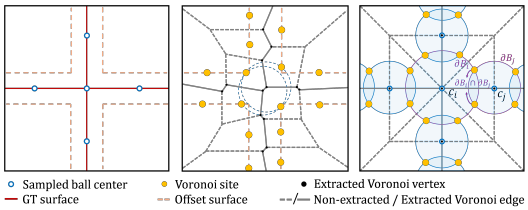

Figure 1: Site placement controls the connectivity of the Voronoi edges at a non-manifold junction, shown here in 2D. Left: the input surface. Middle: arbitrarily placed sites, which leave the junction without a shared vertex and add spurious edges in smooth regions. Right: ball-induced sites, which make the edges incident to the junction share a vertex.

For the resulting ball cover $\mathcal{B} = \{B_i = B(\mathbf{c}_i, r_i)\}$ with union $U(\mathcal{B}) = \bigcup_i B_i$, the Voronoi sites are the exposed triple-sphere intersections of the balls in the cover:

$$\mathcal{P}(\mathcal{B}) = \{\mathbf{p} \in \partial B_i \cap \partial B_j \cap \partial B_k : i < j < k,\ \mathbf{p} \notin \mathrm{int}(B_\ell), \quad \forall B_\ell \in \mathcal{B}\}. \tag{2}$$

The sites on ∂B_i are the closest sites to $\mathbf{c}_i$, so their cells share $\mathbf{c}_i$. If four of these sites are noncoplanar, $\mathbf{c}_i$ is a Voronoi vertex. Similarly, the sites on the intersection circle $\partial B_i \cap \partial B_j$ are the nearest sites to both centers, so their cells contain both centers and, by convexity, the segment $\overline{\mathbf{c}_i\mathbf{c}_j}$. If the circle contains at least three sites and both centers are Voronoi vertices, this segment is a shared Voronoi edge. Thus, the ball construction suppresses spurious facets in smooth regions and enforces shared Voronoi vertices or edges along sharp features and junctions, as visualized in Fig. 1.

It is worth noting that VoroCrust relies on an explicit input mesh both to build the ball cover and to locate the sharp features and non-manifold junctions, whereas a neural UDF provides no explicit surface.

4 PROPOSED METHOD

Given a neural UDF f_θ approximating the distance to an unknown target surface $\bar{\mathcal{S}}$, our goal is to recover a single-layer triangular mesh $\mathcal{M}$, with emphasis on the connectivity of surface sheets at non-manifold junctions. Since the field provides no explicit surface, we start from the near-surface double-layered proxy $\mathcal{D}$ of DCUDF2 (Chen et al., 2025), which is geometrically faithful but topologically incorrect; every sheet is duplicated, and every junction is buried in the duplicate-layer gap. We therefore treat $\mathcal{D}$ only as a geometric prior and rebuild connectivity from the ball-induced Voronoi sites.

Our pipeline, namely VoroFacet, proceeds in two passes, as illustrated in Fig. 2. In the first pass, we construct an adaptive ball cover of $\mathcal{D}$, with balls sized to fill the duplicate-layer gap. From this cover, Voronoi sites are generated using the exposed triple-sphere intersection rule, and a geometric facet-selection rule is applied to produce a single-layer intermediate mesh $\mathcal{M}_0$. In the second pass, candidate junction curves of $\mathcal{M}_0$ are consolidated into a junction graph $\mathcal{G}$; junction balls are then placed along $\mathcal{G}$, and a finer manifold ball cover is added within surface patches. From the refined cover, sites are again generated via the same rule, and the resulting sites around junctions are deliberately cospherical, inducing shared Voronoi vertices or edges. Applying the same facet-selection

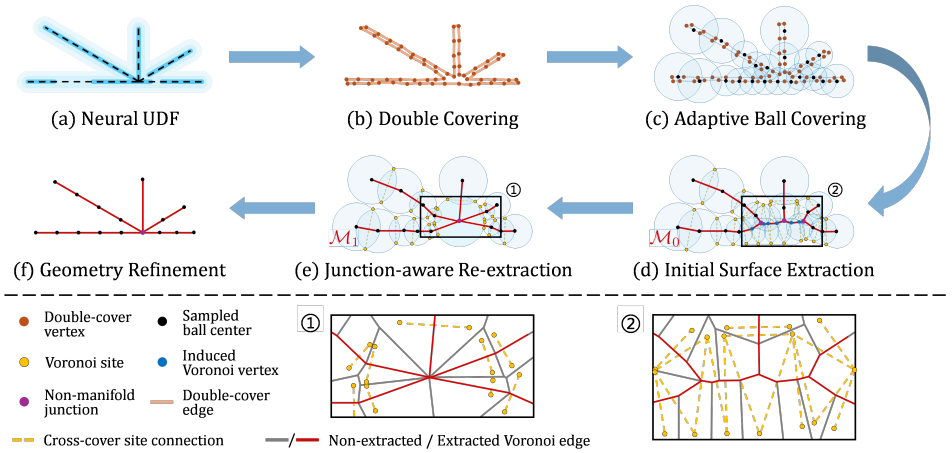

Figure 2: Illustration of the algorithmic pipeline of our VoroFacet on a 2D toy model with boundaries and non-manifold junctions.

rule yields $\mathcal{M}_1$. Thus, the two passes realize geometric observation, junction estimation, and connectivity construction, while the subsequent constrained remeshing and three-stage refinement carry out geometric refinement.

4.1 Initial Surface Extraction

Adaptive Ball Cover. Given the double-layered proxy $\mathcal{D}$, our aim is to construct a ball cover that fills the duplicate-layer gap, especially the wider gap near junctions, while keeping distinct sheets separate. We first set a baseline radius R^0_{gap}, the minimum admissible radius that ensures the desired gap coverage. However, using this radius uniformly would cover flat regions with many unnecessarily small balls, making the cover inefficient. We therefore initialize every ball with a larger radius R_{init} and shrink it only where it would bridge a gap between distinct sheets, never below R^0_{gap}.

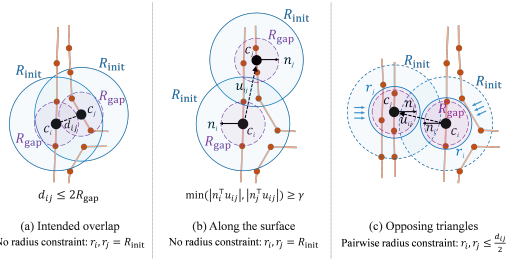

Figure 3: Adaptive ball cover construction. The balls are left unconstrained when the space between the two triangles has to be covered, which holds both for a pair closer than $2R^0_{\text{gap}}$ (left) and for a pair that extends along the surface (center). For an opposing pair, that space has to remain uncovered, so the radii are capped at half their distance (right).

VoroCrust prevents such bridging through co-smoothness, which allows balls to overlap only within smoothly connected surface patches. Unfortunately, the duplicated layers of $\mathcal{D}$ are disconnected, making this connectivity-based criterion inapplicable here. We instead introduce an explicit opposing-triangle test to prevent balls from bridging gaps between distinct sheets.

Specifically, for each triangle T_i of $\mathcal{D}$, we place a candidate ball $B_i = B(\mathbf{c}_i, r_i)$ centered at its centroid $\mathbf{c}_i$. For two nonadjacent triangles T_i and T_j, define $d_{ij} = |\mathbf{c}_i - \mathbf{c}_j|$ and $\mathbf{u}_{ij} = \frac{\mathbf{c}_i - \mathbf{c}_j}{d_{ij}}$. We declare T_i and T_j *opposing* when

$$2R^0_{\text{gap}} < d_{ij} < 2R_{\text{init}} \quad \text{and} \quad \min\left(|\mathbf{n}_i^\top \mathbf{u}_{ij}|, |\mathbf{n}_j^\top \mathbf{u}_{ij}|\right) \ge \gamma. \tag{3}$$

Since all radii start at R_{init}, balls may overlap. Overlap is allowed when the gap between the two triangles has to be filled, and when they extend along the surface rather than facing each other (Fig. 3, left and center). Only opposing pairs, which face each other across a wider separation, are constrained to $r_i, r_j \le \frac{d_{ij}}{2}$, so that their balls do not bridge the gap between distinct sheets (Fig. 3, right).

To ensure coverage, let ρ_i denote the radius required for a ball centered at $\mathbf{c}_i$ to cover its seed triangle T_i with a small margin. If $r_i < \rho_i$, we split T_i at its edge midpoints into four child triangles and recompute their balls; the required coverage radius is halved after each subdivision. Since $r_i \ge R^0_{\text{gap}}$, each triangle is subdivided at most $O\left(\log\left(1 + \rho_i / R^0_{\text{gap}}\right)\right)$ times. We then greedily

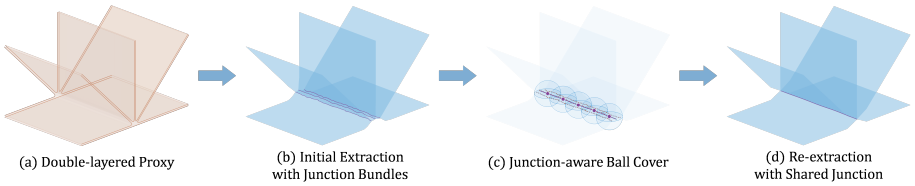

Figure 4: Illustration of junction-aware re-extraction. (a) The double-layered proxy leaves a gap around the non-manifold junction. (b) Initial extraction produces a bundle of nearby junction curves. (c) The curves are consolidated into a junction graph, along which junction balls are placed. (d) Re-extraction joins the incident sheets along a single shared junction curve.

select balls in decreasing order of radius, skipping those whose seed triangles are already covered, yielding $\mathcal{B}_0$ with $\mathcal{D} \subseteq U(\mathcal{B}_0)$.

Surface Extraction. For a ball cover $\mathcal{B}$ of $\mathcal{D}$, we construct the Voronoi diagram of the exposed sites via Eq. (2) and select each bounded facet F_{ab} that satisfies

$$\mathrm{vert}(F_{ab}) \subseteq U(\mathcal{B}) \quad \text{and} \quad \overline{\mathbf{p}_a\mathbf{p}_b} \cap \mathcal{D} \neq \emptyset. \tag{4}$$

This purely local test requires no global inside/outside labels and thus applies to open surfaces and non-manifold junctions. Applying this selection to $\mathcal{B}_0$ yields the initial single-layer mesh $\mathcal{M}_0$, whose non-manifold edges form candidate junction curves. Since several nearby curves may correspond to the same junction, we consolidate them in the second pass.

4.2 Junction-Aware Re-extraction

Junction Graph Estimation. Edges of $\mathcal{M}_0$ incident to more than two facets form candidate junction curves. Since the first pass lacks junction positions, a junction on $\bar{\mathcal{S}}$ where four or more sheets meet may appear in $\mathcal{M}_0$ as a bundle of nearby but disjoint curves (Fig. 4b). We densely sample these curves, then greedily select cluster seeds with spacing δ_G larger than the typical bundle width, so that seeds align along the curves rather than across them. Assigning each sample to its nearest seed and connecting clusters that contain adjacent samples gives the junction graph $\mathcal{G}$, with cluster centroids as vertices (Fig. 4c).

Junction-aware Cover and Re-extraction. We then place overlapping junction balls $\mathcal{B}_G$ of radius R_G along $\mathcal{G}$, sized to fill the junction gaps and isolate adjacent sheets. Let $U_G = U(\mathcal{B}_G)$. We clip $\mathcal{D}$ to the region outside U_G, cutting triangles that cross the junction balls. The remaining surface has only thinner duplicate-layer gaps. We therefore repeat the cover-and-extraction step of Section 4.1 on the clipped triangles, with only two changes: the balls are placed on the remaining triangles outside U_G, and the minimum admissible radius is reduced from R^0_{gap} to R^1_{gap}. This lower threshold marks more pairs as opposing, thereby restoring gaps between distinct sheets that are wider than $2R^1_{\mathrm{gap}}$ but were bridged by the first cover.

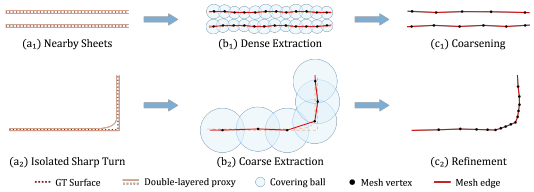

Figure 5: 2D illustration of geometry-adaptive remeshing. Top: two closely spaced flat sheets lead to an unnecessarily dense initial extraction, which remeshing coarsens. Bottom: an isolated sharp turn is extracted too coarsely, so remeshing adds vertices around it.

We then generate exposed sites from the refined cover $\mathcal{B}_1$ via Eq. (2) and apply the facet selection, i.e., Eq. (4), to obtain $\mathcal{M}_1$, a single-layer, self-intersection-free mesh whose non-manifold incidence is dictated by the junction balls (Fig. 4d).

4.3 Remeshing and Geometry Refinement

Geometry-adaptive Remeshing. The triangle density of $\mathcal{M}_1$ follows inter-sheet spacing rather than local geometry, over-sampling closely spaced flat sheets while leaving detailed regions coarse; this limits the accuracy of subsequent fitting (Fig. 5b). We therefore replace it with a geometry-adaptive triangulation. Since isotropic remeshing requires manifold inputs, we cut $\mathcal{M}_1$ along its non-manifold curves and remesh each sheet independently, resampling non-manifold curves and open

boundaries at the target resolution as constraints. Using CGAL (Pouget & Cazals, 2024; Coeurjolly et al., 2024), we estimate curvature, derive target edge lengths, and perform constrained isotropic remeshing (Fig. 5c); the sheets are then reassembled through their shared constrained curves.

Three-stage Geometry Refinement. Remeshing improves sampling but not vertex positions: non-manifold vertices remain at the cluster centroids of the candidate junction curves in $\mathcal{M}_0$. These centroids are derived from point clusters rather than from the intersections of the incident sheet extensions, so they may deviate substantially from the true junction. Denser sampling near junctions also places adjacent manifold vertices close to these incorrect positions. Jointly minimizing the UDF and Laplacian would still pull each non-manifold vertex toward a local centroid of its neighbors, not the true junction. We therefore decouple refinement into three stages: (I) fit the manifold sheets away from junctions; (II) reposition the non-manifold vertices from the fitted sheets; (III) refine the manifold surface with the non-manifold vertices fixed (Fig. 6).

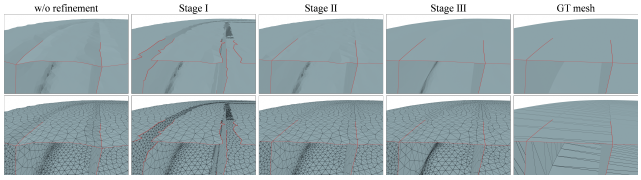

Figure 6: Visual effect of the three-stage geometry refinement at a non-manifold junction. From left to right: before refinement; Stage I, fitting manifold sheets; Stage II, repositioning non-manifold vertices; Stage III, refining the manifold surface with non-manifold vertices fixed; and the ground truth. Shaded and wireframe views are shown in the top and bottom rows, respectively.

To stabilize the fitting and keep the vertices uniformly distributed, we use the Laplacian $(\Delta \mathbf{x})_v = \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)}(\mathbf{x}_u - \mathbf{x}_v)$, where $\mathcal{N}(v)$ is the one-ring neighbor set for manifold vertices and the adjacent curve-neighbor set for non-manifold vertices, the latter denoted Δ_{curve}.

Stage I: Manifold-sheet fitting. First, we remove the non-manifold vertices, denoted $\mathcal{V}_N$, and their incident faces, then optimize the remaining manifold sheets $\mathcal{M}^- = (\mathcal{V}^-, \mathcal{E}^-, \mathcal{F}^-)$ by

$$E_{\text{sheet}} = \lambda_u \sum_{v\in\mathcal{V}^-} \big|f_\theta(\mathbf{x}_v)\big| + \lambda_e \sum_{\overline{uv}\in\mathcal{E}^-} \left(\frac{\|\mathbf{x}_u - \mathbf{x}_v\| - \ell^0_{uv}}{\ell^0_{uv}}\right)^2, \tag{5}$$

where ℓ^0_{uv} is the post-remeshing edge length. The UDF term pulls the sheets onto $\bar{\mathcal{S}}$, while the edge-length term prevents neighboring vertices from collapsing into the same local UDF minimum.

Stage II: Non-manifold vertex fitting. Next, we restore the non-manifold vertices $\mathcal{V}_N$ and their incident faces, keeping all manifold vertices fixed. For each $v \in \mathcal{V}_N$, we collect a set $\mathcal{F}^q_v$ of nearby fitted faces from each incident sheet. Assuming local planarity near the junction, each face $f \in \mathcal{F}^q_v$ defines a plane through its centroid $\mathbf{c}_f$ with normal $\mathbf{n}_f$, and the junction vertex should lie on all such planes. This leads to

$$E_{\text{nm}} = \frac{\lambda_q}{N_q}\sum_{v\in\mathcal{V}_N}\sum_{f\in\mathcal{F}^q_v}\big[\mathbf{n}_f^\top(\mathbf{x}_v - \mathbf{c}_f)\big]^2 + \frac{\lambda_c}{|\mathcal{V}_N|}\sum_{v\in\mathcal{V}_N}\|\Delta_{\text{curve}}\mathbf{x}_v\|^2 + \frac{\lambda_a}{|\mathcal{V}_N|}\sum_{v\in\mathcal{V}_N}\|\mathbf{x}_v - \mathbf{x}^0_v\|^2, \tag{6}$$

with $N_q = \sum_{v\in\mathcal{V}_N}|\mathcal{F}^q_v|$ and $\mathbf{x}^0_v$ the position of v before this stage. The curve Laplacian smooths the junction curve, and the weak anchor limits drift where the plane constraints are degenerate.

Stage III: Manifold refinement. Finally, with the non-manifold and boundary vertices fixed, we refine the manifold surface following DCUDF2 (Chen et al., 2025). Manifold vertices within three graph rings of the non-manifold curves, denoted $\mathcal{V}_B$, receive only Laplacian regularization, whereas the remaining vertices $\mathcal{V}_F$ and the faces $\mathcal{F}_F$ whose vertices all lie in $\mathcal{V}_F$ also minimize their UDF values:

$$E_{\text{ref}} = \lambda_l \sum_{v\in\mathcal{V}_B\cup\mathcal{V}_F}\omega_v\|\Delta\mathbf{x}_v\|^2 + \lambda_u\left(\sum_{v\in\mathcal{V}_F}\alpha_v\big|f_\theta(\mathbf{x}_v)\big| + \sum_{f\in\mathcal{F}_F}\alpha_f\big|f_\theta(\mathbf{c}_f)\big|\right), \tag{7}$$

where ω_v is the Laplacian weight and α_v, α_f are per-vertex and per-face weights. The output thus retains the connectivity of $\mathcal{M}_1$ while aligning both manifold and non-manifold vertices with $\bar{\mathcal{S}}$.

5 EXPERIMENTS

5.1 EXPERIMENT SETTINGS

Baselines. We compare nine UDF mesh extractors from the three families discussed in Sec. 2. MeshUDF (Guillard et al., 2022), CAP-UDF (Zhou et al., 2024), GeoUDF (Ren et al., 2023), HR-UDF (Stella et al., 2025), and MIND (Chen et al., 2026) infer grid-corner signs or labels and use Marching Cubes lookup tables to determine face connectivity within each cell. DualMesh-UDF (Zhang et al., 2023) and DCx (Bao et al., 2026) place one vertex in each cell containing the surface and connect vertices in neighboring cells to form faces. DCUDF (Hou et al., 2023) and DCUDF2 (Chen et al., 2025) establish connectivity by applying Marching Cubes to a positive UDF level set, then deform the resulting mesh toward the target surface.

Implementation Details. All experiments run on an AMD EPYC 7542 system with a single NVIDIA RTX A6000 GPU (48 GB). We optimize vertex positions with VectorAdam (Ling et al., 2022) and use a default extraction resolution of 256^3. We obtain each UDF using a state-of-the-art learning method (Yang et al., 2025), trained on 100k unoriented surface points per object or 1,000 points/m^2 for scenes. All extractors query the same UDF; for CAP-UDF and GeoUDF, we use only their extraction modules. DCUDF and DCUDF2 apply minimum-cut post-processing only in the manifold-object experiments. Before evaluation, we normalize each reference mesh to a unit box and apply the same transformation to its reconstruction.

Hyperparameters. All length parameters are specified in normalized neural-field coordinates. We initialize $\mathcal{D}$ using the corresponding DCUDF2 baseline configuration, without minimum-cut separation. We set $(R_{\text{init}}, R^0_{\text{gap}}, R^1_{\text{gap}}, R_G) = (0.01, 0.0075, 0.003, 0.0125)$, $\gamma = 0.5$, and $\delta_G = 0.015$. Adaptive remeshing uses target edge lengths in $[0.003, 0.01]$. The three refinement stages use $(200, 200, 120)$ iterations and a base learning rate of 5×10^{-4}. The UDF and edge-length weights in Stage I are $(1000, 1000)$; the QEF, curve-Laplacian, and anchoring weights in Stage II are $(1000, 1600, 100)$. In Stage III, both UDF terms have unit weight, and both Laplacian terms have weight 2000. These extraction and refinement parameters are shared across datasets.

Evaluation Metrics. For surface reconstruction we report Chamfer distance (CD), F1-score (threshold 0.004), and normal consistency (NC) over 100k area-uniform samples per mesh, plus the number of connected components (Comp). For non-manifold junctions, the predicted curves are the mesh edges incident to three or more faces and the ground-truth curves are extracted as described in Appendix A; both are sampled at 10k points, the prediction uniformly by arc length. We measure NM-CD, the sum of the two directional mean nearest-neighbor distances between the two sets, NM-F1 (threshold 0.004), and NM-LR, the median over models of the predicted-to-ground-truth junction length ratio (ideal value 1), which indicates junction redundancy. CD and NM-CD are scaled by 10^3, and F1, NC, and NM-F1 are percentages.

5.2 RESULTS AND COMPARISONS

Evaluation on Non-manifold Objects. We evaluate on 50 shapes from the non-manifold ABC dataset introduced by SALS (Ren et al., 2025), constructed by intersecting pairs of ABC shapes (Koch et al., 2019). VoroFacet achieves the best surface CD and F1, with a larger advantage in junction reconstruction (Table 1): NM-F1 exceeds the strongest baseline by 12.61 percentage points, NM-CD decreases from 45.81 to 37.74, and NM-LR is closest to one. These results indicate more accurate junction recovery with less redundant junction geometry. Fig. 7 shows that DCUDF and DCUDF2 bend separate manifold sheets around the junction, forming a gap rather than a shared intersection. The other baselines exhibit holes or spurious connections near junctions. VoroFacet instead joins incident sheets along shared junction curves while maintaining high geometric accuracy.

Table 1: Quantitative results on **Non-manifold Objects**. **Bold** / underline: best / second-best across all methods. —: no explicit NM edges on any model.

Method	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1	Comp	Time (s)
MeshUDF [ECCV'22]	5.727	78.97	97.12	69.58	26.26	0.111	241.9	9.124
CAP-UDF [TPAMI'24]	5.939	76.26	96.00	—	—	—	4157	174.7
GeoUDF [ICCV'23]	5.580	80.26	96.44	113.9	21.75	2.570	346.5	6.859
MIND [NeurIPS'26]	5.959	76.70	95.14	<u>45.81</u>	38.62	<u>1.883</u>	76.76	200.6
HR-UDF [NeurIPS'25]	5.580	<u>80.30</u>	97.17	—	—	—	5421	102.9
DualMesh-UDF [ICCV'23]	5.632	79.77	96.75	130.3	8.715	219.8	17.06	12.98
DCx [TOG'26]	5.645	79.02	96.92	46.44	<u>53.19</u>	3.575	8.52	75.59
DCUDF [TOG'23]	5.563	80.18	**98.07**	—	—	—	20.42	178.2
DCUDF2 [TVCG'25]	<u>5.562</u>	80.24	<u>98.05</u>	—	—	—	20.42	153.0
VoroFacet (Ours)	**5.549**	**80.46**	98.02	**37.74**	**65.80**	**1.619**	1.380	418.0

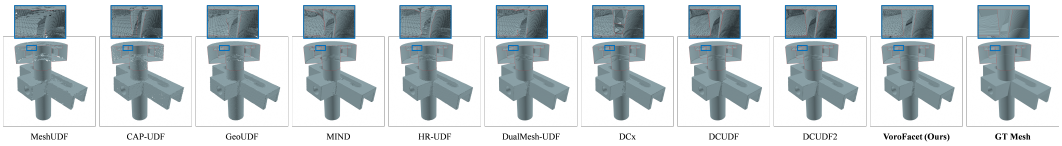

Figure 7: Visual comparison on a **non-manifold** object. Close-ups highlight junction connectivity: our VoroFacet joins incident sheets along shared junction curves, while methods under comparison leave holes, introduce spurious connections, or retain duplicated layers that leave gaps at junctions. *Zoom in for details.*

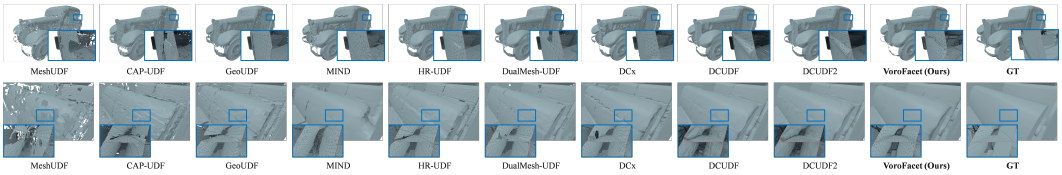

Figure 8: Visual comparison on a **ShapeNet-Car** model, showing its exterior (top) and interior (bottom). Close-ups show that VoroFacet keeps nearby but distinct sheets separate and joins incident sheets at junctions, whereas methods under comparison merge distinct sheets or fail to recover the correct junction connectivity. *Zoom in for details.*

Evaluation on Complex Interior Structures. ShapeNet-Car models (Chang et al., 2015) contain many thin sheets and non-manifold junctions in their interiors; we evaluate on 20 models, retaining all interior faces. VoroFacet keeps competitive surface accuracy while achieving the best NM-CD and NM-F1, with an NM-LR of 0.900 (Table 2). Its advantage in junction reconstruction therefore extends to these complex interior structures. Fig. 8 shows that VoroFacet resolves closely spaced sheets and their junctions, whereas the baselines merge distinct sheets or introduce holes, spurious connections, or duplicated layers.

Table 2: Quantitative results on **ShapeNet-Car**. **Bold** / underline: best / second-best across all methods. —: no explicit NM edges on any model.

Method	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1	Comp	Time (s)
MeshUDF [ECCV'22]	6.073	76.51	91.71	52.19	6.354	0.021	1154	12.60
CAP-UDF [TPAMI'24]	5.784	78.55	89.15	—	—	—	13342	267.5
GeoUDF [ICCV'23]	5.299	83.75	91.00	37.86	22.02	0.507	792.5	8.131
MIND [NeurIPS'26]	6.393	75.52	87.17	59.51	8.881	0.312	126.1	299.6
HR-UDF [NeurIPS'25]	5.309	**84.15**	90.75	—	—	—	3432	157.4
DualMesh-UDF [ICCV'23]	5.388	83.05	89.71	31.19	22.47	19.29	51.7	21.38
DCx [TOG'26]	5.605	81.30	88.64	25.47	30.80	1.751	52.75	361.7
DCUDF [TOG'23]	5.304	83.48	**92.27**	—	—	—	185.7	320.4
DCUDF2 [TVCG'25]	**5.291**	83.74	92.01	—	—	—	185.9	277.5
VoroFacet (Ours)	5.370	83.61	91.29	**21.79**	**36.67**	**0.900**	13.95	609.2

Evaluation on 3D Scenes. To assess generalization from individual objects to scenes, we evaluate on five real scans from the 3D Scene dataset (Zhou & Koltun, 2013). VoroFacet produces the fewest connected components while keeping competitive geometric accuracy, so each scene is recovered as a few coherent pieces rather than many fragments. We refer readers to Appendix C for the scene results.

Evaluation on Manifold Objects and Open Boundaries. We evaluate on 50 manifold shapes from FAMOUS (Erler et al., 2020) and Thingi32 (Maruani et al., 2023), and on 20 MGN models (Bhatnagar et al., 2019), whose garment surfaces have open boundaries. On both sets, VoroFacet is at or near the best surface accuracy and keeps the number of connected components close to one (Table 3). Manifold, open-boundary, and non-manifold structures are therefore all reconstructed within a single pipeline, which produces accurate single-layer meshes without minimum-cut post-processing. We refer readers to Appendix C for the open-boundary results.

Table 3: Quantitative results on **Manifold Objects**. **Bold** / underline: best / second-best.

Method	CD ↓	F1 ↑	NC ↑	Comp	Time (s)
MeshUDF [ECCV'22]	4.608	89.26	97.89	104.88	7.67
CAP-UDF [TPAMI'24]	4.615	89.22	97.63	2415.96	167.62
GeoUDF [ICCV'23]	4.518	**90.34**	97.69	65.4	6.39
MIND [NeurIPS'26]	4.572	89.90	97.81	6.78	121.17
HR-UDF [NeurIPS'25]	4.530	90.24	97.77	3190.04	78.22
DualMesh-UDF [ICCV'23]	4.569	89.57	97.72	1.86	9.45
DCx [TOG'26]	4.539	90.00	97.95	6.52	73.54
DCUDF [TOG'23]	4.529	90.08	**98.36**	1.08	303.04
DCUDF2 [TVCG'25]	4.528	90.15	98.32	1.06	216.11
VoroFacet (Ours)	**4.515**	90.33	98.24	1.28	306.59

5.3 Ablation Studies

VoroFacet constructs the output connectivity through two extraction passes and subsequently refines the geometry. We evaluate these designs on 20 non-manifold objects with four variants: replacing the adaptive ball cover with fixed minimum radii, and removing the junction-aware re-extraction, the geometry-adaptive remeshing, or the three-stage refinement.

The three-stage refinement matters most for the junction geometry. Without it (Fig. 6), the non-manifold vertices stay at the cluster centroids, so the junction curves are kinked, deviate from the intersection of the incident sheets, and fold the adjacent surfaces; with it, each junction is a single curve on that intersection, matching the ground truth. We refer readers to Appendix B for the other variants.

Table 4: Component ablations on 20 non-manifold objects.

Variant	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1	Time (s)
Fixed minimum radii	**5.608**	79.78	98.15	28.166	66.43	1.643	1824.6
w/o junction-aware re-extraction	5.644	79.17	98.05	25.886	**75.67**	1.930	372.3
w/o remeshing	5.631	79.71	97.96	24.639	72.72	1.517	397.9
w/o three-stage refinement	5.630	79.72	98.01	25.855	63.44	1.553	387.0
Full method	5.609	**79.80**	**98.16**	**23.331**	74.79	**1.486**	428.2

Table 4 is consistent with these observations. The adaptive ball cover primarily improves efficiency: with fixed minimum radii, flat regions are covered by many unnecessarily small balls, which increases the runtime by $4.3\times$ without improving accuracy. Removing the re-extraction markedly increases NM-LR, as nearby curves in $\mathcal{M}_0$ that represent the same junction are no longer merged by the junction graph and junction balls. Many of these extra points still fall within the matching threshold of the ground truth, which raises recall and gives this variant the highest NM-F1 in Table 4, even though its predicted junction length is nearly twice the ground truth. Removing the remeshing slightly degrades the overall accuracy, since the triangle density of $\mathcal{M}_1$ follows the inter-sheet spacing rather than the local geometry; its effect is more apparent around sharp features than in the averaged metrics. Removing the three-stage refinement causes the largest drop in NM-F1, whereas the surface metrics decrease only marginally. The extracted sheets already lie close to the target surface, while the non-manifold vertices stay inaccurate until Stage II. Appendix E breaks down the runtime of the full method, and Appendix D further analyzes the sensitivity to the main parameters and the extraction resolution.

6 Conclusion and Discussion

We have presented VoroFacet, a novel pipeline that extracts single-layer meshes from neural UDFs while handling complex structures including non-manifold junctions. VoroFacet treats a near-surface double-layered proxy of the target as a geometric prior and derives the output connectivity from the Voronoi diagram of the sites generated by an adaptive ball cover of that proxy, producing a single layer rather than duplicated geometry. Then a junction-aware re-extraction joins the sheets that meet at a junction, and remeshing and a three-stage refinement refine the geometry of the resulting mesh. Experiments on complex shapes show that VoroFacet recovers junctions that existing methods leave as holes, spurious connections, or gaps, while maintaining competitive surface accuracy. More broadly, these results suggest that a geometrically faithful proxy can guide surface extraction even when its topology is incorrect, and that the connectivity of the incident sheets can instead be constructed from a Voronoi partition.

Three limitations remain and suggest directions for future work. First, the ball radii are determined by the gaps of the double-layered proxy, i.e., the duplicate-layer gap on smooth sheets and the wider gap at junctions. If two thin sheets are separated by a distance comparable to or smaller than these gaps, the balls cover the space between them, so no sites are generated there and the two sheets are extracted as a single layer. A double cover that stays closer to the surface, or a more robust spatial partition in its place, would allow smaller ball radii and thus support thinner structures. Second, our method is not designed to preserve sharp features, since the Laplacian-based refinement smooths the normal discontinuity across a sharp edge. Such an edge could instead be constructed by feature balls placed along it, as junction balls do at non-manifold junctions, but this requires detecting the sharp edges from the UDF and preserving them during remeshing and refinement. Third, both extraction passes yield self-intersection-free meshes, but the three-stage refinement can introduce a few self-intersections. It remains open whether constrained or intersection-aware optimization can keep the output self-intersection-free without losing geometric accuracy.

AI USE STATEMENT

In preparing this manuscript, we used OpenAI's ChatGPT-5.6 solely to assist with improving the clarity and quality of the writing. Generative AI tools were not used to generate, modify, or preprocess experimental data. All AI-assisted text was critically reviewed, verified, and revised by the authors as appropriate. The authors take full responsibility for the final content of the manuscript, including its text, technical claims, and mathematical results.

ETHICS STATEMENT

This work does not involve human participants, personal or sensitive data, or automated decisions concerning individuals. We are not aware of ethical risks specific to the proposed method beyond those generally associated with mesh extraction and geometric processing.

REFERENCES

Ahmed Abdelkader, Chandrajit L Bajaj, Mohamed S Ebeida, Ahmed H Mahmoud, Scott A Mitchell, John D Owens, and Ahmad A Rushdi. Vorocrust: Voronoi meshing without clipping. *ACM Transactions on Graphics (TOG)*, 39(3):1–16, 2020.

Qingchao Bao, Xuhui Chen, Jingpeng Yin, Fei Hou, Wencheng Wang, Hong Qin, and Ying He. Dual contouring over expanded cubes (dcx) for zero-level set extraction from neural unsigned distance functions. *ACM Transactions on Graphics (TOG)*, 45(4):1–20, 2026.

Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5420–5430, 2019.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Xuhui Chen, Fugang Yu, Fei Hou, Wencheng Wang, Zhebin Zhang, and Ying He. Dcudf2: Improving efficiency and accuracy in extracting zero level sets from unsigned distance fields. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):9052–9065, 2025.

Xuhui Chen, Fei Hou, Wencheng Wang, Hong Qin, and Ying He. Mind: Material interface generation from udfs for non-manifold surface reconstruction. *Advances in Neural Information Processing Systems*, 38:120283–120306, 2026.

Zhiqin Chen, Andrea Tagliasacchi, Thomas Funkhouser, and Hao Zhang. Neural dual contouring. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020.

David Coeurjolly, Jaques-Olivier Lachaud, Konstantinos Katrioplas, Sébastien Loriot, Ivan Paden, Mael Rouxel-Labbé, Hossam Saeed, Jane Tournois, and Ilker O. Yaz. Polygon mesh processing. In *CGAL User and Reference Manual*. CGAL Editorial Board, 6.0.1 edition, 2024. URL https://doc.cgal.org/6.0.1/Manual/packages.html#PkgPolygonMeshProcessing.

Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *European conference on computer vision*, pp. 108–124. Springer, 2020.

Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. In *European conference on computer vision*, pp. 576–592. Springer, 2022.

Fei Hou, Xuhui Chen, Wencheng Wang, Hong Qin, and Ying He. Robust zero level-set extraction from unsigned distance fields based on double covering. *ACM Transactions on Graphics (TOG)*, 42(6):1–15, 2023.

Qijia Huang, Pierre Kraemer, and Dominique Bechmann. Offsetaxis: Udf mesh reconstruction via offset-volume medial axis extraction. *arXiv preprint arXiv:2605.15369*, 2026.

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 339–346, 2002.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *2019 IEEE/CVF conference on computer vision and pattern Recognition (CVPR)*, pp. 9593–9603. Ieee, 2019.

M Kohlbrenner and M Alexa. Meshing unsigned distance fields with regular triangulations. In *Computer Graphics Forum*, pp. e70524. Wiley Online Library, 2026.

Selena Zihan Ling, Nicholas Sharp, and Alec Jacobson. Vectoradam for rotation equivariant geometry optimization. *Advances in Neural Information Processing Systems*, 35:4111–4122, 2022.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Nissim Maruani, Roman Klokov, Maks Ovsjanikov, Pierre Alliez, and Mathieu Desbrun. Voromesh: Learning watertight surface meshes with voronoi diagrams. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14519–14528. IEEE, 2023.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4455–4465. IEEE, 2019.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 165–174. IEEE, 2019.

Marc Pouget and Frédéric Cazals. Estimation of local differential properties of point-sampled surfaces. In *CGAL User and Reference Manual*. CGAL Editorial Board, 6.0.1 edition, 2024. URL https://doc.cgal.org/6.0.1/Manual/packages.html#PkgJetFitting3.

Huan Ren, Wenfei Yang, Xiang Liu, Shifeng Zhang, and Tianzhu Zhang. Learning shape-independent transformation via spherical representations for category-level object pose estimation. In *International Conference on Learning Representations*, volume 2025, pp. 90178–90198, 2025.

Siyu Ren, Junhui Hou, Xiaodong Chen, Ying He, and Wenping Wang. Geoudf: Surface reconstruction from 3d point clouds via geometry-guided distance representation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14168–14178. IEEE, 2023.

Federico Stella, Nicolas Talabot, Hieu Le, and Pascal Fua. Neural surface detection for unsigned distance fields. In *European Conference on Computer Vision*, pp. 394–409. Springer, 2024.

Federico Stella, Nicolas Talabot, Hieu Le, and Pascal Fua. High resolution udf meshing via iterative networks. In *Advances in Neural Information Processing Systems*, pp. 170903–170933, 2025. doi: 10.52202/085713-5692.

Ningna Wang, Xiana Carrera, Christopher Batty, Oded Stein, and Silvia Sellán. Spudd: Superpower contouring of unsigned distance data. *arXiv preprint arXiv:2604.19568*, 2026a.

Ningna Wang, Zilong Wang, Xiana Carrera, Xiaohu Guo, and Silvia Sellán. Voroudf: Meshing unsigned distance fields with voronoi optimization. *arXiv preprint arXiv:2602.02907*, 2026b.

Chuanxiang Yang, Yuanfeng Zhou, Guangshun Wei, Long Ma, Junhui Hou, Yuan Liu, and Wenping Wang. Monge-ampere regularization for learning arbitrary shapes from point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6809–6822, 2025.

Congyi Zhang, Guying Lin, Lei Yang, Xin Li, Taku Komura, Scott Schaefer, John Keyser, and Wenping Wang. Surface extraction from neural unsigned distance fields. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22474–22483. IEEE, 2023.

Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Cap-udf: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7475–7492, 2024.

Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (ToG)*, 32(4):1–8, 2013.

A Ground-Truth Junction Extraction from CAD Meshes

A junction curve in a CAD mesh appears in one of three ways. The first is a mesh edge shared by three or more triangles, which the connectivity already records. The second is the segment along which two triangles cross. The third is a mesh edge that stays very close to one or more sheets without intersecting them and runs nearly parallel to them, for instance when it projects into the interior of a sheet rather than onto its boundary, or when it runs parallel to the boundaries of two or more sheets. This last case is not marked as a junction in either the connectivity or the intersections, and it is the one we mainly need to extract.

We sample the first two directly. The remaining mesh edges are the candidates for the third case, and we sample each of them and judge the sample points one by one. A sample point lies on a junction curve when at least three surface directions meet there, whereas a smooth sheet contributes only two. We count these directions from the triangles around the point. We collect the candidate triangles, namely the triangles whose distance to the point is below a tolerance ε and whose plane is within 3° of parallel to the edge it lies on. A candidate triangle extends on both sides of the edge when the edge runs through its interior, and on one side only when the triangle ends at one of its own edges, which we detect by testing whether the triangle has an edge nearly parallel to the sample edge. Two directions count as distinct when the angle between them is at least 3°, and we retain the point if at least three distinct directions remain. The retained points form the ground-truth junction curves against which we compute NM-CD and NM-F1.

B Additional Qualitative Results

Figures 9 and 10 show the visual effect of the two components. Without the junction-aware re-extraction, a junction is split into two nearby curves, and the faces between them are distorted; with it, the incident sheets meet along a single shared curve. Without the geometry-adaptive remeshing, the triangles around a sharp feature are coarse and irregular, and the fitted surface is jagged; with it, the sampling follows the local geometry and the feature is recovered. Both effects are local, so they are only partly reflected in the averaged metrics of Table 4, where removing the re-extraction mainly raises NM-LR from 1.486 to 1.930.

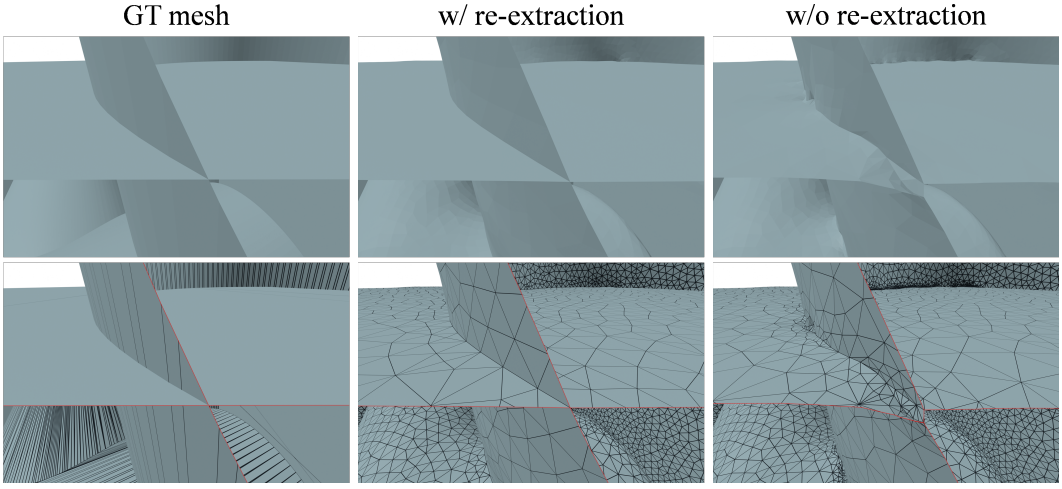

Figure 9: Visual effect of junction-aware re-extraction.

C Additional Quantitative Results

This section completes the comparisons of Sec. 5.2. Table 5 reports the 3D scenes and Table 6 the open-boundary objects.

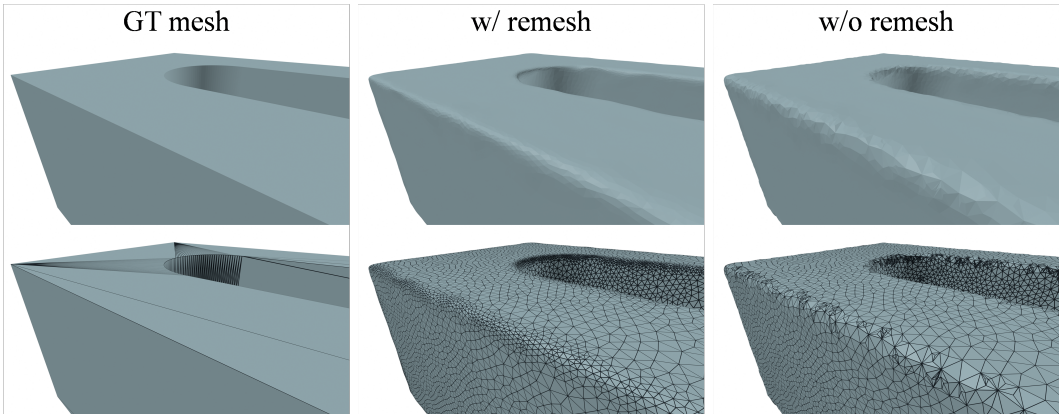

Figure 10: Visual effect of geometry-adaptive remeshing.

Table 5: Quantitative results on **3D Scenes**. **Bold** / underline: best / second-best. All values are averaged over five scenes.

Method	CD ↓	F1 ↑	NC ↑	Comp	Time (s)
MeshUDF [ECCV'22]	3.866	92.88	**94.32**	455	9.4
CAP-UDF [TPAMI'24]	3.638	95.87	91.91	4316	151.1
GeoUDF [ICCV'23]	3.547	96.44	93.20	450.6	9.2
MIND [NeurIPS'26]	3.848	94.14	92.72	19.6	103.4
HR-UDF [NeurIPS'25]	3.575	96.72	92.84	1963	87.4
DualMesh-UDF [ICCV'23]	3.612	96.30	92.05	9.2	11.0
DCx [TOG'26]	3.835	94.01	92.97	56.8	35.7
DCUDF [TOG'23]	**3.510**	96.66	94.18	24.2	102.0
DCUDF2 [TVCG'25]	3.524	**96.76**	94.09	23.4	91.0
VoroFacet (Ours)	3.664	95.86	93.40	6.4	199.9

D Sensitivity Analysis

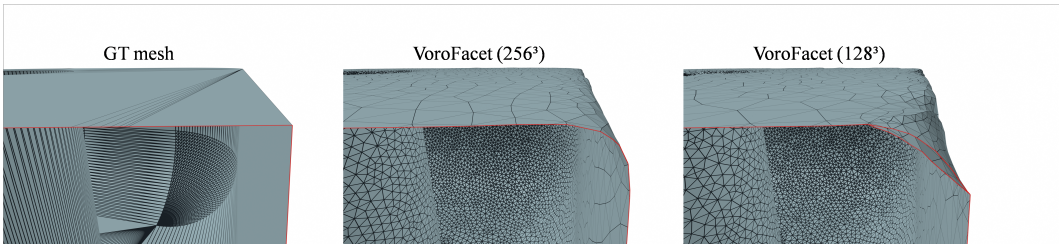

Figure 11: Visual effect of the extraction resolution near a sharp feature.

Tables 7 to 11 vary one parameter at a time on the same 20 non-manifold objects used in Sec. 5.3. VoroFacet is stable across the tested ranges except at 128^3, where the proxy follows the surface less closely and the wider duplicate-layer gap admits spurious sites (Fig. 11), and at the two smallest gap thresholds, which fail in the same way. These settings reduce NM-F1 from 74.79 to 20.44, 71.75, and 70.27, respectively. The remaining parameters change NM-F1 by less than 1.5 points, and 512^3 improves it only slightly, so we keep 256^3 and the default values listed in Sec. 5.1.

Table 6: Quantitative results on **Open-boundary Objects**. **Bold** / underline: best / second-best.

Method	CD ↓	F1 ↑	NC ↑	Comp	Time (s)
MeshUDF [ECCV'22]	2.830	99.22	98.93	12.65	5.46
CAP-UDF [TPAMI'24]	2.943	98.53	98.47	1056.35	155.75
GeoUDF [ICCV'23]	2.889	99.03	98.58	28.9	5.76
MIND [NeurIPS'26]	2.942	98.58	98.66	2.05	47.64
HR-UDF [NeurIPS'25]	2.938	98.86	98.68	991.65	64.40
DualMesh-UDF [ICCV'23]	2.860	99.29	98.73	1.3	5.54
DCx [TOG'26]	2.869	98.94	98.69	4.1	16.51
DCUDF [TOG'23]	**2.800**	**99.48**	**99.13**	3.3	122.21
DCUDF2 [TVCG'25]	2.809	99.43	99.09	3.3	101.70
VoroFacet (Ours)	2.843	99.19	98.93	1.05	175.04

Table 7: Effect of extraction resolution on 20 non-manifold objects.

Resolution	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1
128^3	5.722	78.68	96.93	60.303	20.44	5.591
256^3 (default)	5.609	79.80	98.16	**23.331**	74.79	1.486
512^3	**5.602**	**79.86**	**98.21**	25.356	**78.28**	**1.397**

E Runtime Efficiency

VoroFacet takes 418 seconds per non-manifold object on average (Table 1). Constructing $\mathcal{D}$ with DCUDF2 accounts for 41.2% of the runtime; the two passes of ball covering and surface extraction, including junction estimation, account for 33.9%, remeshing for 14.5%, and three-stage refinement for 8.8%. The remaining 1.6% is spent on final checks and other overhead. Proxy construction and the two extraction passes together account for approximately 75% of the runtime, making them the main targets for acceleration.

Table 8: Effect of the first-pass gap threshold R^0_{gap}.

R^0_{gap}	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1
0.006	**5.608**	**79.83**	98.15	26.181	71.75	1.557
0.0075 (default)	5.609	79.80	**98.16**	**23.331**	**74.79**	**1.486**
0.009	5.613	79.81	**98.16**	24.001	74.58	1.541

Table 9: Effect of the second-pass gap threshold R^1_{gap}.

R^1_{gap}	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1
0.002	5.611	79.77	98.14	29.053	70.27	1.757
0.003 (default)	**5.609**	**79.80**	98.16	23.331	74.79	1.486
0.004	**5.609**	79.79	**98.17**	**21.836**	**75.74**	**1.403**

Table 10: Effect of the junction-ball radius R_G.

R_G	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1
0.01	**5.609**	79.81	98.13	23.596	74.52	1.622
0.0125 (default)	**5.609**	79.80	98.16	**23.331**	**74.79**	1.486
0.015	5.610	**79.82**	**98.18**	23.742	74.64	**1.422**

Table 11: Effect of the junction clustering spacing δ_G.

δ_G	CD ↓	F1 ↑	NC ↑	NM-CD ↓	NM-F1 ↑	NM-LR → 1
0.01	**5.608**	**79.83**	**98.17**	23.307	**75.32**	1.490
0.015 (default)	5.609	79.80	98.16	23.331	74.79	**1.486**
0.02	5.613	79.76	98.16	**23.016**	73.85	1.543