PATCHWORK: A POLYHEDRAL IMPLICIT REPRESENTATION FOR HIGHLY DETAILED 3D SHAPES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce patchwork, a highly expressive shape representation that explicitly encodes 3D shapes as unions of polyhedra while admitting a simple closed-form zero-level-set equation. It is grounded in a rigorous mathematical framework that yields provable complexity bounds and guarantees approximation of arbitrary shapes to arbitrary precision. To fit patchwork to a given shape, we pair conventional gradient-based optimization with a novel combinatorial matching scheme. Experiments show that our approach achieves superior quality with 3–6× fewer parameters compared to existing alternatives on average, and scales to highly detailed artist-created assets. Patchwork further supports explicit polygonalization and conforming volume mesh extraction, making it a versatile, scalable, yet compact representation for downstream tasks, with potential for geometric learning.



Figure 1: A collection of high-quality shapes (Zhang et al., 2025) expressed in our representation. The left 8 cases (explicit polygonalization) average under 4MB of parameters, while the right (Marching Cubes at 1536^3 resolution) uses 40MB to represent the mesh over 800 MB. Insets show normals and polyhedra (randomly colored) visualized via ray marching over our implicit form.

1 INTRODUCTION

Recent advances in modern graphics center around the revolution in 3D representations. Neural implicit representations encode shapes as continuous functions (Park et al., 2019), circumventing the cubic memory complexity of discrete voxel grids and achieving unrivaled representational expressiveness (Sitzmann et al., 2020); volume density fields (Mildenhall et al., 2020) and 3D Gaussian representations (Kerbl et al., 2023) enable novel-view synthesis from images, bypassing the conventional modeling–texturing–rendering pipeline. Recently, foam-based 3D representations have attracted considerable attention for their competitive ray-tracing performance (Govindarajan et al., 2025; 2026). Under such a representation, the space is partitioned into convex polyhedral cells, typically via Voronoi or Power tessellation, with each cell associated with additional attributes such as sign or volume density, and the surface geometry is then encoded as a subset of the cell interfaces.

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.

Although loosely characterized as a hybrid implicit–explicit representation (Rech et al., 2025), it lacks a precise implicit formulation and a theoretical analysis of its expressive power. Moreover, despite its geometric learning and meshing potential (Maruani et al., 2023), fitting a foam-based representation to an existing geometry remains challenging: current fitting methods yield coarse, low-resolution results (Maruani et al., 2023; Charawi et al., 2026), with demonstrated expressiveness significantly outclassed by neural implicit variants.

In this work, we revisit *Viro's patchworking* idea from pure mathematics (Itenberg & Viro, 1996) to derive a formal yet simple expression for the implicit foam-based surface representation, which we term *patchwork* in acknowledgment of its origin. In patchworks, we unify the convex cells as power cells, encode them algebraically as linear functions, each signed and composed via a global maximum, resulting in a closed-form zero level set equation. We derive its complexity bounds and prove its ability to approximate arbitrary shapes with arbitrary precision.

To realize patchworks's expressiveness in practice, we develop a novel gradient-based optimization scheme: In contrast to the conventional practice of replacing the maximum with a soft variant (e.g., LogSumExp (Deng et al., 2020)) to distribute gradient signals over multiple cells, we keep the hard maximum and match the best-fit cells with respect to input constraints to ensure clean gradient updates. Our scheme is simple to implement, retains high fitting throughput, and is compatible with deep learning optimizers such as Adam. We evaluate our method on multiple datasets of varying shape complexity, showing that patchwork has ample capacity to encode detailed geometry, surpassing highly expressive neural implicit counterparts such as SIREN (Sitzmann et al., 2020), while requiring 3–6$\times$ less raw floating-point storage on disk on average. We further demonstrate patchwork's scaling capacity across a set of complex structures, as well as practical applications unique to our representation.

Our contributions are summarized as follows:

- We formalize the implicit foam-based surface representation in closed form. Our representation, patchwork, bridges neural networks and known mathematical theories, and is in theory capable of approximating arbitrary shapes with arbitrary precision.
- We propose a novel optimization scheme that makes fitting our representation viable with gradient descent, yielding results superior to baselines while requiring, on average, 3–6$\times$ fewer floating-point parameters.

2 Related work

In this section, we focus on reviewing the literature from the machine learning side and leave the mathematical origins to the Appendix A.

Neural shape representations. The representation of 3D shapes using neural networks has been explored through various approaches (Xie et al., 2022). Early methods focus on designing losses to regularize the distance field, notably the sign-agnostic variants (Atzmon & Lipman, 2020b;a; Lipman, 2021) and the Eikonal constraint (Gropp et al., 2020). Others improve the network's representational power with architectural modifications, such as the sine activation function (Sitzmann et al., 2020) and positional encoding (Mildenhall et al., 2020; Tancik et al., 2020). Another line of research relocates the expressiveness of neural networks into auxiliary parameter-heavy spatial data structures. NGLOD (Takikawa et al., 2021) organizes features in a sparse octree, providing adaptive levels of detail, whereas InstantNGP (Müller et al., 2022) uses a multiresolution hash table and achieves extremely fast training speed. More relevant to ours are methods that represent shape as a composition of primitives. Tulsiani et al. (2017) assembles cuboids, CvxNet (Deng et al., 2020) composes shapes as convex cells, and BSP-Net (Chen et al., 2020) performs recursive binary space partitioning. Although we also use convex cells, our decomposition is conforming, and we partition the full space using power diagrams (Aurenhammer, 1987) without recursion.

Foam-based fitting methods. VoroMesh (Maruani et al., 2023) pioneers the gradient-based fitting of explicit Voronoi surface meshes, but its parameter count scales cubically with grid resolution and its reconstructed surfaces exhibit large seams. DCCVT (Charawi et al., 2026) initializes from a pre-fitted implicit signed distance function (SDF) and clips the Voronoi cells against the jointly optimized zero-level set, reducing those artifacts. RadiantFoam (Govindarajan et al., 2025)

Table 1: Feature table of foam-based methods.

	Clipping-free	No aux. field	Explicit polygons	Scalable
VoroMesh (Maruani et al., 2023)	✓	✓	✓	✗
DCCVT (Charawi et al., 2026)	✗	✗	✓	✗
RadiantFoam (Govindarajan et al., 2025)	✓	✓	✗	✓
PowerFoam (Govindarajan et al., 2026)	✓	✓	✗	✓
SDFoam (Rech et al., 2025)	✓	✗	✓	✓
Ours	✓	✓	✓	✓

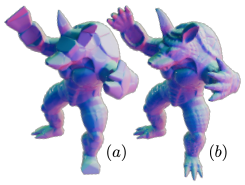


Figure 2: Fitting result (a) without and (b) with our discrete matching.

tessellates the space with Voronoi cells and defines per-cell density for volume rendering. PowerFoam (Govindarajan et al., 2026) generalizes it with power cells and bisects each cell to capture macro-scale geometry. Neither of them supports explicit polygonalization and requires Truncated-SDF fusion (Newcombe et al., 2011) for surface extraction. SDFoam (Rech et al., 2025) fits a guiding implicit SDF along Voronoi sites and extracts the surface as the boundary of the union of cells whose centers have negative signed distance. In contrast, our method yields a power diagram that decomposes the full space, without clipping or relying on a guiding field, and supports explicit polygonalization. Tab. 1 summarizes the full feature comparison. To the best of our knowledge, PowerCrust (Amenta et al., 2001) is the only classical method that, like ours, extracts the surface directly from the faces of a power diagram (rather than dual triangulation, restricted or bounded variants), but it does not offer an implicit representation and requires approximating the medial axis, a computation that is notoriously difficult to make robust and scale up (Wang et al., 2024; 2025).

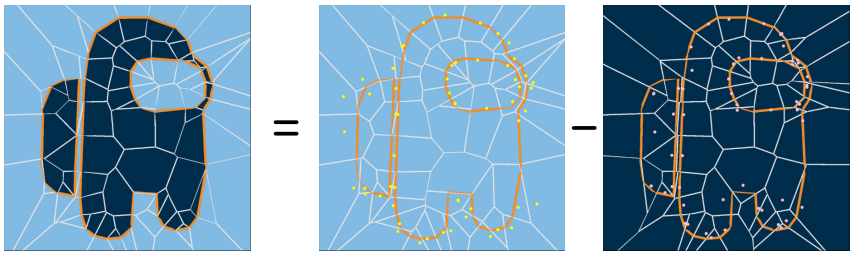


Figure 3: Illustration of a 2D patchwork: subtracting the piecewise linear functions associated with the dark (negative) power diagram from the light (positive) one yields the patchwork (orange curve).

3 METHOD

In this section, we introduce our fitting method. Given a target surface in a representation that lets us sample point-normal pairs (x_k, n_k), we aim to fit a field $f(x)$ whose zero-level set describes the surface, so that $f(x_k) = 0$ and $\nabla f(x_k) \parallel n_k$. We employ fields of the following special form.

Definition 1. *A patchwork field is a function $\mathbb{R}^3 \to \mathbb{R}$ of the form*

$$f(x) = \max_{1 \le i \le N^+} \underbrace{\{\langle a_i^+, x\rangle + c_i^+\}}_{l_i^+(x)} - \max_{1 \le j \le N^-} \underbrace{\{\langle a_j^-, x\rangle + c_j^-\}}_{l_j^-(x)}, \tag{1}$$

with $N^+, N^- \in \mathbb{N}$, parameters $a_i^+, a_j^- \in \mathbb{R}^3$, $c_i^+, c_j^- \in \mathbb{R}$. A patchwork is the zero level set $f(x) = 0$.

For each $i = 1, \ldots, N^+$, the locus of points $x \in \mathbb{R}^3$ where the first maximum in Eq. (1) is attained at this particular i, i.e. $l_i^+(x) = \max_k l_k^+(x)$, is called the i-th positive cell (Fig. 3 middle). Analogously, we define j-th negative cell (Fig. 3 right). The patchwork field is a linear function in the intersection of a positive and a negative cell, hence is a piecewise linear function. We also define a patchwork cell as the set of all points x where a selected linear function $l_i^+(x)$ or $l_j^-(x)$ is greater or equal than all the other ones, both $l_k^+(x)$ and $l_k^-(x)$ (Fig. 3 left). A step-by-step introduction of these core concepts, theoretical expressiveness, and complexity analysis are collected in Appendices B –E. (Even when essentially known, those results are difficult to extract from existing literature).

Optimization Challenges. A standard approach to similar fitting problems is to supervise or regularize with off-surface samples. However, patchwork is not a distance field, making pre-acquiring ground-truth values or applying the Eikonal regularization (Gropp et al., 2020) infeasible. Next, its

piecewise nature allows field values at two query points x_j, x_k to come from the same pair of linear functions (when x_j, x_k lie in the same positive and negative cells). These query points, therefore, share the same gradient, leaving no capacity to model two possibly conflicting directional constraints n_j, n_k. (Fig. 4 (a-b)). Replacing the hard maximum with a soft variant would further blend incompatible constraints into neighboring cells and degrade the results (Fig. 2 (a)). To this end, departing from the conventional practice of making everything smooth, we deliberately keep the maximum assignment *as-discrete-as-possible*.

Algorithm Overview. We initialize the patchwork field using the power diagram (**Geometric Initialization**). Then we iteratively refine it using the following five steps. First, we sample a set of K point-normal pairs (x_k, n_k) viewed as constraints. Second, we select a matching pair of cells (one positive, one negative) for each of the constraints (x_k, n_k) from four candidates (**Discrete Matching**). Third, we stabilize the assignment to limit oscillations in the discrete matching (**Assignment Stabilization**). Fourth, we fit the surface so that all constraints (x_k, n_k) are covered by their matched pair of cells (**Cell Fitting**). Fifth, we identify and remove redundant linear functions (**Pruning**). This step includes **Normal Inversion Handling**.

Geometric Initialization. Eq. (1) can be rewritten by negating the linear functions and completing the squares as follows:

$$f(x) = \min_{1\le i\le N^-}\{|x-p_i^-|^2 - w_i^-\} - \min_{1\le j\le N^+}\{|x-p_j^+|^2 - w_j^+\}, \tag{2}$$

where $p_i^+ = \frac{1}{2}a_i^+, p_j^- = \frac{1}{2}a_j^- \in \mathbb{R}^3$ and $w_i^+ = \frac{1}{4}|a_i^+|^2 + c_i^+, w_j^- = \frac{1}{4}|a_j^-|^2 + c_j^- \in \mathbb{R}$. Thus, our positive and negative cells correspond to the cells of two power diagrams (Fig. 3) with sites $\{p_i^+\}_{i=1}^{N^+}$ and $\{p_j^-\}_{j=1}^{N^-}$. Inspired by VoroCrust (Abdelkader et al., 2020), given a set of samples $\{(x_i, n_i)\}_{i=1}^{N}$, we place the zero-weighted sites to double-cover the surface, that is, set

$$N^\pm = N, \qquad a_i^\pm = 2(x_i \pm \delta\cdot n_i), \qquad c_i^\pm = -\frac{1}{4}|a_i^\pm|^2, \tag{3}$$

where $\delta = \frac{0.01}{\tau}$ is a small displacement for a shape normalized to $[-1,1]^3$, and $\tau \in \mathbb{R}$ is a scaling hyperparameter based on the geometric feature scale.

Discrete Matching. For each query (x_k, n_k), we choose a matching pair of cells such that the difference of corresponding linear functions is best aligned with the normal n_k: one from the positive cell containing x_k and its neighbors, and one from the negative cell containing x_k and its neighbors. Since it is difficult to reconstruct the cell neighborhood structure, we only choose among the closest neighboring cells, i.e., the "second-maximum" ones, that are guaranteed to be adjacent if nonempty (Fig. 4(c-d)). Namely, for each fixed k, take the values of i for which $l_i^+(x_k) = \langle a_i^+, x_k\rangle + c_i$ is maximal and second maximal. These indices are denoted $\dot{k}^+$ and $\ddot{k}^+$. Analogously, define $\dot{k}^-$ and $\ddot{k}^-$. We choose

$$(k^+, k^-) = \underset{i\in\{\dot{k}^+,\ddot{k}^+\}, j\in\{\dot{k}^-,\ddot{k}^-\}}{\arg\max} \{\mathrm{sim}(a_i^+ - a_j^-, n_k)\}, \tag{4}$$

where sim is the cosine similarity. The k^+th positive and k^-th negative cell form the matching pair.

Assignment Stabilization. Empirically, we observe that two linear functions can overlap during the fitting (see nearly coincident sites in Fig. 3), so that a small gradient oscillation can alter the matching assignment, causing instability. Therefore, we set $k^\pm := \dot{k}^\pm$ when the angle difference between $a_{k^+}^+ - a_{k^-}^-$ and $a_{\dot{k}^+}^+ - a_{\dot{k}^-}^-$ is less than 5°.

Cell Fitting. We introduce the loss functions and update parameters $a_{k^\pm}^\pm$ and $c_{k^\pm}^\pm$ as if the maxima in (1) were attained at the indices $k^\pm$:

$$\mathcal{L}_{\text{sur}} = \frac{1}{K}\sum_{k=1}^{K}|l_{k^+}^+(x_k) - l_{k^-}^-(x_k)|, \qquad \mathcal{L}_{\text{normal}} = \frac{1}{K}\sum_{k=1}^{K}(1-\mathrm{sim}(a_{k^+}^+ - a_{k^-}^-, n_k)). \tag{5}$$

We then encourage the matched pair of cells to cover x_k and become the true maxima using the loss

$$\mathcal{L}_{\text{cover}} = \frac{1}{K}\sum_{k=1}^{K}\left(\text{ReLU}\left(l_{\dot{k}^+}^+(x_k) - l_{k^+}^+(x_k)\right) + \text{ReLU}\left(l_{\dot{k}^-}^-(x_k) - l_{k^-}^-(x_k)\right)\right). \tag{6}$$

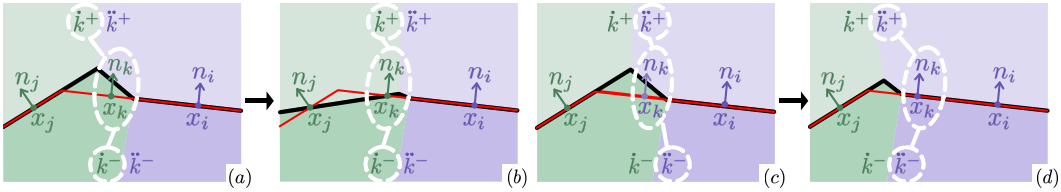


Figure 4: Discrete matching, where the red line indicates the target, the black line indicates the current zero level set, the light/dark green and light/dark purple regions refer to $\dot{k}^{\pm}$th and $\ddot{k}^{\pm}$th cells, respectively. (a) In the standard approach, the constraint (x_k, n_k) is assigned to $\dot{k}^+$th positive and $\dot{k}^-$th negative cells (the ones that contain x_k). (b) This rotates the zero isosurface normal to match both n_j and n_k, causing it to deviate from the target. (c) In contrast, our method matches the constraint (x_k, n_k) to $\ddot{k}^+$th positive and $\ddot{k}^-$th negative cells, (d) reducing the approximation error while maintaining the correct zero isosurface normal direction.

Pruning. During optimization, some cells may become empty or drift away from the surface as floaters. In either case, they no longer contribute to the active surface patches and can be removed freely. We pair each linear function with an auxiliary variable $s_i^{\pm} \in (0, 1]$ measuring importance and remove those whose $s_i^{\pm}$ falls below 0.5 during optimization. We initialize all importance to be 1 and update $s_i^{\pm}$ via the straight-through gradient estimator (Bengio, 2013) using the loss

$$\mathcal{L}_{\text{prune}} = \frac{1}{K}\sum_{k=1}^{K}\left(1 - s_{k^+}^{+} + S_k^{-} + s_{\widetilde{k}^+}^{+} + s_{\widetilde{k}^-}^{-}\right), \quad S_k^{-} = \begin{cases} 1 - s_{k^-}^{-}, & \text{if } \operatorname{sim}(a_{k^+}^{+} - a_{k^-}^{-}, n_k) \geq 0, \\ s_{k^-}^{-}, & \text{if } \operatorname{sim}(a_{k^+}^{+} - a_{k^-}^{-}, n_k) < 0, \end{cases} \tag{7}$$

where $\widetilde{k}^{\pm}$ denotes the element of the set $\{\dot{k}^{\pm}, \ddot{k}^{\pm}\}$ distinct from $k^{\pm}$. After the optimization, we only keep linear functions with $s_i^{\pm} > 0.5$.

Normal Inversion Handling. In initializing a given shape, we offset all surface samples by the same δ, regardless of local feature size within the shape. For thorny features, this can cause negative cells to overshoot the surface and extend outwards unbounded. Although this can be penalized with $\mathcal{L}_{\text{normal}}$, the cosine similarity would rotate two opposing cells by 180°, causing large structural changes and destabilizing the fitting process. This is the reason for the second case in Eq. (7).

Implementation details For each shape, we normalize its bounding box to $[-1, 1]^3$ and uniformly sample N point-normal pairs for initialization, with N depending on the experiment setup. During fitting, we additionally sample K point-normal pairs per iteration, and minimize the total loss

$$\mathcal{L}_{\text{total}} = 10 \cdot \mathcal{L}_{\text{sur}} + \mathcal{L}_{\text{normal}} + 100 \cdot \mathcal{L}_{\text{cover}} + \mathcal{L}_{\text{prune}}$$

using Adam (Kingma & Ba, 2014) over 10000 iterations. We set the learning rate to $3 \times 10^{-4}/\tau$, with step decay of factor 0.25 for every 40% of iterations. We implement the training code in PyTorch (Paszke et al., 2019) and implement the global scan of maximum/second-maximum as a fused kernel (Dao et al., 2022; Zhang & Wonka, 2025) using Triton (Tillet et al., 2019). We also implement a variant as a 4D KD-tree using cudaKDTree (Wald, 2023). We run all experiments on a single RTX3090 GPU (24 GB VRAM) and report performance in Tab. 5. We extract our meshes via explicit polygonalization (See Appendix H for details). We will release our full implementation and model checkpoints after the anonymous review to ensure reproducibility.

4 Experiments

4.1 Surface Fitting

We assess the expressiveness of patchwork on the surface fitting task over two shape sets spanning three well-established datasets, with increasing model capacity to match the shape complexity.

Datasets ABC and Thingi10k (Koch et al., 2019; Zhou & Jacobson, 2016); the former is a large collection of CAD models, while the latter consists of 10K more general shapes. We pick 100 models from each, following (Erler et al., 2020; Huang et al., 2023), as our first shape set. ShapeNet (Chang et al., 2015) is a large-scale repository of CAD shapes. Following Neural Splines (Williams et al., 2021), we use the same 260-shape subset as our second shape set, consisting of 20 shapes per

Table 2: Quantitative results on ABC / Thingi10K. The bold text indicates the best score; the underlined text indicates the second-best. 'Ours #Params' is the average parameter size after pruning.

Method	CH ↓	HD ↓	FS ↑	NC ↑	IoU ↑	#Params
MC	4.43 / 3.55	5.77 / 6.72	47.86 / 43.82	92.96 / 94.99	89.97 / 93.88	132651
RFTA	3.73 / 3.31	5.67 / 5.41	53.10 / 52.88	94.47 / 96.10	91.55 / 94.68	132651
SPSR	2.27 / 1.88	**3.31 / 4.06**	60.43 / 65.71	96.87 / 97.82	95.36 / 97.59	131070
PoNQ	2.41 / 2.14	4.38 / 4.25	63.91 / 60.22	96.70 / 96.90	95.89 / 97.53	131072
IGR	3.07 / 1.94	5.17 / 4.08	64.19 / 58.60	97.75 / **98.95**	95.16 / 98.06	132094
SIREN	2.79 / 3.09	21.7 / 31.4	57.50 / 61.81	**98.45** / 98.73	95.61 / 97.94	132865
VoroMesh	2.94 / 2.21	4.92 / 5.12	64.39 / 59.55	96.65 / 95.77	96.21 / 97.37	128625
Ours init	5.68 / 5.48	20.8 / 27.6	58.20 / 64.10	87.31 / 90.53	92.64 / 95.49	131072
Ours	**1.90 / 1.80**	3.81 / 6.78	**73.22 / 77.27**	98.05 / 98.17	**98.39 / 99.14**	45192

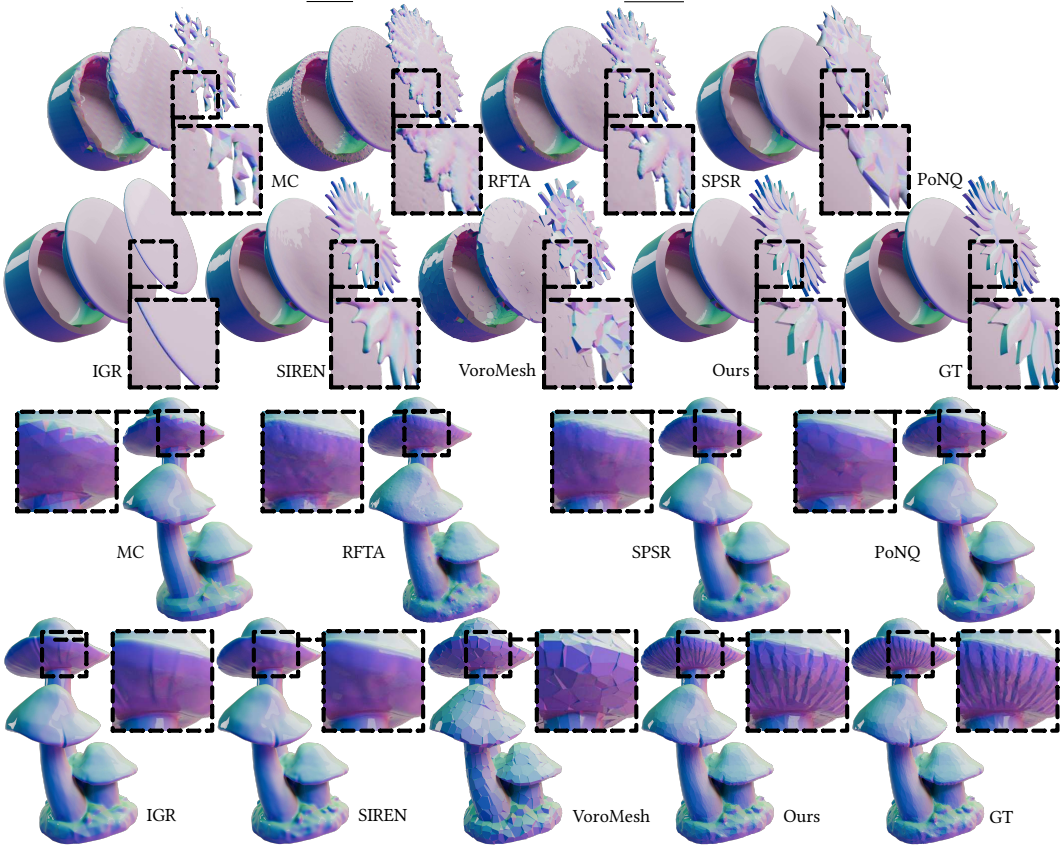


Figure 5: Visual results on ABC (Koch et al., 2019) and Thingi10k (Zhou & Jacobson, 2016).

category across 13 categories. We follow CLAY (Zhang et al., 2024) and Dora (Chen et al., 2025) to preprocess each shape to be manifold (Wei et al., 2025).

Model capacity. We initialize patchwork with $N = 16384$ and 65536 surface samples across two shape sets, set $K = 16384$ for all and τ to 1.0 and 2.0, respectively.

Baselines We compare against VoroMesh (Maruani et al., 2023), the foam-based representation, and three common topology-agnostic families: regular grids, evaluated via MC (Lorensen & Cline, 1987) and RFTA (Sellán et al., 2024); point sets, evaluated via SPSR (Kazhdan & Hoppe, 2013) and PoNQ (Maruani et al., 2024); and neural implicit, via IGR (Gropp et al., 2020) and SIREN (Sitzmann et al., 2020). Since our final parameter count varies across shapes due to pruning, we match all baselines to the parameter count at our initialization (i.e., $8N$, excluding auxiliary $s^{\pm}$), the upper bound of our final raw floating-point storage size on disk. For fitting-based methods (PoNQ, IGR, SIREN, VoroMesh), we resample $K = 16384$ surface samples each iteration to match our protocol; exact configurations and fitting details are given in the Appendix G.2. We additionally report our final parameter counts after pruning, averaged within each shape set.

Table 3: Quantitative results on the ShapeNet dataset. The bold text indicates the best score; the underlined text indicates the second-best. 'Ours #Params' is the average parameter size after pruning.

Method	CH ↓	HD ↓	FS ↑	NC ↑	IoU ↑	#Params
MC	29.2	26.7	17.63	87.36	15.39	531441
RFTA	16.7	17.3	26.45	88.67	27.24	531441
SPSR	3.09	**3.10**	33.75	86.89	54.64	524286
PoNQ	3.48	9.13	42.43	85.08	58.10	524288
IGR	4.70	11.1	<u>54.85</u>	94.69	<u>70.99</u>	526334
SIREN	<u>2.66</u>	8.37	49.82	**97.89**	66.08	527873
VoroMesh	63.3	45.6	8.03	62.43	10.41	526848
Ours init	6.56	24.4	30.44	68.00	47.80	524288
Ours	**1.65**	<u>3.28</u>	**64.26**	<u>97.77</u>	**96.69**	90808

Metrics For quantitative evaluation, we normalize all results in range $[-1, 1]$, sample 1M points on both ground truth and raw output surfaces and report Chamfer Distance (CH, $\times 10^{-3}$) (Fan et al., 2017), Hausdorff Distance (HD, $\times 10^{-2}$), F-score (FS) (Knapitsch et al., 2017) with a cutoff 0.1% largest dimension of the bounding box, Normal Consistency (NC, $\times 10^{-2}$) and Intersection-over-Union (IoU, $\times 10^{-2}$) (Peng et al., 2020). For IGR and SIREN, we use MC of resolution 512^3 to extract the underlying surfaces.

Quantitative Results Patchwork achieves the best Chamfer Distance, F-Score, and IoU across all datasets. Our F-score consistently exceeds the second-best method by roughly 10 percentage points, demonstrating both higher accuracy and completeness (Knapitsch et al., 2017). Moreover, it does so with only 34% and 17% of the matched parameter budget after pruning.

Qualitative Results Compared to VoroMesh, the foam-based surface representation, patchwork substantially outperforms it on smoother, more complex shapes, with the gap widening as shape complexity increases. Compared to IGR, patchwork represents intricate geometric detail more faithfully; compared to SIREN, our hard maximum preserves sharper creases. More visual results are available in Appendix G.1.

4.2 Scalability

To examine scalability, we compare against GeomDist (Zhang et al., 2025), which encodes shapes as distributions via a diffusion model and achieves SOTA geometric fidelity, surpassing even hash-grid-based methods (Müller et al., 2022). Our method faithfully captures outer surfaces for all cases and offers explicit polygonalization, achieving higher accuracy on 7 of 10 shapes (mean Chamfer Distance of 2.380 $\times 10^{-3}$ vs. 3.702 $\times 10^{-3}$), but falls short on the remaining 3 in handling self-intersection and non-manifold interior parts, resulting in spurious geometry enclosed within the outer surface. Our results are visualized in Fig. 1; details are supplied in the Appendix G.2.

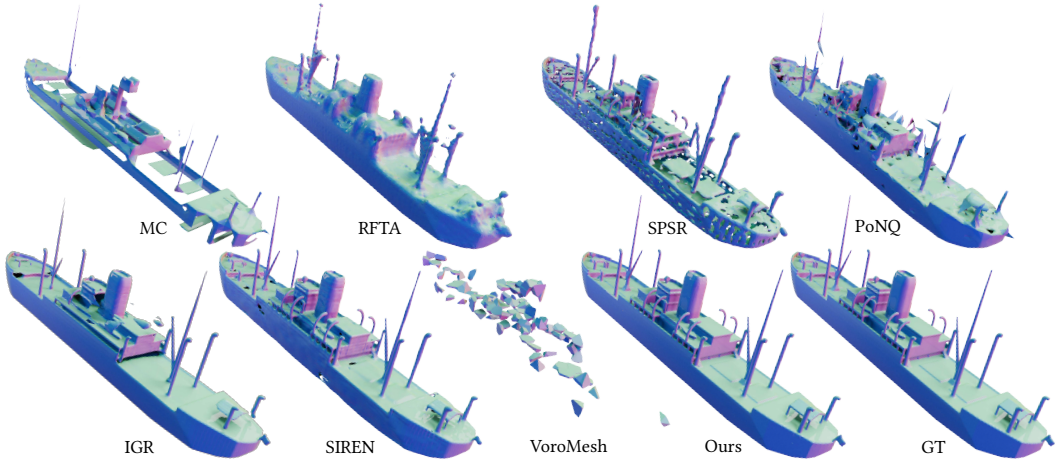


Figure 6: Qualitative results on ShapeNet (Chang et al., 2015).

Table 4: Ablation studies of Discrete Matching (DM), Assignment Stabilization (AS), Geometric Initialization (GI), Pruning (P), Normal Inversion Handling (NIH), and Learning Rate Scheduler (LRS) on the ABC / Thingi10k dataset. The bold text indicates the best score.

	CH ↓	HD ↓	FS ↑	NC ↑	IoU ↑
w.o. DM	2.48 / 2.39	4.93 / **5.88**	60.83 / 61.59	97.55 / 97.26	96.05 / 97.34
w.o. AS	6.15 / 3.15	33.9 / 13.3	66.98 / 74.74	93.64 / 97.44	95.46 / 97.84
w.o. GI	141 / 106	102 / 78.1	9.99 / 10.1	56.48 / 57.83	21.32 / 33.71
w.o. P	4.92 / 3.84	24.5 / 22.7	71.27 / 76.27	96.23 / 97.39	95.19 / 97.28
w.o. NIH	2.08 / 2.51	4.55 / 8.13	72.64 / 76.74	97.89 / **98.17**	97.96 / 98.05
w.o. LRS	2.17 / 2.96	7.53 / 9.35	72.05 / 74.80	97.94 / 97.89	97.22 / 97.16
Full	**1.90 / 1.80**	**3.81** / 6.78	**73.22 / 77.27**	**98.05 / 98.17**	**98.39 / 99.14**

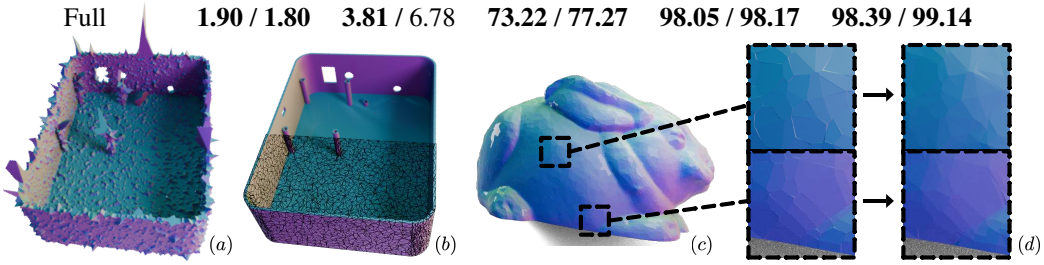


Figure 7: (a) Our geometric initialization, (b) optimization result (top) and its tessellation (bottom). (c) The raw output of our explicit polygonalization can have minor seams; (d) these can mostly be dissolved with our edge collapsing routine.

4.3 Ablation Studies

We ablate our major design choices to isolate their contributions to our final performance (Tab. 4). Specifically, we study discrete matching, assignment stabilization, geometric initialization, progressive pruning, normal inversion handling, and the learning rate scheduler on ABC/thingi10k datasets.

Discrete Matching (DM). We ablate the importance of discrete matching against fitting patchwork directly as a black-box implicit function. Such an implementation is not unique; we report the best-performing variant we were able to tune, with details in the Appendix I. The discrete matching improves the Chamfer Distance and F-score, and significantly boosts the visual fidelity.

Assignment Stabilization (AS). To assess the impact of assignment stabilization, we keep the matched maximum assignment regardless of how small their angle differences are. This configuration is prone to gradient noise, such that the pruning set is hard to stabilize, resulting in significantly more floators, as indicated by a substantially higher Hausdorff Distance.

Geometric Initialization (GI). We replace the geometric initialization with random uniform initialization. Although it can fit to some degree, the foam tessellation is significantly sparse. Moreover, it is incompatible with our pruning loss, such that the resulting scene is populated with floators.

Pruning (P). We disable pruning, so that polyhedra that fall out of the surface remain present, yielding a higher Hausdorff Distance.

Normal Inversion Handling (NIH). We remove the Normal Inversion Handling and observe that some outwards-extended inner polyhedra remain unpruned after the optimization.

Learning Rate Scheduler (LRS). In our formulation, the difference of power distance is subject to a constant shift, so the learning rate scheduler anneals the oscillation in that null space. Moreover, it also stabilizes the maximum assignment, the pruning set, towards the end of optimization. Removing it introduces gaps between patches and floators off the surface, degrading the visual quality.

5 Applications

Beyond typical applications of implicit volumetric representations, such as compression, ray tracing, and inside/outside classification (Fig. 1), our method additionally supports explicit extraction of conforming volumetric meshes of convex volume elements, enabling direct downstream tasks like fracture simulation (Fig. 8).

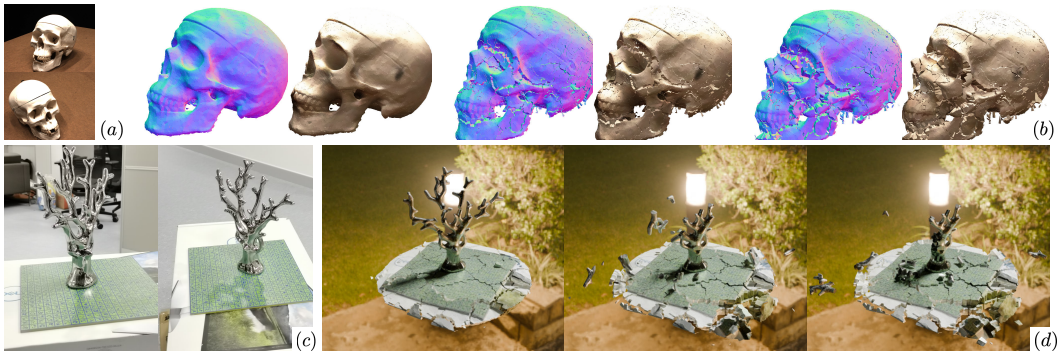


Figure 8: Fracture simulation. Our representation is fitted from pretrained neural rendering models on multi-view images (a, c); fracture sequences are rendered via (b) Neural Deferred Shading (Worchel et al., 2022) and (d) relit with inverse-rendered BRDF materials (Liu et al., 2023).

6 Limitations and Future Work

Limitations. As in related lines of work (Tab. 1), our optimization landscape is sensitive to local minima and requires careful initialization. Our initialization may produce incorrectly extended polyhedra that are not pruned due to insufficient sampling or erroneous normal constraints (e.g., self-intersections in the GT mesh). We observe both cases in the Thingi10K dataset, as indicated by our higher Hausdorff Distance. Our method requires clean constraints and is sensitive to noise (Fig. 9). On curved surfaces, seams can appear when our tessellation does not fully align with the ground truth mesh. This is a historical challenge dating back to Variational Surface Approximation (Cohen-Steiner et al., 2004). For visual results, we mitigate this with a custom edge-collapse routine (Fig. 7 (d)). We expect generalizing the cocircularity constraint of concurrent work VoroLight (Lu et al., 2026) to our setting would further improve visual quality.

Future work. Our current optimization does not guarantee a theoretically minimal configuration and is limited to closed manifold surfaces. In principle, however, inter-polyhedral facets of foams can exhibit more diverse topologies, including open and non-manifold surfaces. We leave them, together with adaptive polyhedra placement and densification extensions, for future work. Finally, given the increasing interest in part-level shape understanding via volumetric feature fields (Liu et al., 2025; Zhu et al., 2026), our representation may serve as an effective medium for applications such as semantics-aware shape decomposition and fracturing.

Table 5: Per-iteration performance of our Triton / KD-tree implementations with batch size $K = 16384$ over varying N, measured on a single NVIDIA RTX3090. The parenthesized values show KD query time only.

N	Time (s)	VRAM (MB)
16384	0.0335 / 0.0341 (0.0035)	352 / 384
65536	0.0375 / 0.0614 (0.0101)	364 / 394
655360	0.1927 / 0.1060 (0.0392)	520 / 620
1048576	0.2992 / 0.1510 (0.0395)	616 / 794
16777216	4.6207 / 2.1561 (1.3151)	4458 / 6796

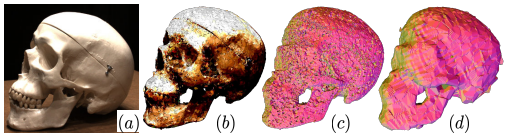


Figure 9: Noisy constraints degrade our output to low-poly-like results. (a) Reference view. (b) Textured and normal rendering (c) of the pretrained SDFoam raw output. (d) Our fit using constraints estimated from volume density.

7 Conclusions

We present patchwork, a foam-based 3D representation that explicitly encodes shapes as piecewise polyhedra while exhibiting a simple implicit form. We formally prove its representation bounds and approximation capability, and introduce a combinatorial matching scheme that enables fitting patchwork to its theoretical potential. Extensive comparisons against existing baselines under carefully matched parameter counts show that patchwork surpasses them while requiring 3–6$\times$ fewer parameters on average. We further demonstrate its unique ability to extract conforming volume meshes, enabling direct downstream fracture simulation. We envision patchwork becoming a valuable foundation for geometric learning.

AI USE STATEMENT

In this work, we have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, or interpret results. Generating synthetic data sets, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. We used generative AI tools to help with writing and reviewing the manuscript for consistency and grammar mistakes. We have reviewed, fact-checked, and revised all AI-assisted texts. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

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, 39(3):1–16, May 2020. ISSN 1557-7368. doi: 10.1145/3337680. URL http://dx.doi.org/10.1145/3337680.

Motasem Alfarra, Adel Bibi, Hasan Hammoud, Mohamed Gaafar, and Bernard Ghanem. On the decision boundaries of neural networks: A tropical geometry perspective. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(4):5027–5037, 2022.

Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA '01, pp. 249–266, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133669. doi: 10.1145/376957.376986. URL https://doi.org/10.1145/376957.376986.

Matan Atzmon and Yaron Lipman. Sald: Sign agnostic learning with derivatives, 2020a. URL https://arxiv.org/abs/2006.05400.

Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data, 2020b. URL https://arxiv.org/abs/1911.10414.

Franz Aurenhammer. Power diagrams: properties, algorithms and applications. SIAM journal on computing, 16(1):78–96, 1987.

Yoshua Bengio. Estimating or propagating gradients through stochastic neurons. arXiv preprint arXiv:1305.2982, 2013.

Marie-Charlotte Brandenburg, Georg Loho, and Guido Montufar. The real tropical geometry of neural networks for binary classification. Transactions on Machine Learning Research, 2024.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. URL https://arxiv.org/abs/1512.03012.

Wylliam Cantin Charawi, Adrien Gruson, Jane Wu, Christian Desrosiers, and Diego Thomas. Dccvt: Differentiable clipped centroidal voronoi tessellation. In Proceedings of the 15th International Conference on 3D Vision (3DV), Vancouver, BC, Canada, March 2026.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), pp. 16251–16261, June 2025.

Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 45–54, 2020.

David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. In ACM siggraph 2004 papers, pp. 905–914. 2004.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems, 35:16344–16359, 2022.

Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

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.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 605–613, 2017.

Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97, pp. 209–216, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0897918967. doi: 10.1145/258734.258849. URL https://doi.org/10.1145/258734.258849.

Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Radiant foam: Real-time differentiable ray tracing, 2025. URL https://arxiv.org/abs/2502.01157.

Shrisudhan Govindarajan, Daniel Rebain, Dor Verbin, Kwang Moo Yi, Anish Prabhu, and Andrea Tagliasacchi. Power foam: Unifying real-time differentiable ray tracing and rasterization, 2026. URL https://arxiv.org/abs/2604.24994.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes, 2020. URL https://arxiv.org/abs/2002.10099.

Dmitrii Andreevich Gudkov. The topology of real projective algebraic varieties. Russian Mathematical Surveys, 29(4):1–79, 1974. doi: https://doi.org/10.1070/RM1974v029n04ABEH001288.

Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4369–4379, 2023.

Ilia Itenberg and Oleg Viro. Patchworking algebraic curves disproves the Ragsdale conjecture. Mathematical Intelligencer, 18(4), 1996.

Ilia Itenberg and Oleg Viro. Asymptotically maximal real algebraic hypersurfaces of projective space. In Gökova Geometry/Topology conference, pp. 91–105, 2006.

Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. ACM Trans. Graph., 32(3), July 2013. ISSN 0730-0301. doi: 10.1145/2487228.2487237. URL https://doi.org/10.1145/2487228.2487237.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. ACM Trans. Graph., 42(4):139–1, 2023.

Mikhail Khristoforov, Mikhail Skopenkov, and Stanislav Smirnov. A generalization of Cardy's and Schramm's formulae. Communications in Mathematical Physics, 406(5):105, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics, 36(4), 2017.

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 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

Yaron Lipman. Phase transitions, distance functions, and implicit neural representations, 2021. URL https://arxiv.org/abs/2106.07689.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. 2025.

Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multi-view images. In SIGGRAPH, 2023.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH Comput. Graph., 21(4):163–169, August 1987. ISSN 0097-8930. doi: 10.1145/37402.37422. URL https://doi.org/10.1145/37402.37422.

Jiayin Lu, Ying Jiang, Yumeng He, Yin Yang, and Chenfanfu Jiang. Vorolight: Learning voronoi surface meshes via sphere intersection, 2026. URL https://arxiv.org/abs/2512.12984.

Diane Maclagan and Bernd Sturmfels. Introduction to tropical geometry, volume 161. American Mathematical Soc., 2015.

Petros Maragos, Vasileios Charisopoulos, and Emmanouil Theodosis. Tropical geometry and machine learning. Proceedings of the IEEE, 109(5):728–755, 2021.

Nissim Maruani, Roman Klokov, Maks Ovsjanikov, Pierre Alliez, and Mathieu Desbrun. Voromesh: Learning watertight surface meshes with voronoi diagrams. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 14565–14574, 2023.

Nissim Maruani, Maks Ovsjanikov, Pierre Alliez, and Mathieu Desbrun. Ponq: a neural qem-based mesh representation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3647–3657, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. URL https://arxiv.org/abs/2003.08934.

Jason Miller, Scott Sheffield, and Wendelin Werner. CLE percolations. Forum of Mathematics, Pi, 5:e4, 2017. doi: 10.1017/fmp.2017.5.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph., 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 127–136, 2011. doi: 10.1109/ISMAR.2011.6092378.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 2019.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pp. 523–540. Springer, 2020.

Olimjoni Pirahmad, Helmut Pottmann, and Mikhail Skopenkov. Flexible polyhedral nets in isotropic geometry. Beitr. Algebra Geom., 67:583–617, 2026. URL https://doi.org/10.1007/s13366-025-00820-y.

Antonella Rech, Nicola Conci, and Nicola Garau. Sdfoam: Signed-distance foam for explicit surface reconstruction, 2025. URL https://arxiv.org/abs/2512.16706.

Jing Ren, Biao Zhang, Bojian Wu, Jianqiang Huang, Lubin Fan, Maks Ovsjanikov, and Peter Wonka. Intuitive and efficient roof modeling for reconstruction and synthesis. ACM Trans. Graph., 40(6), December 2021. ISSN 0730-0301. doi: 10.1145/3478513.3480494. URL https://doi.org/10.1145/3478513.3480494.

Silvia Sellán, Yingying Ren, Christopher Batty, and Oded Stein. Reach for the arcs: Reconstructing surfaces from sdfs via tangent points. In SIGGRAPH 2024 Conference Papers, 2024.

Silvia Sellán, Christopher Batty, and Oded Stein. Reach for the spheres: Tangency-aware surface reconstruction of sdfs, 2023. URL https://arxiv.org/abs/2308.09813.

Nicholas Sharp et al. Polyscope, 2019. www.polyscope.run.

Ali Siahkamari, Aditya Gangrade, Brian Kulis, and Venkatesh Saligrama. Piecewise linear regression via a difference of convex functions. In International conference on machine learning, pp. 8895–8904. PMLR, 2020.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. Advances in neural information processing systems, 33:7462–7473, 2020.

Youran Sun, Yihua Liu, and Yi-Shuai Niu. Understand the effectiveness of shortcuts through the lens of dca, 2024. URL https://arxiv.org/abs/2412.09853.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11358–11367, 2021.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. Advances in neural information processing systems, 33:7537–7547, 2020.

Philippe Tillet, H. T. Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL 2019, pp. 10–19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367196. doi: 10.1145/3315508.3329973. URL https://doi.org/10.1145/3315508.3329973.

J.W. Tukey. Exploratory Data Analysis. Addison-Wesley series in behavioral science. Addison-Wesley Publishing Company, 1977. ISBN 9780201076165.

Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2635–2643, 2017.

Oleg Viro. Progress in the topology of real algebraic varieties over the last six years. Russian Mathematical Surveys, 41(3):55–82, 1986. doi: https://doi.org/10.1070/RM1986v041n03ABEH003317.

Ingo Wald. Gpu-friendly, parallel, and (almost-)in-place construction of left-balanced k-d trees, 2023. URL https://arxiv.org/abs/2211.00120.

Ningna Wang, Hui Huang, Shibo Song, Bin Wang, Wenping Wang, and Xiaohu Guo. Mattopo: Topology-preserving medial axis transform with restricted power diagram, 2024. URL https://arxiv.org/abs/2403.18761.

Ningna Wang, Rui Xu, Yibo Yin, Zichun Zhong, Taku Komura, Wenping Wang, and Xiaohu Guo. Matstruct: High-quality medial mesh computation via structure-aware variational optimization. In Proceedings of the SIGGRAPH Asia 2025 Conference Papers, SA Conference Papers '25, pp. 1–12. ACM, December 2025. doi: 10.1145/3757377.3763840. URL http://dx.doi.org/10.1145/3757377.3763840.

Xinyue Wei, Fanbo Xiang, Sai Bi, Anpei Chen, Kalyan Sunkavalli, Zexiang Xu, and Hao Su. Neumanifold: Neural watertight manifold reconstruction with efficient and high-quality rendering support. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 731–741. IEEE, 2025.

Francis Williams, Matthew Trager, Joan Bruna, and Denis Zorin. Neural splines: Fitting 3d surfaces with infinitely-wide neural networks, 2021. URL https://arxiv.org/abs/2006.13782.

Markus Worchel, Rodrigo Diaz, Weiwen Hu, Oliver Schreer, Ingo Feldmann, and Peter Eisert. Multi-view mesh reconstruction with neural deferred shading. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6187–6197, June 2022.

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. Computer Graphics Forum, 2022. ISSN 1467-8659. doi: 10.1111/cgf.14505.

Qiaochu Yuan. Hypersurfaces, 4-manifolds, and characteristic classes, 2014. URL https://qchu.wordpress.com/2014/06/16/hypersurfaces-4-manifolds-and-characteristic-classes/.

Biao Zhang and Peter Wonka. efunc: An efficient function representation without neural networks. arXiv preprint arXiv:2505.21319, 2025.

Biao Zhang, Jing Ren, and Peter Wonka. Geometry distributions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1495–1505, October 2025.

Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In International Conference on Machine Learning, pp. 5824–5832. PMLR, 2018.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets, 2024. URL https://arxiv.org/abs/2406.13897.

Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. arXiv preprint arXiv:1605.04797, 2016.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. Partsam: A scalable promptable part segmentation model trained on native 3d data, 2026. URL https://arxiv.org/abs/2509.21965.

A Origins of the method

Originally, patchwork representation for curves appeared in pure mathematics as a constructive approach to famous *Hilbert's sixteenth problem*. This problem was stated in the milestone talk "Mathematical Problems" by David Hilbert at the Second International Congress of Mathematicians in 1900. The first part of this problem asked for a possible number and mutual position of components of a curve represented by an algebraic equation of a given degree. This question remains widely open, starting from degree 8. It further decomposes into two sub-tasks: to *construct* an algebraic equation that represents a curve of given topological type and to *bound* the complexity of algebraic curves of given degree.

Patchworking is an approach to this construction, which traces back to the works by Gudkov (Gudkov, 1974) from the 1970s and was later formalized and generalized by Viro (Viro, 1986). See an elementary introduction in (Itenberg & Viro, 1996). Using this approach, a counterexample to the

original Hilbert's result was constructed (seventy years later), as well as to other famous conjectures in algebraic geometry, and Hilbert's problem was solved for equations of degree up to 7. Furthermore, it served as a motivation and foundation to *tropical geometry*, which studies piecewise-linear analogs of algebraic curves and surfaces (Maclagan & Sturmfels, 2015). The underlying formalism has much similarity with neural networks, so it is natural to build a bridge from patchworks to neural shape representations; cf. (Brandenburg et al., 2024).

The *complexity bounds* for algebraic curves are obtained in *real algebraic geometry* by very different methods (Yuan, 2014). An example is Harnack's curve theorem, which bounds the number of components of an algebraic curve of given degree. Such results also bound the representation capacity of patchworks. For algebraic surfaces in space, a tight bound is unknown starting from degree 5; see (Itenberg & Viro, 2006).

Our approach is directly inspired by the mathematical concept of patchworking (Viro, 1986; Itenberg & Viro, 1996; 2006) and tropical geometry (Maclagan & Sturmfels, 2015). For applications of tropical geometry for deep learning, see (Alfarra et al., 2022; Brandenburg et al., 2024; Maragos et al., 2021; Zhang et al., 2018).

B Introduction to the method

In this supplementary material, we introduce our main concept step by step. We start with a concise definition in 2D, with a discussion of its visualization and expressive power. Then we generalize it to 3D, rephrasing Definition 1 in an equivalent form, more suitable for theoretical analysis.

B.1 2D

We represent a curve in the plane as the zero level set of a field $f\colon \mathbb{R}^2 \to \mathbb{R}$ of a certain special form.

Definition 2 (See Figs. 10–11). *A 2D patchwork field is a function $\mathbb{R}^2 \to \mathbb{R}$ of the form*

$$f(x,y) = \max_{\substack{1\le i\le n:\\ s_i=+1}} \{a_i x + b_i y + c_i\} - \max_{\substack{1\le i\le n:\\ s_i=-1}} \{a_i x + b_i y + c_i\}\,, \tag{8}$$

depending on the parameters $n \in \mathbb{Z}_{>0}$ (called width), $s_i \in \{+1,-1\}$ (signs), and $a_i, b_i, c_i \in \mathbb{R}$, where the first maximum is taken over indices $i \in \{1,\dots,n\}$ such that $s_i = +1$ and the second maximum is over i such that $s_i = -1$. A patchwork is the curve in the plane given by $f(x,y) = 0$.

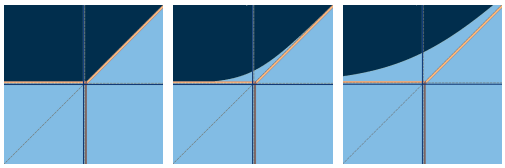


Figure 10: Left: A patchwork (the boundary between the dark and light areas). The design lines (blue), the equality lines (dashed), the candidate intervals (brown/orange), and the active intervals (orange). Just two design lines represent a concave angle. Middle and right: smoothed patchworks of increasing smoothness T (from left to right). The patchwork (left) is the limiting case $T \searrow 0$.

Visualization. This construction is visualized as follows; see Fig. 10. Consider the patchwork $f(x,y) = 0$ and let us determine its shape. Introduce the linear function

$$l_i(x,y) := a_i x + b_i y + c_i \qquad \text{for each } i = 1,\dots,n. \tag{9}$$

Assume that the functions $l_1(x,y),\dots,l_n(x,y)$ are distinct and one of them is a positive constant. If $l_i(x,y)$ is not a constant, then $l_i(x,y) = 0$ is an equation of a line in the plane, referred to as a design line. For instance, in Fig. 10, $l_1(x,y) = x$, $l_2(x,y) = y$, $l_3(x,y) = 0.01$ so that the design lines are the coordinate axes.

Observe that function (8) vanishes only if both maxima are equal, in particular, $l_i(x,y) = l_j(x,y)$ for some $i \ne j$. The lines given by the equations $l_i(x,y) = l_j(x,y)$ for all pairs of distinct indices

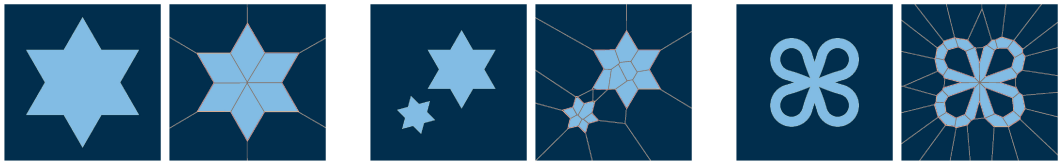


Figure 11: Nonconvex (left), disconnected (middle), and non-simply-connected (right) shapes and their approximations by patchworks computed by optimization (right per pair). Each patchwork is the boundary between a dark and a light region. These compact representations have just $n = 12, 37, 55$ design lines (left to right). The candidate intervals (brown/orange) decompose the plane into polygons. The "interior" of the patchwork is the union of the polygons corresponding to the indices i with $s_i = -1$, and the patchwork is the union of active intervals (orange).

$i, j \in \{1, \dots, n\}$ are called equality lines. In what follows, fix such a pair $i \neq j$. For instance, in Fig. 10, the equality lines are $x = y$, $x = 0.01$, and $y = 0.01$.

Next, a point (x, y) on the equality line satisfies $f(x, y) = 0$, only if the value $l_i(x, y) = l_j(x, y)$ is maximal among $l_1(x, y), \dots, l_n(x, y)$. The intervals where

$$l_i(x, y) = l_j(x, y) \geq \max_k l_k(x, y) \tag{10}$$

are called candidate intervals. They can be straight line segments, rays, or lines. For instance, in Fig. 10, these are three rays emanating in the north-east, south, and west directions. Their union is known as the tropical line.

The candidate intervals split the plane into convex polygons, possibly empty or unbounded, given by the condition $l_i(x, y) \geq \max_k l_k(x, y)$ for some i. Each one is referred to as the candidate polygon (or the patchwork cell) corresponding to the index i. In Fig. 10 on the left, ones with $s_i = -1$ are light, and the one with $s_i = +1$ is dark.

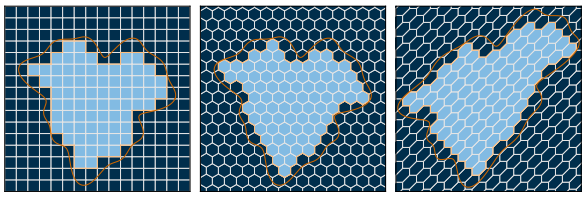


Figure 12: Patchworks (orange) approximating a given curve (red). The construction can employ a square grid (left), a honeycomb lattice (middle), or an affine transformation of the latter (right). The squares/hexagons inside the curve are light, the other ones are dark, and the patchwork separates the colors.

Finally, to satisfy $f(x, y) = 0$, the signs s_i and s_j must be opposite. Candidate intervals with this property are called active intervals. In Fig. 10, they are orange.

We conclude that the patchwork is the union of active intervals (if some $l_i(x, y)$ is a positive constant, and the parameters are in general position; see Appendix C).

Relation to power diagrams and piecewise linear functions. Eq. (8) can be rewritten in the following form by negating the linear functions and completing the squares:

$$f(x) = \min_{\substack{1 \leq i \leq n: \\ s_i = -1}} \{|x - p_i|^2 - w_i\} - \min_{\substack{1 \leq i \leq n: \\ s_i = +1}} \{|x - p_i|^2 - w_i\}, \tag{11}$$

where $p_i = \frac{1}{2}(a_i, b_i) \in \mathbb{R}^2$ and $w_i = \frac{1}{4}(a_i^2 + b_i^2) + c_i \in \mathbb{R}$. This form shows that the candidate polygons are exactly the cells of the power diagram with points p_i and weights w_i. However, Eq. (1) is simpler and more convenient for both theoretical analysis and optimization.

Next, a patchwork field is clearly linear in each candidate polygon, hence is a piecewise-linear function. Conversely, each continuous piecewise linear function can be represented in form (8), thus is a patchwork field; see (Siahkamari et al., 2020). However, form (1) is more convenient for us.

Expressive power. Our construction allows visual control and compact representations even of complicated non-convex shapes; see Fig. 11. It also suggests the following design of a patchwork approximating a given curve; see Fig. 12.

Let the curve be inside the box $|x|, |y| \le 1$. Superimpose a grid of step $1/N$. Paint the squares lying inside the curve light, and the remaining ones dark. The union of the sides separating colors is called a digital curve. It can be explicitly represented as a patchwork. For notational convenience, replace the index i in (8) with a pair of indices $k, l \in [-N, N]$, so we write $a_{k,l}$ instead of a_i etc. Set

$$a_{k,l} = k, \quad b_{k,l} = l, \quad c_{k,l} = (1 - k^2 - l^2)/(2N),$$
$$s_{k,l} = \begin{cases} +1, & \text{if the candidate polygon} \\ & \text{corresponding to the pair } k, l \text{ is dark,} \\ -1, & \text{otherwise.} \end{cases} \tag{12}$$

In particular, $l_{0,0}(x, y) = 1/(2N)$. The resulting candidate polygons form a square grid inside the box (this can be shown using power diagrams or (Pirahmad et al., 2026, Lemma 1)), and the patchwork coincides with the digital curve. Thus, any plane curve can be approximated by patchworks. See Theorem 3 in Appendix D. There, we also discuss hexagonal lattices (see Fig. 12 on the right; cf. (Khristoforov et al., 2025, Lemma 3.8)). We can further optimize the representation; see Fig. 11.

We also introduce smoothed patchworks by replacing the maximum in (8) with a suitable soft-max.

Definition 3 (See Fig. 10)**.** *A smoothed 2D patchwork field is*

$$F(x, y) = T \log \sum_{i=1}^{n} s_i \exp \frac{a_i x + b_i y + c_i}{T} \tag{13}$$

depending on an additional parameter $T \in \mathbb{R}_{>0}$ (called smoothness) and defined on the subset of $\mathbb{R}^2$ where the sum is positive. A smoothed patchwork is the curve in the plane given by $F(x, y) = 0$.

The smaller the smoothness parameter T is, the closer the smoothed patchwork is to a patchwork with the same remaining parameters. See Theorem 1 in Appendix C. We obtain some patchwork complexity bounds in Appendix E, (see Tab. 6).

B.2 3D

The construction in 3D is completely analogous. We represent a closed surface S as the zero level set of a field $f\colon \mathbb{R}^3 \to \mathbb{R}$ of a certain special form. Compared to a polygonal mesh, this gives a simple way to determine if a point x is inside S by computing the sign of $f(x)$.

Definition 4 (See Fig. 13)**.** *A patchwork field is a function $\mathbb{R}^3 \to \mathbb{R}$ of the form*

$$f(x) = \max_{\substack{1 \le i \le n: \\ s_i = +1}} \{\langle a_i, x\rangle + c_i\} - \max_{\substack{1 \le i \le n: \\ s_i = -1}} \{\langle a_i, x\rangle + c_i\}, \tag{14}$$

depending on the parameters $n \in \mathbb{Z}_{>0}$ (called width), $s_i \in \{+1, -1\}$ (signs), $a_i \in \mathbb{R}^3$, and $c_i \in \mathbb{R}$, where the first maximum is taken over indices $i \in \{1, \dots, n\}$ such that $s_i = +1$ and the second maximum is over i such that $s_i = -1$. A patchwork is the surface in $\mathbb{R}^3$ given by $f(x) = 0$.

We also introduce a smoothed patchwork field

$$F(x) = T \log \sum_{i=1}^{n} s_i \exp \frac{\langle a_i, x\rangle + c_i}{T} \tag{15}$$

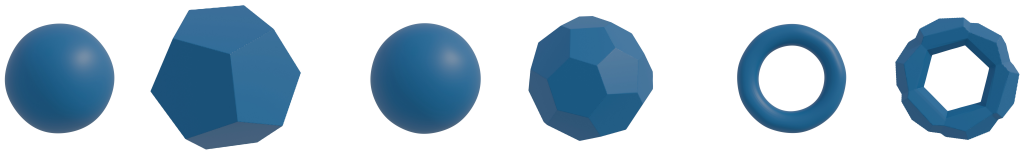
depending on an additional parameter $T \in \mathbb{R}_{>0}$ (called smoothness) and defined on the subset of $\mathbb{R}^3$ where the sum is positive. A smoothed patchwork is the surface in $\mathbb{R}^3$ given by $F(x) = 0$.

Visualization. Introduce linear functions $\mathbb{R}^3 \to \mathbb{R}$

$$l_i(x) = \langle a_i, x\rangle + c_i. \tag{16}$$

The planes $l_i(x) = 0$ (with $l_i(x) \ne \text{const}$) are called design planes. The planes $l_i(x) = l_j(x)$ for distinct $i, j \in \{1, \dots, n\}$ are called equality planes. The (possibly unbounded) convex polygons given by the conditions

$$l_i(x) = l_j(x) \ge \max_k l_k(x) \tag{17}$$



(a) (b) (c) (d) (e) (f)

Figure 13: Smoothed patchworks approximating a sphere (a and c) and a torus (e) and their limiting cases $T \searrow 0$ (b, d, f). The smoothed patchworks (a, c, e) visually coincide with the shapes, although their width is just $n = 13, 31, 30$ (left to right). For (a and b), the 12 design planes contain the faces of the dodecahedron (b), and the 13th linear function is constant. For (e and f), just 30 design planes represent an intricate nonconvex shape.

are called candidate polygons. Active polygons are the candidate polygons such that $s_i s_j < 0$. The candidate polygons decompose space into convex polyhedra (possibly empty or unbounded) given by the condition $l_i(x) \geq \max_k l_k(x)$ for some i. These polyhedra are called candidate polyhedra.

For general position functions $l_i(x)$ including a positive constant, the patchwork is just the union of active polygons. For small smoothness $T > 0$, the smoothed patchwork is close to the union. See Theorem 2 in Appendix C. This leads to compact surface representations; see Fig. 13. Expressive power and complexity bounds of 3D (smoothed) patchworks are established in Theorem 4 in Appendix D and Tab. 7 in Appendix E, respectively.

C Dependence on the parameters

In this supplementary material, we discuss the dependence of the patchwork on the parameters and clarify the technical assumptions required for Section B above. The derivations there are valid in general position, that is, unless the parameters satisfy certain specific relations. We show that a general position patchwork $f(x, y) = 0$ is a limiting case of $F(x, y) = 0$ as $T \searrow 0$; see Fig. 10.

More precisely, with notation (8)–(9), we say that $a_i, b_i, c_i \in \mathbb{R}$ are general position parameters including a positive constant, if the following conditions hold:

- for some index i, we have $l_i(x, y) = \text{const} > 0$;
- for distinct indices i, the functions $l_i(x, y)$ are distinct;
- for any triple of distinct indices i, j, k, the equation $l_i(x, y) = l_j(x, y) = l_k(x, y)$ has finitely many solutions;
- for any four distinct indices $i_1, \dots, i_4$, the equation $l_{i_1}(x, y) = \cdots = l_{i_4}(x, y)$ has no solutions.

By the argument in Appendix B, for general position parameters including a positive constant, the patchwork $f(x, y) = 0$ is the union of active segments. Here, none of the conditions can be dropped. For instance, for $l_1(x, y) = x$, $l_2(x, y) = y$, $l_3(x, y) = 1$, $l_4(x, y) = (x + y + 1)/3$, $s_1 = s_2 = s_3 = -s_4 = 1$ (violating the last condition), the set $f(x, y) = 0$ consists of a single point $(1, 1)$, whereas there are no active segments. The following folklore theorem holds.

Theorem 1 (Tropical limit)**.** *With notation (8), (9), and (13), for any general position parameters including a positive constant, the Hausdorff distance between the sets given by $F(x, y) = 0$ and $f(x, y) = 0$ tends to zero as $T \searrow 0$.*

Proof. Set $\beta := 1/T$. Take $\varepsilon > 0$ and consider the ε-neighborhood U of the patchwork $f(x, y) = 0$ and the candidate polygon P_i corresponding to an index i. It suffices to show that the sign of $e^{\beta F(x,y)} - 1$ in $P_i \setminus U$ equals s_i for sufficiently large β, where the function $e^{\beta F(x,y)}$ is extended to points (x, y) where the argument of the logarithm in Eq. (13) is negative. Clearly, then the Hausdorff distance between the sets given by $F(x, y) = 0$ and $f(x, y) = 0$ is $O(\varepsilon)$ and tends to zero as $T \searrow 0$.

We start by showing that $f(x, y)$ is bounded from 0 in $\mathbb{R}^2 \setminus U$. Indeed, take a box B containing all pairwise intersections of equality lines. The complement of $B \cup U$ can be covered by rays

starting at the boundary of B and not crossing the equality lines. On each ray, $f(x,y)$ is linear and non-vanishing. Hence, $|f(x,y)|$ attains a minimum at the starting point. By compactness, there is $\delta = \delta(\varepsilon) > 0$ such that $|f(x,y)| > \delta$ in $B \setminus U$, hence in $\mathbb{R}^2 \setminus U$.

Next, take $(x,y) \in P_i \setminus U$. By the definition of P_i, the sign of $f(x,y)$ equals s_i. Assume that $s_i = +1$ (the other case is analogous). Then $f(x,y) > \delta$. Pick any

$$j \in \arg\max_{k:s_k<0} \{l_k(x,y)\}.$$

Then $l_i(x,y) > \delta + l_j(x,y)$. Since some $l_k(x,y) = c > 0$ is a positive constant, it follows that $l_i(x,y) \geq c$. Then

$$e^{\beta F(x,y)} = \sum_{k:s_k>0} e^{\beta l_k(x,y)} - \sum_{k:s_k<0} e^{\beta l_k(x,y)} \geq \begin{cases} \left(e^{\beta\delta} - n\right) e^{\beta l_j(x,y)}, & \text{if } l_j(x,y) \geq 0, \\ e^{\beta c} - n, & \text{if } l_j(x,y) < 0, \end{cases} \tag{18}$$

We conclude that $e^{\beta F(x,y)} > 1$, once

$$\beta > \frac{\log(n+1)}{\min\{\delta, c\}}. \tag{19}$$

Analogously, $e^{\beta F(x,y)} < 1$ in $P_i \setminus U$ for $s_i = -1$. QED □

Let us give a few remarks on the dependence on the parameters other than T. The change of a_i, b_i, c_i, s_i affects only the boundary of the candidate polygon corresponding to index i, and hence is a local change of the patchwork $f(x,y) = 0$. The changes to the smoothed patchwork $F(x,y) = 0$ exponentially decrease with distance from the polygon. When a sign s_i flips, a polygon is abruptly added to or removed from the "interior" of the patchwork $f(x,y) = 0$. For the smoothed patchwork $F(x,y) = 0$, a similar jump occurs.

In 3D, we say that $a_i \in \mathbb{R}^3$ and $c_i \in \mathbb{R}$ are general position parameters including a positive constant if

- for some index i, we have $l_i(x) = \text{const} > 0$;
- for any $k \geq 2$ distinct indices $i_1, \ldots, i_k$, we have

$$\dim\{x \in \mathbb{R}^3 : l_{i_1}(x) = \cdots = l_{i_k}(x)\} \leq 4 - k. \tag{20}$$

For such parameters, the patchwork $f(x) = 0$ is the union of active polygons. The following folklore theorem is proved analogously to Theorem 1.

Theorem 2 (Tropical limit)**.** *With notation (14)–(16), for any general position parameters including a positive constant, the Hausdorff distance between the sets given by $F(x) = 0$ and $f(x) = 0$ tends to zero as $T \searrow 0$.*

D EXPRESSIVE POWER

In this supplementary material, we demonstrate the expressive power of patchworks: first, in 2D, then, in 3D. We prove the following folklore theorem.

Theorem 3 (Expressive power)**.** *For any Jordan curve and any $\varepsilon > 0$, there exists a patchwork (that is again a Jordan curve) within the Hausdorff distance ε from the curve. For a smooth curve, there is a patchwork of width $O(1/\varepsilon^2)$ with this property.*

Here, roughly speaking, a Jordan curve is a closed, continuous curve without self-intersections. More precisely, a Jordan curve is the image of an injective continuous map of a circle into the plane. A Jordan curve need not be smooth, so that the theorem applies to fractal curves as well. The generalization to curves having finitely many disjoint components is straightforward. Thus, we obtain a method to approximate essentially any plane curve by a patchwork.

Recall that any piecewise linear function can be represented in form (8). Thus, any polyline without self-intersections or a disjoint union of such polylines can be represented as a patchwork. The advantage of our method is a simple, explicit expression for the parameters.

Proof. We prefer to use a hexagonal lattice for approximation. Replace the index i in (13) with a pair of indices $k, l \in [-N, N]$, so we write $a_{k,l}$ instead of a_i etc. Consider a patchwork with parameters

$$a_{k,l} = k, \quad b_{k,l} = l, \quad c_{k,l} = -(k^2 + l^2 + kl - 1)/N, \quad s_{k,l} \in \{+1, -1\}. \tag{21}$$

The resulting candidate polygons are hexagons with the sides parallel to the lines $x = 0$, $y = 0$, and $x = y$. They form an affine-transformed honeycomb lattice; see Fig. 12 on the right. The resulting patchwork (that is, the union of polylines separating candidate polygons corresponding to the indices k, l with $s_{k,l}$ of opposite signs) is called a hex-curve (of step $1/N$).

It suffices to prove Theorem 3 for a smooth curve, because any Jordan curve can be approximated by smooth ones in Hausdorff distance. By (Khristoforov et al., 2025, Lemma 3.8), for any smooth Jordan curve, there is a constant C (depending on the curve) such that for any sufficiently small mesh step $\delta > 0$ there is a Jordan hex-curve of step δ within Hausdorff distance $C\delta$ from the smooth curve. For $\delta < \varepsilon/C$, the hex-curve is within Hausdorff distance ε. Assume WLOG that the smooth curve lies in the box $|x|, |y| < 1$. Then the hex-curve is the patchwork with parameters (21) and has width at most $(2/\delta)^2 = O(1/\varepsilon^2)$, as required. QED □

Note that for a non-smooth curve, the Hausdorff distance to the hex-curve can decrease with $\delta \to 0$ slower than any linear function in δ (Khristoforov et al., 2025, Remark after Lemma 3.8).

As a byproduct, our representation allows direct sampling of certain complicated random fractals. Namely, if one paints the hexagons of a honeycomb lattice in two colors randomly with equal probabilities independently of each other, then the hex-curve approximates a random fractal known as CLE_6 (Miller et al., 2017). Patchwork parameters (21) can be easily modified to represent the hex-curve (before the affine transformation of the honeycomb lattice) and thus to sample the random CLE_6 fractal. Other random fractals arising from sampling random patchworks can also be of interest and are the subject of future work.

Another side remark is that the relation between the patchwork parameters and the tesselation by candidate polygons is most visual in the context of isotropic geometry and metric duality; see (Pirahmad et al., 2026, Section 2) for an introduction.

Now we proceed to the 3D case. Let us approximate a given surface $S \subset [-1, 1]^3$ by a patchwork. Take an integer N and replace the index i in (15) with a triple of indices $k, l, m \in [-N, N]$, so we write $a_{k,l,m}$ instead of a_i etc. Set

$$a_{k,l,m} = (k, l, m), \quad c_{k,l,m} = \frac{1 - k^2 - l^2 - m^2}{2N}, \quad s_{k,l,m} = \begin{cases} -1, & \text{if } \frac{(k,l,m)}{N} \text{ is inside } S, \\ +1, & \text{otherwise.} \end{cases} \tag{22}$$

The resulting candidate polygons are squares forming a cubic grid. The patchwork is the union of the squares separating cube midpoints inside and outside S. This union is called a digital surface. For small $T > 0$ and large N, the smoothed patchwork is close to the digital surface and hence to S. We arrive at the following folklore theorem.

Theorem 4 (Expressive power)**.** *Any compact embedded surface without boundary in $\mathbb{R}^3$ can be approximated by patchworks, that is, for each $\varepsilon > 0$, there exists a patchwork within the Hausdorff distance ε from the surface.*

Here, by a compact embedded surface without boundary, we mean the image of an injective continuous map of a compact two-dimensional manifold into $\mathbb{R}^3$. The manifold need not be connected, and the map need not be smooth.

Proof. It suffices to show that the Hausdorff distance between the initial surface S and the digital surface D tends to zero as $\delta \to 0$, because any digital surface is a patchwork with parameters (22). Consider the set S_δ of points inside S at a distance $\geq 2\delta$ from it. All those points lie in the cubes inside S; otherwise, there would be a cube of diameter $\sqrt{3}\,\delta$ intersecting both S and S_δ. All the points outside of S lie in the cubes, not inside S. Thus, D lies between S and S_δ, and the Hausdorff distance between S and D is not greater than the one between S and S_δ. The latter distance tends to zero as $\delta \to 0$: indeed, applying Dini's theorem to the functions $f_\delta \colon S \to \mathbb{R}$, $x \mapsto \operatorname{dist}(x, S_\delta)$, we conclude that $\max_{x \in S} f_\delta(x) \to 0$ as $\delta \to 0$. QED □

Table 6: A few lower bounds on the precision complexity M of a smoothed patchwork representing a given bounded curve. The x-projection degree D is the maximal number of intersections of the curve with a y-parallel line not contained in the curve. See Section E for details.

Curve complexity measure	**Notation**	**Bound**	**Underlying theorem**
x-projection degree	D	$M \geq D$	Bézout's theorem
number of components	C	$M \geq \sqrt{2C}$	Harnack's curve theorem

Table 7: A few lower bounds on the precision complexity M of a smoothed patchwork representing a given closed surface (having a smooth extension to complex projective space). The xy-projection degree D is the maximal number of intersections of the surface with a z-parallel line not contained in the surface. The number of handles G is the total genus of all the surface components.

Surface complexity measure		**Bound**	**Underlying theorem**
xy-projection degree	D	$M \geq D$	Bézout's theorem
number of components	C	$M \geq \sqrt[3]{2C}$	Generalized Harnack Inequality
number of handles	G	$M \geq \sqrt[3]{2(C+G)}$	(Itenberg & Viro, 2006) and Proposition 1

E Representation complexity

Now we address the question of the representation complexity of patchworks, that is, what complexity of patchworks is required to represent the shape of given complexity. Remarkably, for some complexity measures, we can achieve lower bounds (cf. problem of lower bounds in computational complexity theory).

In this work, we focus on a single toy example of a complexity measure to provide a proof of concept for patchwork complexity lower bounds. Namely, we study rough bounds on the numerical precision (the number of digits in the decimal representation) of the patchwork coefficients a_i and b_i (a_i in the 3D case) required to represent a given shape. So far, we have not attempted to bound the width n, which is a more natural but more complicated complexity measure for the analysis, to be addressed in future work.

Without loss of generality, we assume that all the coefficients a_i and b_i are integers, because we can always scale them by a common factor and absorb it into T, without changing the smoothed patchwork. We also assume that they are non-negative, because we can always add a common constant to them and add two auxiliary design lines preserving the smoothed patchwork. Then, introduce the patchwork precision complexity

$$M := \max_i \{a_i + b_i\}. \tag{23}$$

The value $\log_{10} M$ roughly measures the numerical precision of the coefficients before scaling and adding a constant.

Analogously, in 3D, denote $a_i = (a_{i,1}, a_{i,2}, a_{i,3})$, assume $a_{i,j} \in \mathbb{Z}_{\geq 0}$ for all i, j, and set

$$M := \max_i \{a_{i,1} + a_{i,2} + a_{i,3}\}. \tag{24}$$

In Tables 6–7, we collect a few lower bounds on M for a smoothed patchwork with $T > 0$ and all $a_i, b_i \in \mathbb{Z}_{\geq 0}$ (or $a_{i,j} \in \mathbb{Z}_{\geq 0}$) representing a given shape. They come from the observation that the exponential map $(x, y) \mapsto \left(e^{x/T}, e^{y/T}\right)$ takes the smoothed patchwork to an algebraic set of degree M, and the known theorems on the complexity of the latter.

Conversely, if for some $T > 0$, the exponential map takes a given set to an algebraic hypersurface of degree M (in the positive orthant), then the set is a smoothed patchwork of precision complexity M. This is a toy example of a patchwork complexity upper bound.

Specifically, in Tables 6–7, we use Bézout's and Harnack's theorems, and the following result.

Table 8: Baseline configurations at matched parameter counts for Roof Modeling (Tab. 10) / ABC&Thingi10k (Tab. 2) / Shapenet (Tab. 3) datasets.

Method	#Prims	#Params/Prims	#Layers	#Neurons	#Params
MC	20^3 / 51^3 / 81^3	1			8000 / 132651 / 531441
RFTA	20^3 / 51^3 / 81^3	1			8000 / 132651 / 531441
SPSR	1365 / 21845 / 87381	6			8190 / 131070 / 524286
PoNQ	512 / 8192 / 32768	16			8192 / 131072 / 524288
IGR			4	64 / 256 / 512	8641 / 132094 / 526334
SIREN			4	64 / 256 / 512	8641 / 132865 / 527873
VoroMesh	14^3 / 35^3 / 56^3	3			8232 / 128625 / 526848
Ours Init	2048 / 32768 / 131072	4			8192 / 131072 / 524288

Proposition 1 (Algebraic surface complexity bound)**.** *Consider a bounded algebraic surface of degree M in $\mathbb{R}^3$ with C components and the sum G of the genera of all components. Assume that the extension of the algebraic surface to complex projective space is smooth. Then*

$$M(M^2 - 4M + 6) \geq 2(C + G). \tag{25}$$

Proposition 1 follows from known results surveyed in (Itenberg & Viro, 2006), but the proof requires familiarity with basic topology and algebraic geometry.

Proof. Since the algebraic surface is bounded and non-singular, it is a disjoint union of finitely many closed compact orientable surfaces, and its projective closure in real projective space has the same topological type. Then the projective closure has the sum $C + 2G + C$ of the Betti numbers over $\mathbb{Z}/2\mathbb{Z}$. By the Generalized Harnack Inequality (Itenberg & Viro, 2006, Theorem 1.1), this sum is not greater than the analogous sum for the extension (Zariski closure) to the complex projective space. By the known computation (see e.g., (Yuan, 2014) or (Itenberg & Viro, 2006, p. 93)), the latter sum equals $M(M^2 - 4M + 6)$, and the desired inequality follows. QED □

F Future theoretical Work

Patchworks represent piecewise-linear shapes, and their smoothed versions are smooth. It is interesting to modify the construction to allow sharp corners and edges for non-piecewise-linear curves and surfaces. It is interesting to get bounds on the precision complexity and width of the patchwork representation of a shape; for instance, to determine if the maximal number of intersections of the shape with a line is a lower bound on both. Furthermore, we conjecture that the upper bound $O(1/\varepsilon^2)$ on the width described in Theorem 3 can be improved to $O(1/\varepsilon)$.

G Experiment details

G.1 Additional visual results

We provide extended visual comparisons for ABC (Fig. 14), Thingi10k (Fig. 15) and ShapeNet (Fig. 16) datasets.

G.2 Baseline configurations

The configurations of baselines and ours are summarized in Tab. 8.

Regular grid. Given an SDF grid, the de facto way to extract the underlying surface is Marching Cubes (MC) (Lorensen & Cline, 1987), but it overlooks the SDF's spherical property (Sellán et al., 2023), underutilizing the stored information. Reach for the Arc (RFTA) (Sellán et al., 2024) explicitly exploits this property and achieves the state-of-the-art performance at recovering high-quality surfaces from discrete grids. We compare MC and RFTA on SDF grids computed from ground-truth mesh data. We count 1 float per grid cell, and round the grid resolution to match the budget.

Point set. Given oriented point clouds, the axiomatic method, Screened Poisson Surface Reconstruction (SPSR) (Kazhdan & Hoppe, 2013), fits an implicit indicator function and then extracts its

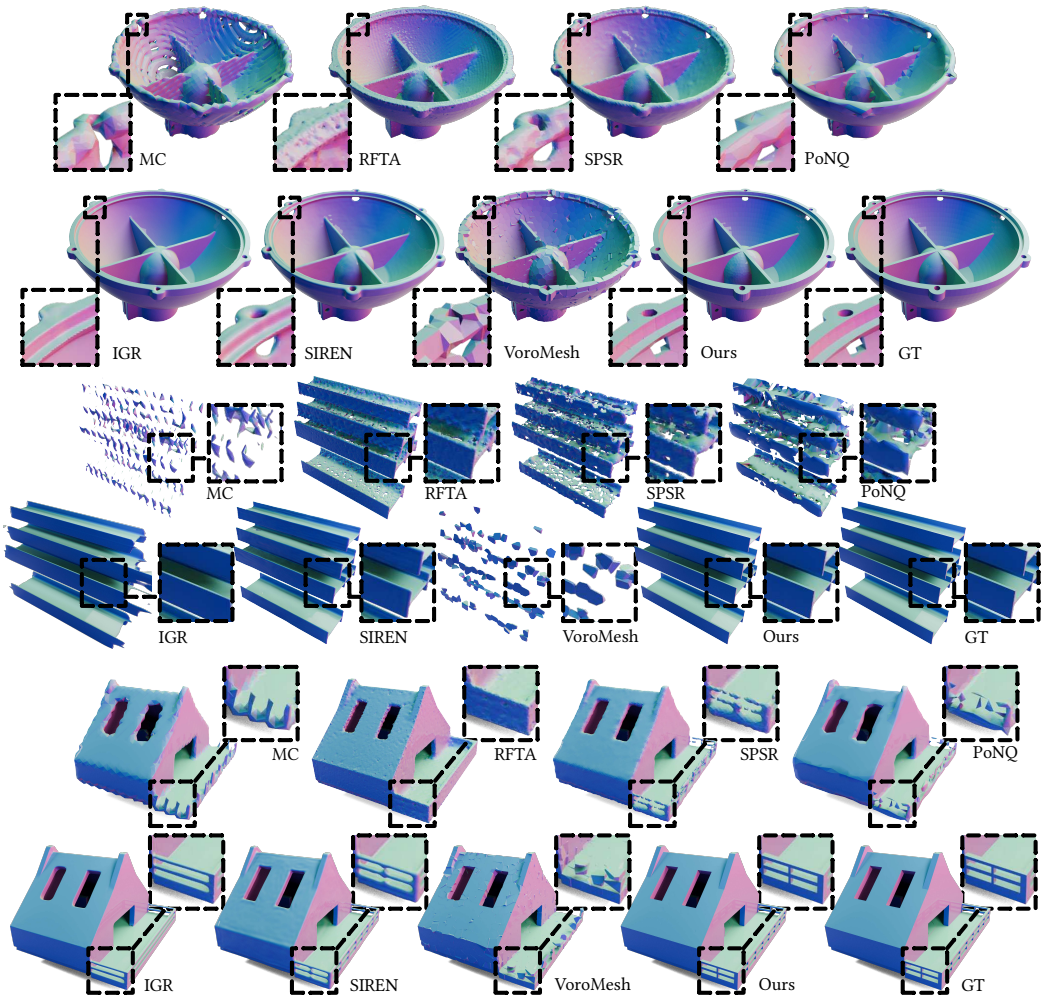


Figure 14: Extended visual results on ABC (Koch et al., 2019).

isosurface. PoNQ (Maruani et al., 2024) associates each oriented sample with a local $\mathbb{R}^{4\times4}$ quadric matrix to guide triangulation for better sharp-edge and topology handling. We count the number of floats per budget, that is, 6 for SPSR (point and normal in $\mathbb{R}^3$) and 16 for PoNQ (point and normal in $\mathbb{R}^3$, symmetric 4×4 matrix stored in 10 floats), then round the point set count to match the budget.

Neural implicit. Neural implicit representations use coordinate MLPs to encode SDFs as continuous functions, hence requiring either an (approximately) piecewise linear activation function or high expressiveness to model discrete jumps such as sharp edges. We compare IGR (Gropp et al., 2020), which pairs Softplus activations of large beta with Eikonal regularization to better model piecewise-smooth geometries. We also compare SIREN (Sitzmann et al., 2020), which leverages sine activations to model high-frequency signals. For simplicity, we fix the number of layers to 4, ignore the input layer parameter count, and round widths to conventional choices.

Foam-based. VoroMesh (Maruani et al., 2023) fits double covering point clouds as Voronoi generators and extracts crossing boundaries as reconstructed meshes. Since they initialize generators using voxel centers, we count 3 floats per grid cell and then round the grid resolution.

G.3 Scaling comparison

We select 10 cases with visually identifiable outer surfaces from the GeomDist (Zhang et al., 2025) shape collection, and fit with $N = 655360/1048576$, that is, $5242880/8388608$ parameters at initialization (abbreviated as 5M and 8M). We set batch size $K = 65536$, and the internal scale $\tau = 8$.

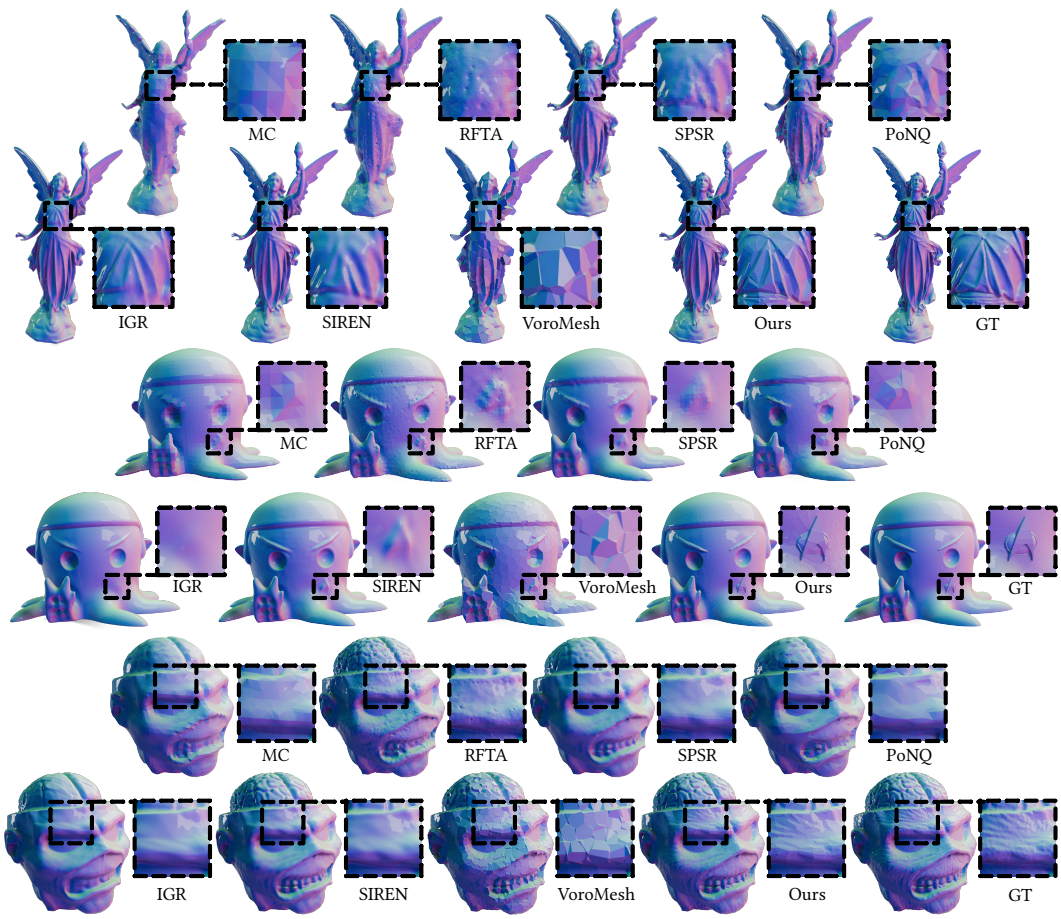


Figure 15: Extended visual results on Thingi10k (Zhou & Jacobson, 2016).

Table 9: Quantitative results of GeomDist (Zhang et al., 2025) dataset. The bold text indicates the best score.

	Chamfer Distance ($\times 10^{-3}$) $\downarrow$			**# Parameters**		
	GeomDist	Ours (5M)	Ours (8M)	GeomDist	Ours (5M)	Ours (8M)
Archimedes	3.301	2.168	**2.021**	5532680	671144	956492
Mouse	4.008	3.314	**2.616**	5532680	780768	1037680
Parthenon	3.760	2.335	**2.298**	5532680	864324	1079340
Archer	3.484	2.170	**2.150**	5532680	627688	925668
Jellyfish	4.832	2.680	**2.629**	5532680	919388	1026312
Lamp	3.754	2.268	**2.266**	5532680	992940	1143804
Warrior	2.774	1.723	**1.718**	5532680	543332	807084
Loong	**3.706**	4.440	4.425	5532680	749640	937172
Wukong	**4.412**	17.87	18.87	5532680	722716	905676
Lion	**4.017**	50.01	50.05	5532680	662820	919940

Each $10K$-iteration of fitting takes about 18/26 minutes and uses 644/812 MB of VRAM on a single NVIDIA RTX 3090 GPU with our KDTree variant. For GeomDist, we use the official checkpoints. Following their implementation, we measure Chamfer distance over 10M samples with respect to GT surfaces.

Compared to GeomDist, which requires over 40 hours on 4 A100 GPUs (2.5 min/epoch × 1000 epochs) to fit a single shape, our approach is significantly more efficient and offers additional ben-

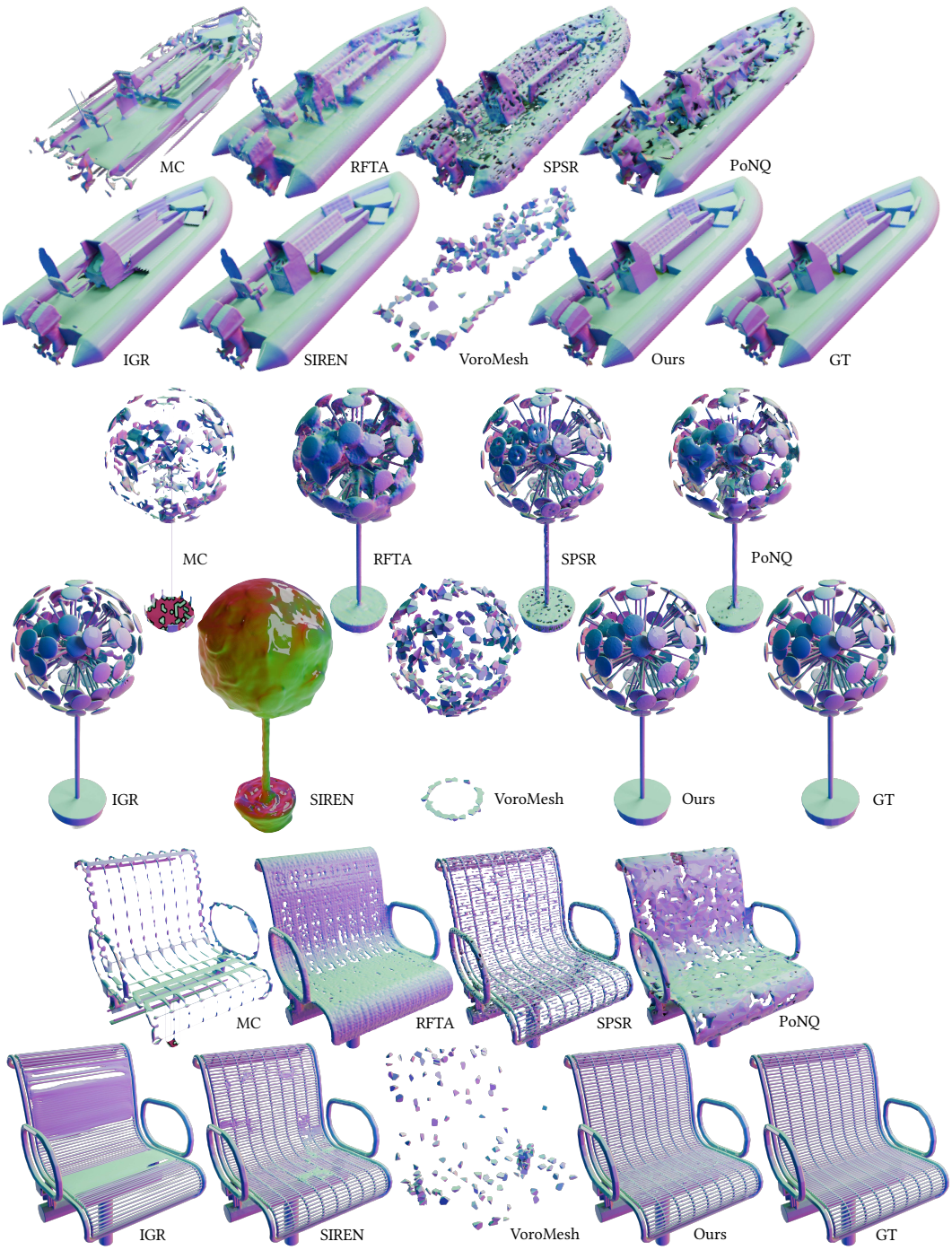


Figure 16: Extended qualitative results on ShapeNet (Chang et al., 2015).

efits of an implicit form and explicit polygonalization. As shown in Fig. 17 and Tab. 9, our model captures faithful global geometry and slightly outperforms GeomDist in mean Chamfer Distance at a matched parameter count for 7 of 10 shapes, with accuracy improving further as parameters increase. However, our formulation cannot handle self-intersecting / non-manifold interior structures, resulting in large Chamfer Distance in the bottom three cases of Tab. 9. See Fig. 17 (bottom right) for visualization.

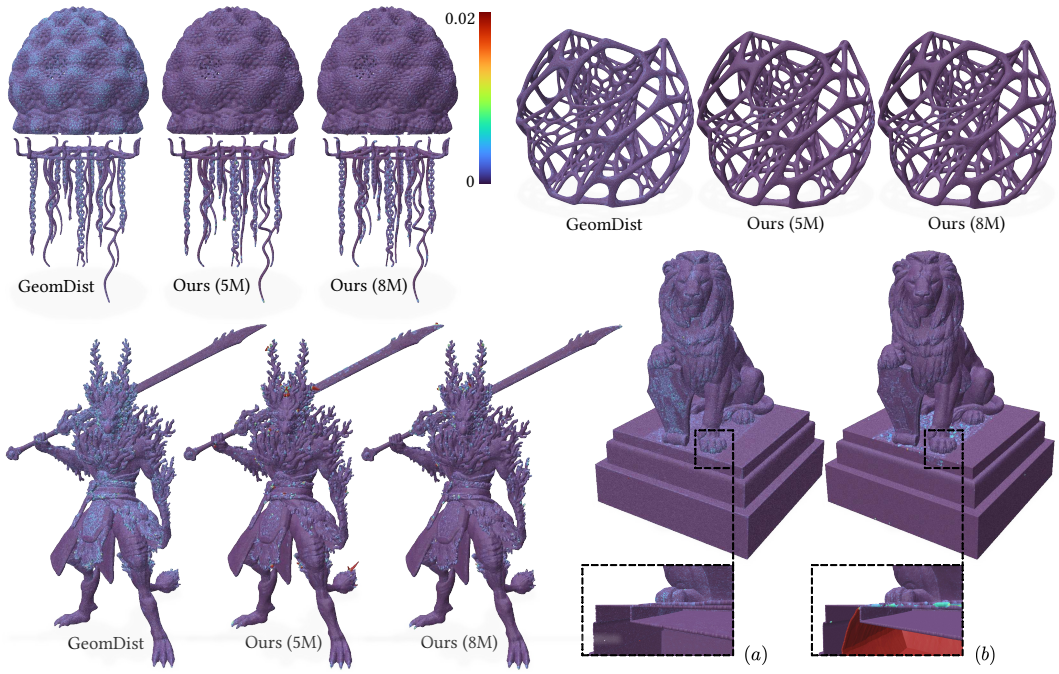


Figure 17: Qualitative results of Jellyfish (top left), Lamp (top right), Loong (bottom left), with 10M samples colored by error map using Polyscope (Sharp et al., 2019). As illustrated by the Lion case (Bottom right), although the exterior surface is well represented compared to GeomDist (a), the interior error is large because we cannot handle the topology of self-intersecting structures (b).

To explore extreme scaling, we fit the Wukong case with $N = 16777216$ (i.e., over 134M parameters at initialization), $K = 65536$ and $\tau = 8$ for $10K$ iterations. It takes 6 hours with 6830MB VRAM to fit on a single NVIDIA RTX 3090 GPU. The final parameter count is 10215668, and the result is visualized in Fig. 1 (Right).

G.4 Additional results on Roof Modeling

To evaluate patchwork's expressiveness on low-complexity geometry, we additionally compare on the Roof modeling (Ren et al., 2021) dataset, which consists of 3K unique building roof meshes that are geometrically simple yet rich in sharp creases. We randomly select 200 cases, set $N = 1024$, $K = 16384$ and τ to 0.25. Our results are slightly better than VoroMesh in metric terms (Tab. 10), but visually cleaner and much closer to the ground truth (Fig. 18).

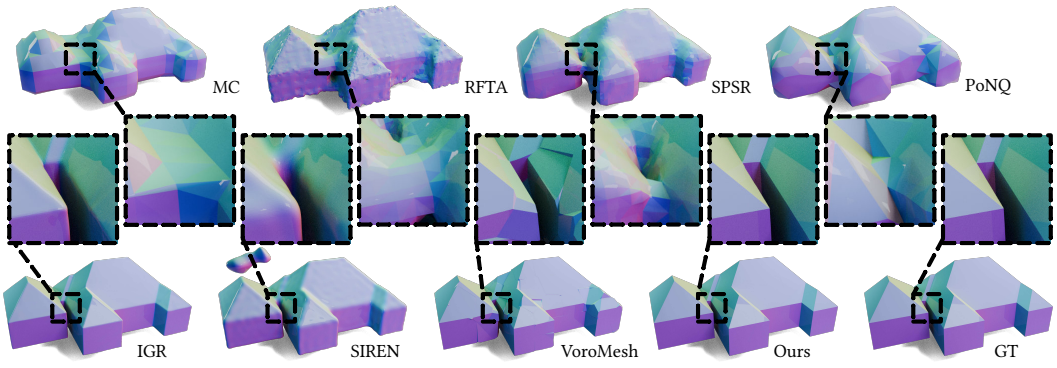


Figure 18: Qualitative results on Roof Modeling (Ren et al., 2021) dataset.

Table 10: Quantitative results on the Roof Modeling dataset. The bold text indicates the best score; the underlined text indicates the second-best. 'Ours #Params' is the average parameter size after pruning.

Method	CH ↓	HD ↓	FS ↑	NC ↑	IoU ↑	#Params
MC	6.79	8.68	49.93	94.65	95.89	8000
RFTA	3.63	5.95	36.41	96.98	97.75	8000
SPSR	4.57	5.88	25.86	96.95	97.21	8190
PoNQ	3.74	9.65	62.05	96.73	98.08	8192
IGR	1.40	<u>1.19</u>	77.21	<u>99.70</u>	99.62	8641
SIREN	21.2	74.0	53.79	94.05	91.02	8641
VoroMesh	<u>1.29</u>	2.05	<u>80.45</u>	99.44	<u>99.92</u>	8232
Ours init	5.90	20.0	58.65	88.03	96.84	8192
Ours	**1.21**	**0.66**	**82.16**	**99.74**	**99.98**	2195

H Explicit Polygonalization

Our method supports extracting the patchwork via explicit polygonalization. For each <u>interior</u> candidate polygon/polyderon (i.e., corresponding to an index i with $s_i < 0$), compute the equality lines/planes $\langle a_i, x\rangle + c_i = \langle a_j, x\rangle + c_j$ for all $j = 1, \ldots, n$ (i.e., the halfspaces that bound the current candidate polygon). Then find its Chebyshev center x by maximizing y over all pairs (x, y) satisfying the constraint

$$\langle a_j - a_i, x\rangle + y||a_j - a_i||_2 + c_j - c_i \leq 0 \qquad \text{for each } j = 1, \ldots, n. \tag{26}$$

If the resulting $y > 0$, then compute the intersections of those halfspaces using convex hull. Then we determine connectivity via the secondary convex hull computation, and only keep faces shared by other exterior candidate polygons/polydera. After performing the same for all interior candidate polygons/polydera, we merge all active segments/polygons for the final patchwork extraction. This extraction procedure is embarrassingly parallel, except for the final merging step.

As mentioned in Sec. 6, some extruding interior polyhedra may survive the pruning due to erroneous normal info. After explicit polygonalization, they extend far beyond the typical Marching Cubes extraction range, yielding extreme outliers that skew the metric statistics. To address this, we compute the log-diagonal of the bounding box for each interior convex cell and discard those exceeding the third quartile by more than three times the interquartile range (Tukey, 1977).

H.1 Edge collapsing routine

During extraction, for each interior convex cell, we compute the area of all polygonal faces, flag those with area less than 10% of the maximal one, and collect their shared candidate polyhedral indices in parallel. For all faces flagged for deletion, we count the unique indices for each vertex and collapse edges shared by faces with different labels by merging the vertex with the lower index count into the one with the higher index count. Finally, we perform the standard quadric edge collapsing (Garland & Heckbert, 1997) to dissolve the remaining flagged faces.

We find this routine highly effective for dissolving artifacts unique to foam-based representation. Its effectiveness is currently bounded by our heuristic face flagging, and we expect future learning-based methods to better identify faces for deletion. Note that we don't apply this routine for metrics computation to ensure fairness.

I Implicit Fitting Formulation

In this section, we describe one variant of the approach that fits patchwork directly as a standard implicit function. This is the variant we tune as best we can, and we use it for an ablation study of our discrete matching scheme.

I.1 FORMATION

We consider patchwork in the following augmented form:

$$\begin{aligned} F(x) = &\frac{1}{\beta^+} \log \sum_{i=1}^{n^+} \exp\left(\beta^+(\langle a_i^+, x\rangle + c_i^+) + \log s_i^+\right) \\ &- \frac{1}{\beta^-} \log \sum_{i=1}^{n^-} \exp\left(\beta^-(\langle a_i^-, x\rangle + c_i^-) + \log s_i^-\right), \end{aligned} \tag{27}$$

where $n^+ + n^- = n$, $a_i^\pm \in \mathbb{R}^3$, $c_i^\pm \in \mathbb{R}$, $s_i^\pm, \beta^\pm \in \mathbb{R}_{>0}$. Note that $s_i^\pm$ can be losslessly incorporated into $c_i^\pm$, and we keep them here only for pruning purposes.

Given an oriented point cloud $\{x_j, n_j\}_{j=1}^m$, where $x_j \in \mathbb{R}^3$ are points and $n_j \in \mathbb{R}^3$ are normal vectors. We aim to fit $F(x)$ such that its zero level set $\Omega_0 = \{x \in \mathbb{R}^3 | F(x) = 0\}$ describes the underlying geometry. To this end, we minimize the surface-approximation error and the normal-misalignment penalty:

$$\mathcal{L}_{\text{sur}} = \frac{1}{m}\sum_{j=1}^m |F(x_j)| \quad \text{and} \quad \mathcal{L}_{\text{normal}} = \frac{1}{m}\sum_{j=1}^m \left(1 - \frac{\nabla F(x_j)\cdot n_j}{\|\nabla F(x_j)\|\|n_j\|}\right). \tag{28}$$

To reduce off-surface parts of Ω_0, we additionally sample m off-surface samples $\{y_k\}_{k=1}^m, y_k \in \mathbb{R}^3 \setminus \Omega_0$ per iteration for a soft occupancy regularization

$$\begin{aligned} g_{dw}(x) &= 4(x-0.5)^2 - 4|x-0.5| + 1, \\ \mathcal{L}_{\text{reg}} &= \frac{1}{m}\sum_{k=1}^m g_{dw}(\text{sigmoid}(-F(y_k)))^2, \end{aligned} \tag{29}$$

where g_{dw} is the double-well potential (Lipman, 2021) that attains minimum at $x = 0, 1$.

The regularization $\mathcal{L}_{\text{reg}}$ additionally forces some candidate polyhedra to expand, leaving the other ones redundant. We hence introduce $\mathcal{L}_{\text{prune}}$ in Sec. I.3 to progressively eliminate unnecessary ones during the fitting process.

Our total losses are

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sur}} + \mathcal{L}_{\text{normal}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{prune}}.$$

I.2 EXPLICIT GEOMETRIC INITIALIZATION

From an optimization point of view, the Patchwork falls into the broad category of Difference-of-Convex (DC) problems. Specifically, its limiting form is the difference of two convex functions. Inspired by Piecewise Linear Regression (PLR) (Siahkamari et al., 2020) in DC literature, we assign each input point x_j two linear functions (i.e., set $n^+ = n^- = m$) and force the two maxima at x_j to be attained with their corresponding coefficients $\{a_j^+, c_j^+, a_j^-, c_j^-\}$. Formally,

$$\begin{aligned} &f(x) = \max_i\{\langle a_i^+, x\rangle + c_i^+\} - \max_i\{\langle a_i^-, x\rangle + c_i^-\}, \\ &\arg\max_i\{\langle a_i^+, x_j\rangle + c_i^+\} = \arg\max_i\{\langle a_i^-, x_j\rangle + c_i^-\} = j. \end{aligned} \tag{30}$$

The two pointwise global-maximum constraints significantly reduce the solution space, yet the formulation remains a DC, a universal approximator.

To further reduce the solution space, we assume $f(x)$ at the initialization approximates the signed distance function. Inspired by proximal DC decomposition (Sun et al., 2024), we perform the following construction

$$h^+(x) = \tfrac{1}{2}\rho\|x\|^2 + \text{sdf}(x),\ h^-(x) = \tfrac{1}{2}\rho\|x\|^2,\ f(x) \approx h^+(x) - h^-(x), \tag{31}$$

where $\rho \in \mathbb{R}$ is some parameter large enough to make $h^+(x)$ convex.

Conviently, with oriented surface samples $\{x_j, n_j\}$, we can set $f(x_j) = \text{sdf}(x_j) = 0$ and $\nabla f(x_j) = \nabla \text{sdf}(x_j) = n_j$, yeilding the closed-form solution

$$\begin{aligned} a_j^+ &= \nabla h^+(x_j) = \rho x_j + n_j, \quad & c_j^+ &= h^+(x_j) - \langle a_j^+, x_j \rangle = -\tfrac{1}{2}\rho \|x_j\|^2 - \langle n_j, x_j \rangle, \\ a_j^- &= \nabla h^-(x_j) = \rho x_j, & c_j^- &= h^-(x_j) - \langle a_j^-, x_j \rangle = -\tfrac{1}{2}\rho \|x_j\|^2. \end{aligned} \tag{32}$$

We defer the detailed derivation to Appendix I.4. In practice, we initialize with $\rho = 200$, $s_i^\pm = 1$, $\beta^\pm = 75$ across all our experiments. Notice the norm of $a_j^\pm$ scales with ρ, making updating its direction stiff. So we apply weight normalization to disentangle its magnitude and direction gradient.

I.3 Progressive pruning

We propose the following loss to progressively eliminate non-contributing linear functions

$$\begin{aligned} w_i^\pm(x) &= \text{sg}\left(\frac{\beta^\pm(\langle a_i^\pm, x \rangle + c_i^\pm) + \log s_i^\pm}{\sum_{l=1}^{n^\pm} \left(\beta^\pm(\langle a_l^\pm, x \rangle + c_l^\pm) + \log s_l^\pm\right)} \right), \\ \mathcal{L}_{\text{prune}}^\pm &= \frac{1}{n^\pm} \sum_{i=1}^{n^\pm} s_i^\pm + \frac{1}{m} \sum_{j=1}^{m} \text{ReLU}\left(1 - \sum_{i=1}^{n^\pm} s_i^\pm \cdot w_i^\pm(x_j)\right), \\ \mathcal{L}_{\text{prune}} &= \mathcal{L}_{\text{prune}}^+ + \mathcal{L}_{\text{prune}}^-, \end{aligned} \tag{33}$$

where sg is the stop gradient operator, and $w_i^\pm$ can be viewed as a typical soft maximum. The first term encourages all $s_i^\pm$ to be diminished, while the second term maintains the magnitude of $s_i^\pm$ that contributes to candidate polygons.

For every fixed iterations, we detect linear functions with $s_i^\pm < 10^{-2}$, and disable them softly by setting $s_i^\pm = 10^{-5}$.

I.4 Initialization derivation

Let us derive geometric initialization (32) from assumptions (30). We start with introducing the functions

$$f^\pm(x) = \max_i \{\langle a_i^\pm, x \rangle + c_i^\pm\}. \tag{34}$$

By our assumptions (30), the maximum for $x = x_j$ is attained at $i = j$. Then it is natural to assume that for x sufficiently close to x_j, we still have

$$f^\pm(x) = \langle a_j^\pm, x \rangle + c_j^\pm \tag{35}$$

and hence

$$\nabla f^\pm(x) = a_j^\pm. \tag{36}$$

Substituting $x = x_j$, we get

$$f^\pm(x_j) = \langle a_j^\pm, x_j \rangle + c_j^\pm \qquad \text{and} \qquad \nabla f^\pm(x_j) = a_j^\pm. \tag{37}$$

Recall definition (31) of convex functions

$$h^+(x) = \tfrac{1}{2}\rho \|x\|^2 + \text{sdf}(x) \qquad \text{and} \qquad h^-(x) = \tfrac{1}{2}\rho \|x\|^2. \tag{38}$$

To satisfy

$$f(x_j) = \text{sdf}(x_j) = 0 \qquad \text{and} \qquad \nabla f(x_j) = \nabla \text{sdf}(x_j) = n_j, \tag{39}$$

it is natural to require the conditions

$$f^\pm(x_j) = h^\pm(x_j) \qquad \text{and} \qquad \nabla f^\pm(x_j) = \nabla h^\pm(x_j). \tag{40}$$

Comparing with equation (37), we get the desired closed-form solution (32):

$$\begin{aligned} a_j^+ &= \nabla h^+(x_j) = \rho x_j + n_j, \quad & c_j^+ &= h^+(x_j) - \langle a_j^+, x_j \rangle = -\tfrac{1}{2}\rho \|x_j\|^2 - \langle n_j, x_j \rangle, \\ a_j^- &= \nabla h^-(x_j) = \rho x_j, & c_j^- &= h^-(x_j) - \langle a_j^-, x_j \rangle = -\tfrac{1}{2}\rho \|x_j\|^2. \end{aligned} \tag{41}$$

The resulting solution automatically satisfies assumptions (30): The plot of linear function (35) is the tangent plane to the plot of the convex function $h^\pm(x)$ at the point $x = x_j$. Thus, the maximum in (34) for $x = x_j$ is attained at $i = j$.