PATCHGF: FEED-FORWARD GEOMETRIC FIELDS FROM UNORIENTED POINT CLOUDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Surface reconstruction and differential-quantity estimation from unoriented point clouds are often addressed separately, although both depend on the same local geometry. We present PatchGF, a feed-forward framework that jointly predicts patch-conditioned unsigned distance fields (UDFs), unoriented normals, mean-curvature magnitudes, and Gaussian curvatures. A shared Point Transformer encodes each patch; query-to-patch cross-attention predicts unsigned distances at nearby spatial locations, while dedicated heads estimate normals and curvatures. A single model is trained exclusively on synthetic quadratic patches and applied directly to unseen point clouds without fine-tuning or per-shape network optimization. Paired queries sampled along analytic normals, with curvature-adaptive offsets, supervise near-surface distance variation. At inference, fixed anchor-centered patches provide consistent local supports, and distance-weighted fusion combines overlapping predictions into a UDF over the observed surface region. Positive-level-set extraction and gradient-based projection provide an initial reconstruction, which can be refined through local surface fitting and adaptive remeshing guided by the predicted differential quantities. Experiments on standard benchmarks and real-world scans evaluate reconstruction and differential-quantity estimation and demonstrate accurate reconstruction under noise and outlier contamination.

1 INTRODUCTION

Unoriented point clouds provide a simple representation of scanned geometry, but they contain neither explicit surface connectivity nor a consistent distinction between the two sides of a surface. Reconstructing a mesh from these samples supports visualization, geometric analysis, and other downstream applications. Neural implicit representations, such as DeepSDF (Park et al., 2019) and Occupancy Networks (Mescheder et al., 2019), describe surfaces through continuous functions, but their signed-distance and occupancy formulations are primarily suited to closed surfaces. Unsigned distance fields (UDFs) remove the need for an inside–outside distinction and can represent surfaces with boundaries and non-manifold structures. Learning and extracting these fields remain challenging because an exact UDF is non-differentiable on the surface and does not change sign across it.

Existing UDF methods differ in how they learn geometric priors and perform inference. Shape-specific methods, including CAP-UDF (Zhou et al., 2024), LevelSet-UDF (Zhou et al., 2023), DUDF (Fainstein et al., 2024), and DEUDF (Xu et al., 2025), optimize a neural field for each input point cloud. This approach can recover fine details but incurs substantial test-time computation and often requires balancing several regularization terms. Methods trained on collections of complete shapes, such as NDF (Chibane et al., 2020) and GIFS (Ye et al., 2022), amortize inference across inputs but require suitable training data and supervision. A complementary approach learns unsigned distances from local point neighborhoods (Ren et al., 2023; Hu et al., 2025). Near a surface, distance depends on the geometry around the closest point, motivating local representations that support feed-forward inference on unseen shapes.

Beyond distance prediction, surface reconstruction benefits from reliable differential quantities. Normals describe local surface orientation, while curvature can guide mesh refinement to preserve detail without excessive tessellation. Accurate UDF values alone do not ensure accurate normals or curvatures, and the non-differentiability of an exact UDF at the surface complicates their recovery through differentiation. DUDF (Fainstein et al., 2024) derives these quantities from derivatives

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.

of a transformed distance field, adding derivative evaluation to its per-shape optimization. LoSF-UDF (Hu et al., 2025) enables efficient local distance prediction but does not explicitly predict differential quantities. These observations motivate a shared local representation that predicts unsigned distances, unoriented normals, and curvatures and uses them together for adaptive surface reconstruction.

We introduce **PatchGF**, a feed-forward framework for learning local geometric fields from unoriented point clouds. PatchGF predicts a patch-conditioned UDF in nearby space and estimates differential quantities from the same encoded patch. The model is trained entirely on synthetic quadratic patches, without a shape-level UDF training corpus. We generate paired queries along analytic surface normals and adapt their offsets to the curvature scale, providing spatially distributed supervision of near-surface UDF variation. A Point Transformer (Zhao et al., 2021) encodes each patch, and query-to-patch cross-attention predicts distances at arbitrary nearby locations. A pointwise normal head combines point features with pooled patch context, while a curvature head predicts mean-curvature magnitude and Gaussian curvature at the patch center. These differential quantities are predicted directly, without second-order differentiation of the learned UDF.

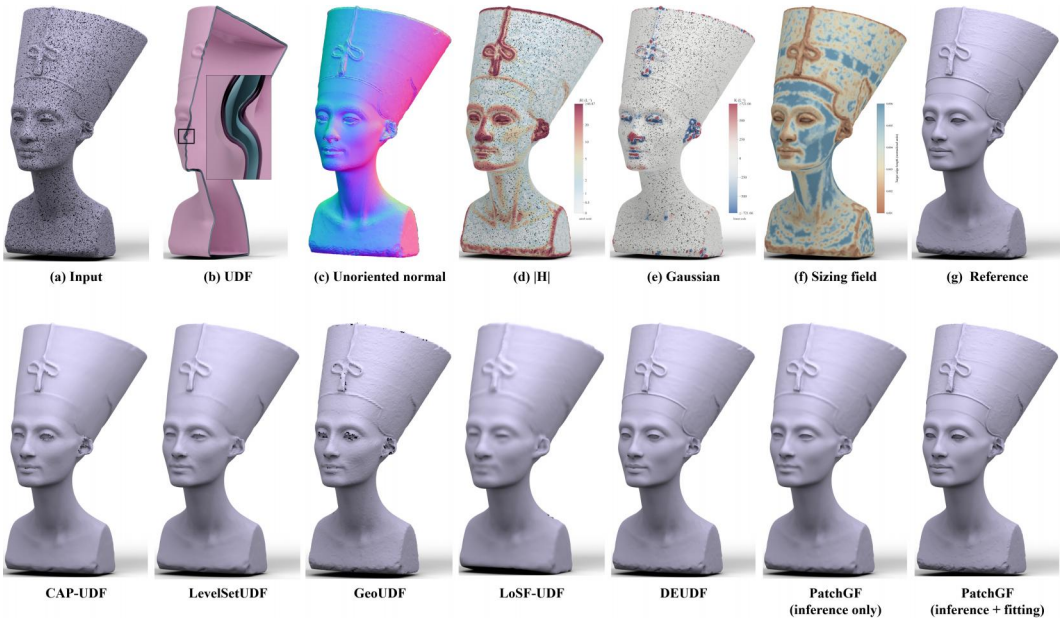
At inference, farthest-point sampling defines fixed, overlapping anchor-centered patches. Nearby spatial queries are evaluated against these fixed supports, and distance-weighted fusion combines overlapping predictions. Queries associated with the same anchor therefore share the same local geometric context. Reconstruction begins with positive-level-set extraction and projection toward the estimated surface. For detailed reconstruction, predicted normals and curvatures guide adaptive extraction, local surface fitting, and remeshing: normals define local fitting frames, curvature informs the fitting supports, and a curvature-derived sizing field allocates mesh resolution. All network predictions are obtained without per-shape parameter optimization (see Figure 1).

Our main contributions are: (1) A shared patch representation that predicts unsigned distances, unoriented point normals, and orientation-invariant curvatures through dedicated prediction heads. (2) A synthetic supervision strategy that uses paired normal-offset queries with curvature-adaptive sampling to learn distance variation near the surface. (3) A reconstruction pipeline that combines fixed-patch field assembly with adaptive extraction, local fitting, and remeshing guided by the predicted differential quantities.

2 Related Work

Signed implicit fields encode an inside–outside distinction and are well suited to representing closed manifold surfaces. Classical approaches include global formulations, such as Poisson surface reconstruction (Kazhdan et al., 2006) and its variants (Kazhdan & Hoppe, 2013; Hou et al., 2022), and local constructions, such as MPU (Ohtake et al., 2003), which blends quadratic surface approximations. Neural approaches parameterize implicit fields using networks. DeepSDF (Park et al., 2019) learns signed distance functions conditioned on shape codes, while Occupancy Networks (Mescheder et al., 2019) predict whether spatial queries lie inside an object. Other methods learn a field directly from an individual unoriented point cloud. Neural-Pull (Ma et al., 2021) uses predicted distances and field gradients to pull spatial queries toward observed surface points, supervising the field through the resulting positional error. NSH (Wang et al., 2023) introduces a singular-Hessian regularizer near the surface to suppress spurious geometry and improve reconstruction. Local neural representations distribute geometric information across spatial regions or surface patches. DeepLS (Chabra et al., 2020) represents geometry through a collection of local signed distance functions decoded from local latent codes. PatchNets (Tretschk et al., 2020) uses learned surface patches with associated positions, orientations, and scales. CoFie (Jiang et al., 2024) learns local coordinate frames that align surface neighborhoods, simplifying the geometry presented to the implicit decoder. These decompositions allow each local function to describe a smaller portion of the surface and provide reusable local shape priors.

Unsigned implicit fields , in particular, unsigned distance fields, can represent surfaces of arbitrary topology, including open surfaces, non-manifold surfaces and non-orientable surfaces. However, their non-differentiability on the surface and lack of sign changes, complicate field learning and mesh extraction. Several methods address these challenges through optimization on individual point clouds. CAP-UDF (Zhou et al., 2024) builds on Neural-Pull (Ma et al., 2021) to learn un-



CAP-UDF	LevelSetUDF	GeoUDF	LoSF-UDF	DEUDF	PatchGF (inference only)	PatchGF (inference + fitting)
CD: 1.9952	CD: 1.1196	CD: 0.9924	CD: 1.6930	CD: 0.5408	CD: 1.0849	CD: **0.3284**
HD: 8.0769	HD: 8.0817	HD: 5.5482	HD: 8.3118	HD: 6.0814	HD: 6.6573	HD: **5.4524**
NC: 0.9871	NC: 0.9879	NC: 0.9880	NC: 0.9727	NC: 0.9924	NC: 0.9902	NC: **0.9945**
Time: $>$ 30 m	Time: $>$ 30 m	Time: **0.65 m**	Time: 0.82 m	Time: 24.55 m	Time: 2.30 m	Time: 11.60 m

Figure 1: Joint geometric prediction and surface reconstruction with PatchGF. Given an unstructured point cloud of 300,000 samples (a), PatchGF predicts unsigned distance (b), unoriented normals (c), mean curvature magnitudes (d) and Gaussian curvatures (e). The predicted differential quantities guide local surface fitting and an adaptive sizing field (f), allocating finer mesh resolution to strongly curved regions to preserve geometric detail. The reference surface is shown in (g). The bottom row compares reconstructions and runtimes with representative UDF-based methods. PatchGF can infer fairly efficient, with 2.30 minutes, to predict the entire geometry fields. GeoUDF and LoSF-UDF have faster inference speeds, but they only predict unsigned distances. Together with adaptive surface fitting, PatchGF achieves the lowest distance errors and highest normal consistency in this example. Chamfer distances (CD) and Hausdorff distances (HD) are reported in units of 10^{-2}; NC denotes normal consistency, and runtimes are in minutes. Our runtime includes both the neural network inference time and per-shape fitting, with the overall runtime still better than the unsupervised learning methods such as CAP-UDF, LevelSetUDF and DEUDF. Best values are highlighted in bold.

signed distances through consistency-aware point pulling. LevelSetUDF (Zhou et al., 2023) uses level-set projection to improve the learned field near its zero level set. DUDF (Fainstein et al., 2024) applies hyperbolic scaling to obtain a differentiable distance representation, which it learns through a modified Eikonal formulation. PrimitiveUDF (Deng et al., 2026) guides attention-based unsigned distance prediction using spherical projection and geometric primitives, improving surface reconstruction of thin, closely-spaced structures. Although these methods reduce direct differentiation at the zero level set, recovering fine geometric details remains challenging. DEUDF (Xu et al., 2025) avoids explicit non-negativity constraints and introduces an objective designed to preserve geometric detail. Supervised learning methods, such as GeoUDF (Ren et al., 2023) and LoSF-UDF (Hu et al., 2025), learn to predict unsigned distances from local point neighborhoods, enabling feed-forward reconstruction of unseen shapes without per-shape network optimization.

Other implicit representations. Alternative representations encode geometry beyond conventional distance fields. SALS (Ren & Hou, 2025) predicts whether and where spatial line segments intersect a surface, then extracts a mesh using edge-based dual contouring. MPF (Kong et al., 2026) separates distance magnitude from sign by learning an unsigned metric field and a smooth phase

field. Both approaches reconsider how geometry is encoded to improve surface reconstruction and extraction.

Normal estimation. Classical normal estimators infer local surface orientation from point neighborhoods. PCA-based methods estimate a tangent plane from the local covariance matrix (Hoppe et al., 1992), with adaptive variants selecting suitable neighborhood sizes (Mitra et al., 2004). Other approaches exploit Voronoi geometry (Amenta & Bern, 1998; Alliez et al., 2007; Mérigot et al., 2011) or vote over candidate directions using a randomized Hough transform (Boulch & Marlet, 2012). Learned estimators infer normals from local point patches, including PCPNet (Guerrero et al., 2018), Nesti-Net (Ben-Shabat et al., 2019), and subsequent methods using learned geometric features and local surface models (Lenssen et al., 2020; Li et al., 2022b; 2023c; Wu et al., 2024; Xiu et al., 2023; Li et al., 2025a;b; 2024). Normals can also be obtained from learned field gradients (Li et al., 2023a;b). PatchGF uses a dedicated normal head whose supervision treats opposite normal directions as equivalent.

Curvature estimation. Curvature describes how a surface bends and is particularly sensitive to noise and sampling density. Classical estimators use local fitting and geometric or statistical constructions (Kalogerakis et al., 2007; Lejemble et al., 2021; Cazals & Pouget, 2005; Lachaud et al., 2023; Béarzi et al., 2018). Learning-based fitting methods, including DeepFit (Ben-Shabat & Gould, 2020), AdaFit (Zhu et al., 2021), and GraphFit (Li et al., 2022a), improve local estimation through learned point weights or offsets. OscuFit (Fu et al., 2026) fits osculating implicit quadrics in a canonical local frame to estimate normals and curvatures jointly. PatchGF directly predicts mean-curvature magnitude and Gaussian curvature from shared patch features and uses these estimates to guide fitting neighborhoods and mesh resolution.

3 Method

3.1 Overview

PatchGF reconstructs surfaces from unoriented point clouds by learning local patch geometry. Given a patch $\mathcal{P}$ centered at $\mathbf{p}_0$, it predicts unsigned distances at nearby spatial queries, unoriented normals at the input points, and mean-curvature magnitude $|H|$ and Gaussian curvature K at the patch center[1]. These complementary predictions share a common patch representation: distances locate the surface, normals describe its local orientation, and curvatures characterize its bending.

We train PatchGF exclusively on synthetic quadratic patches and apply the trained model directly to unseen point clouds, without fine-tuning or per-shape network optimization. At inference, we select fixed anchors, construct overlapping local patches, and combines their UDF predictions at nearby queries through distance-weighted averaging. We then extract a positive level set and project its vertices toward the surface in one step using the predicted distances and UDF gradients. For reconstructing fine details, the predicted normals and curvatures further guide adaptive extraction, local surface fitting, and remeshing.

3.2 Synthetic Training Patches and Supervision

Quadratic patches as local surface models. Near a smooth interior point, a surface can be approximated to second order by a quadratic height function over its tangent plane. Choosing the principal curvature directions as the local axes gives $z = \frac{1}{2}\left(\kappa_1 u^2 + \kappa_2 v^2\right)$, where u and v are tangent-plane coordinates, and κ_1 and κ_2 are the principal curvatures at the patch center. Varying these two curvatures produces flat, cylindrical, elliptic, and saddle-shaped configurations. We sample points by evaluating this function at locations within a disk in the tangent plane. The resulting patches provide analytic surface normals and center-curvature labels, with $|H| = |\kappa_1 + \kappa_2|/2$ and $K = \kappa_1\kappa_2$. This construction provides supervision for all prediction branches without requiring complete training shapes.

[1] Reversing the surface normal changes the sign of H but leaves $|H|$ and K unchanged, making both curvature targets independent of normal orientation.

Paired near-surface queries. To supervise how distance varies around a patch, we sample queries on both sides of the surface along its analytic normals. For a sampled surface point $\mathbf{p}$ with unit normal $\mathbf{n}$, an offset of length ϵ gives $\mathbf{q}^{\pm} = \mathbf{p} \pm \epsilon\mathbf{n}$, and $U(\mathbf{q}^{\pm}) = \epsilon$. The distance labels are exact when $\mathbf{p}$ is the unique closest surface point to both queries. Sampling both directions provides supervision on both sides of the surface without assigning a consistent orientation to the input points.

Adapting the query range to curvature. Highly curved patches require smaller offsets to keep queries close to the local surface. We therefore sample offset lengths below the curvature-dependent limit $\epsilon_{\max} = \min\left(\epsilon_{\text{abs}}, \frac{\alpha}{\max(|\kappa_1|,|\kappa_2|)+\delta}\right)$. Here, $\epsilon_{\text{abs}} = 0.05$ is the maximum offset in the synthetic patch coordinates before normalization, α controls the curvature-dependent scale and is set to 1, and $\delta > 0$ stabilizes the denominator. This rule reduces the query range as curvature increases, while the absolute cap limits offsets on nearly flat patches. See Figure 2(a).

3.3 Patch-Based Geometric Field Prediction

PatchGF uses a shared Point Transformer encoder and three prediction branches for unsigned distances, normals, and curvatures (Figure 4 in Appendix A). These branches describe geometry at different locations: distances are evaluated at nearby spatial queries, normals are predicted at the input points, and curvatures are estimated at the patch center. Given a patch $\mathcal{P} = \{\mathbf{p}_j\}_{j=1}^{n_p}$, the encoder produces pointwise features $\Phi_{\text{patch}}(\mathcal{P})$ describes the geometry around point $\mathbf{p}_j$ using information from the patch. Symmetric pooling then yields a patch descriptor $\mathbf{z}_{\mathcal{P}} = \text{Pool}(\Phi_{\text{patch}}(\mathcal{P}))$, which summarizes the patch independently of the input point ordering. Together, the pointwise features and pooled descriptor provide local detail and broader geometric context for the prediction branches.

UDF prediction. The distance at a spatial query depends on its position relative to the observed surface samples. We capture this relationship through query-to-patch cross-attention. A query $\mathbf{q}$ is embedded as $\mathbf{z}_q = \Phi_{query}(\mathbf{q})$, and its geometric relationship to each patch point is encoded by $\mathbf{e}_{qj} = \phi(\mathbf{q} - \mathbf{p}_j, \|\mathbf{q} - \mathbf{p}_j\|_2)$. For one attention head with key dimension d, the attention scores and aggregated feature are

$$\mathbf{k}_{qj} = \mathbf{W}_K\mathbf{z}_{\mathcal{P}j} + \mathbf{W}_E\mathbf{e}_{qj}, \;\; s_{qj} = \frac{(\mathbf{W}_Q\mathbf{z}_q)^\top\mathbf{k}_{qj}}{\sqrt{d}}, \;\; \widetilde{\mathbf{z}}_q = \sum_{j=1}^{|\mathcal{P}|} \frac{\exp(s_{qj})}{\sum_{l=1}^{|\mathcal{P}|}\exp(s_{ql})} \left(\mathbf{W}_V\mathbf{z}_{\mathcal{P}j} + \mathbf{W}_E\mathbf{e}_{qj}\right),$$

where $\mathbf{z}_{\mathcal{P}j}$ is the encoded feature of point j from the patch, $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learned query, key, and value projections, respectively, while $\mathbf{W}_E$ projects relative geometric embeddings for addition to the keys and values. A lightweight decoder then predicts the local unsigned distance $\widetilde{U}_\theta(\mathbf{q} \mid \mathcal{P}) = \Phi_{\text{udf}}(\widetilde{\mathbf{z}}_q)$, allowing the prediction to depend on the query's relative position within the patch.

Normal prediction. For the input patch $\mathcal{P}$, a shared normal head Φ_{norm} uses each point feature and the pooled context to predict $\widetilde{\mathbf{n}}_{\mathcal{P},i} = \Phi_{\text{norm}}(\mathbf{z}_{\mathcal{P},i}, \mathbf{z}_{\mathcal{P}})$. Then it is normalized through $\widetilde{\mathbf{n}}_{\mathcal{P},i}/\max\left(\|\widetilde{\mathbf{n}}_{\mathcal{P},i}\|_2, \varepsilon_{\text{n}}\right)$, where $\varepsilon_{\text{n}} > 0$ stabilizes normalization. This formulation estimates normals directly from learned geometric features. Accordingly, the normal head is trained with sign-invariant supervision. Predictions and supervision are expressed in the same patch coordinate frame; predictions are transformed back to the common coordinate frame for inference.

Curvature prediction. The curvature head Φ_{curv} uses the pooled descriptor $\mathbf{z}_{\mathcal{P}}$ to estimate curvature at the patch center as $(\widetilde{H}_{\mathcal{P}}, \widetilde{K}_{\mathcal{P}}) = \Phi_{\text{curv}}(\mathbf{z}_{\mathcal{P}})$. The supervised quantities are $|H_{\mathcal{P}}|$ and $K_{\mathcal{P}}$ which are independent of the choice of normal orientation. Moreover, $H^2 - K = (\kappa_1 - \kappa_2)^2/4 \geq 0$, which supplies a geometric consistency condition for the predicted curvatures. The training objectives, loss functions used and model implementation details are discussed in Appendix A.

3.4 Anchor-based Local Field Fusion

At inference, we apply trained PatchGF to an input unoriented point cloud to construct a UDF for surface reconstruction. We select anchors from the input points and construct a fixed local patch around each anchor. For each query, we combine the distance predictions from nearby patches using distance-weighted averaging. Specifically, we collect M anchors $\{\mathbf{x}_m\}_{m=1}^{M} \subset \mathcal{X}$ by farthest-point sampling. For each anchor, we construct its patch $\mathcal{P}_m$ by selecting its n_p nearest neighbors. A patch

supports a query $\mathbf{q}$ when the distance to its nearest sample does not exceed a prescribed threshold r. Thus, we obtain the local field according to Sec. 3.3 as $\widehat{U}_m(\mathbf{q}) = \widehat{U}_\theta(\mathbf{q} \mid \mathcal{P}_m)$. For a query covered by multiple patches, we combine their predictions using normalized distance weights:

$$\widehat{U}(\mathbf{q}) = \frac{\sum_m w_m(\mathbf{q})\widehat{U}(\mathbf{q} \mid \mathcal{P}_m)}{\sum_m w_m(\mathbf{q})}, \qquad w_m(\mathbf{q}) = \exp\left(-\frac{\delta_m(\mathbf{q})^2}{2\sigma_f^2}\right),$$

where $\delta_m(\mathbf{q}) = \min_{\mathbf{p}\in\mathcal{P}_m} \|\mathbf{q}-\mathbf{p}\|_2$ is measured to the nearest sample in that patch, and σ_f controls the weight decay. Weighted fusion gives greater influence to nearby patches and attenuates the impact of discrepancies between overlapping predictions.

3.5 Surface Reconstruction by Projection and Geometry Field Guidance

Extracting a surface from a learned UDF is challenging because the field does not change sign across the surface. Given the UDF predicted by PatchGF, we first extract a mesh at a small positive level and project the resulting offset sheets toward the zero-level surface. To recover finer geometric details, we use the predicted normals and curvatures to construct an adaptive sizing field and guide local surface fitting and remeshing. This refinement adapts mesh resolution to local geometry while bringing the mesh into closer agreement with the input points.

Direct projection. Given the predicted UDF $\widehat{U}$, we extract a positive level $\ell > 0$ of the field. Each extracted vertex $\mathbf{y}$ is then updated once along the negative UDF gradient:

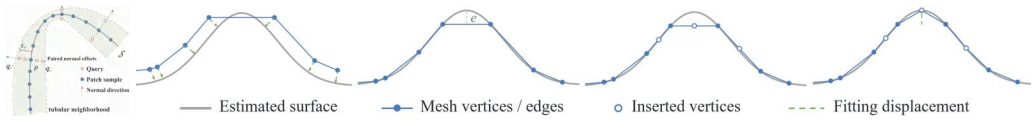
$$\mathbf{y} \leftarrow \mathbf{y} - \widehat{U}(\mathbf{y})\frac{\nabla\widehat{U}(\mathbf{y})}{\|\nabla\widehat{U}(\mathbf{y})\|_2 + \epsilon_g},$$

where $\epsilon_g > 0$ stabilizes gradient normalization. Even with an accurate UDF, projecting both offset sheets onto the surface leaves mesh connectivity unchanged, so coarse elements may still miss fine geometric detail. We therefore use PatchGF's predicted normals and curvatures to construct a sizing field that guides adaptive remeshing, allocating finer resolution where needed to improve reconstruction accuracy.

Geometry field guidance. We construct the sizing field from estimates of the maximum absolute principal curvature obtained from network predictions and local quadratic fits. Both estimates are aggregated over neighbors selected by spatial proximity and unoriented normal compatibility. We combine the aggregated estimates according to fit reliability to obtain the effective curvature κ_{eff}. Reliable fits attenuate larger network estimates, while the aggregated fitted curvature provides a lower bound. Using a local chord-error approximation, we define the curvature-based target edge length as $h_{\text{curv}} = \sqrt{\frac{8\varepsilon_s}{\kappa_{\text{eff}}}}$, where ε_s controls the approximation tolerance. Higher curvature therefore leads to shorter target edges. We additionally impose a lower bound based on local sample spacing and clip the target lengths to a prescribed range. Zero curvature gives the maximum target length, whereas samples without valid support receive the minimum. We then limit target-length variation along edges of a graph connecting compatible samples. Interpolating the resulting lengths in log space with regularized inverse-square distance weights yields the sizing field $\mathcal{G}_0$. The variation bound holds on graph edges but does not imply a global bound on the interpolated field. The sizing field guides octree refinement together with the offset-band and UDF interpolation criteria. PatchGF evaluates the UDF at newly introduced query locations, and conforming tetrahedra across balanced octree cells support positive-level-set extraction.

Local fitting and remeshing. The local quadratic fits used to construct the sizing field also serve as reference surfaces for refining the mesh. Each fit is estimated robustly from input points in an orthonormal frame defined by the predicted normal. Unoriented normal compatibility selects the fitting samples, while curvature controls the support radius, admissible height variation, and quadratic regularization. Curvature guides these choices without serving as a direct regression target. At each supported mesh vertex, we blend corrections obtained by linearizing the local surface equations:

$$\mathbf{d}(\mathbf{x}) = -\sum_i \omega_i(\mathbf{x})\frac{\Phi_i(\mathbf{x})\nabla_{\mathbf{x}}\Phi_i(\mathbf{x})}{\|\nabla_{\mathbf{x}}\Phi_i(\mathbf{x})\|_2^2}, \qquad \sum_i \omega_i(\mathbf{x}) = 1,$$



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

Figure 2: Conceptual illustrations of curvature-adaptive query sampling (a) and local mesh refinement (b)-(e), shown in 2D, where surfaces and meshes are represented by curves and polylines, respectively. (a)Paired training queries are sampled on both sides of the surface along analytic normals, with smaller offset ranges for patches of greater curvature. (b) Initial vertices lie away from the estimated curve. (c) Fitting brings vertices closer to the curve, but chord errors remain. (d) Edge splitting inserts new vertices along the existing chords. (e) Further fitting moves the inserted vertices toward the local fits, recovering geometric detail supported by the input.

Table 1: Quantitative comparison of surface reconstruction methods on clean inputs, inputs with 0.12% noise, and inputs with 10% outliers. We report L_1-Chamfer distance (CD; values multiplied by $\times 10^2$), F1-scores at thresholds of 0.005 and 0.01, and normal consistency (NC). Lower CD and higher F1 and NC indicate better performance. The **best** and second-best results in each dataset-condition-metric combination are shown in bold and underlined, respectively.

Dataset	Method	Clean				Noise (0.12%)				Outliers (10%)			
		CD ↓	$F1^{0.005}$ ↑	$F1^{0.01}$ ↑	NC ↑	CD ↓	$F1^{0.005}$ ↑	$F1^{0.01}$ ↑	NC ↑	CD ↓	$F1^{0.005}$ ↑	$F1^{0.01}$ ↑	NC ↑
ShapeNetCars 48K input	CAP-UDF	2.1767	0.7015	0.9877	0.9664	4.3775	0.3157	0.5053	0.8023	4.4331	0.3103	0.6972	0.7022
	LevelSetUDF	2.1602	0.6731	0.9872	0.9613	3.8509	0.3963	0.5929	0.8039	3.8825	0.3112	0.6843	0.7155
	GeoUDF	3.1512	0.6028	0.8856	0.9620	4.7794	0.3195	0.5095	0.8168	4.2874	0.3207	0.6382	0.7625
	DUDF	2.9879	0.6305	0.9832	0.9636	4.3809	0.3629	0.5132	0.8022	4.3439	0.3055	0.6463	0.6641
	DEUDF	2.0635	0.7057	**0.9952**	0.9694	3.8619	0.4195	0.6092	0.8561	3.7883	0.4146	**0.6846**	0.9051
	LoSF-UDF	2.9988	0.6288	0.8327	0.9618	3.1352	0.4101	0.6047	0.8581	3.8376	0.4523	0.6235	0.9158
	Ours	**2.0261**	**0.7297**	0.9897	**0.9709**	**3.0754**	**0.4783**	**0.7061**	**0.8697**	**3.6932**	**0.5089**	0.6742	**0.9311**
DeepFashion3D 48K input	CAP-UDF	1.1589	0.6570	0.9335	0.9525	3.6500	0.3015	0.6357	0.8141	3.2629	0.3102	0.6047	0.7907
	LevelSetUDF	1.1877	0.6793	0.9555	0.9604	3.8906	0.3797	0.6574	0.8407	2.6654	0.3022	0.6469	0.8512
	GeoUDF	1.2367	0.6811	0.9602	0.9386	3.9661	0.3455	0.5602	0.8377	3.4941	0.3064	0.6067	0.6148
	DUDF	1.0690	0.6455	**0.9832**	0.9521	3.4087	0.3182	0.5833	0.8321	4.6537	0.3041	0.6168	0.6098
	DEUDF	1.0509	0.6797	0.9783	0.9568	2.8913	0.3079	0.7058	0.8565	2.6573	0.4215	0.6992	0.8750
	LoSF-UDF	1.0981	0.6632	0.9298	0.9481	2.8626	0.4093	0.7029	0.8478	1.9354	0.4683	0.6474	0.8778
	Ours	**0.7982**	**0.7057**	0.9668	**0.9667**	**2.5442**	**0.4257**	**0.7368**	**0.8635**	**1.6696**	**0.5054**	**0.7078**	**0.9620**
FamousShapes 100K input	CAP-UDF	2.6355	0.7415	0.9149	0.9511	5.8609	0.4577	0.8123	0.9266	3.6664	0.4729	0.8434	0.9510
	LevelSetUDF	3.0256	0.7677	0.9527	0.9568	5.3744	0.4548	0.8521	0.9008	3.2531	0.4433	0.8520	0.9366
	GeoUDF	2.3784	0.7426	0.9461	0.9310	5.6818	0.4650	0.8442	0.9119	3.4089	0.4457	0.8455	0.9309
	DUDF	4.4148	0.7487	0.9188	0.9070	5.9884	0.4438	0.8348	0.8492	4.4430	0.4347	0.7174	0.8703
	DEUDF	2.5049	0.7776	0.9060	0.9268	5.8744	0.4772	0.8576	0.8849	6.5077	0.4032	0.7091	0.8614
	LoSF-UDF	3.4155	0.7432	0.9262	0.9585	5.7972	0.4316	0.8197	0.9230	4.4450	0.3127	0.7208	0.9585
	Ours	**2.0402**	**0.7861**	**0.9654**	**0.9692**	**4.5415**	**0.4868**	**0.8662**	**0.9326**	**3.0642**	**0.4825**	**0.8646**	**0.9691**

where $\Phi_i(\mathbf{x})$ is the signed height residual relative to local fit i, expressed in length units. The compactly supported weights account for distance, fit reliability, and unoriented normal agreement with the mesh. Displacement smoothing and geometric backtracking limit incoherent vertex motion and deterioration of mesh quality. Following adaptive isotropic remeshing principles (Dunyach et al., 2013), we alternate these vertex updates with edge splitting, edge collapse, edge flips, and tangential redistribution. Each edge takes the geometric mean of the sizing-field values $\mathcal{G}_0$ at its endpoints as its target length. Subsequent fitting moves newly inserted vertices toward the reference surfaces, allowing the refined mesh to recover curved detail supported by the input (Fig. 2). The local fits and sizing field remain fixed throughout refinement. Thus, the UDF provides initialization, while the predicted differential quantities guide mesh resolution and fitting to the input geometry. Comparisons are provided in Appendix C and Fig. 13.

4 Experiments

4.1 Comparisons

UDF learning. We compare against the unsupervised UDF methods CAP-UDF, LevelSet-UDF, DUDF, and DEUDF and the supervised methods GeoUDF and LoSF-UDF. Evaluation datasets covers ShapeNet-Cars (Chang et al., 2015), DeepFashion3D (Zhu et al., 2020), FamousShapes (Li et al., 2024), and real-world scans from SRB (Berger et al., 2013) (quantitative and visual results are in

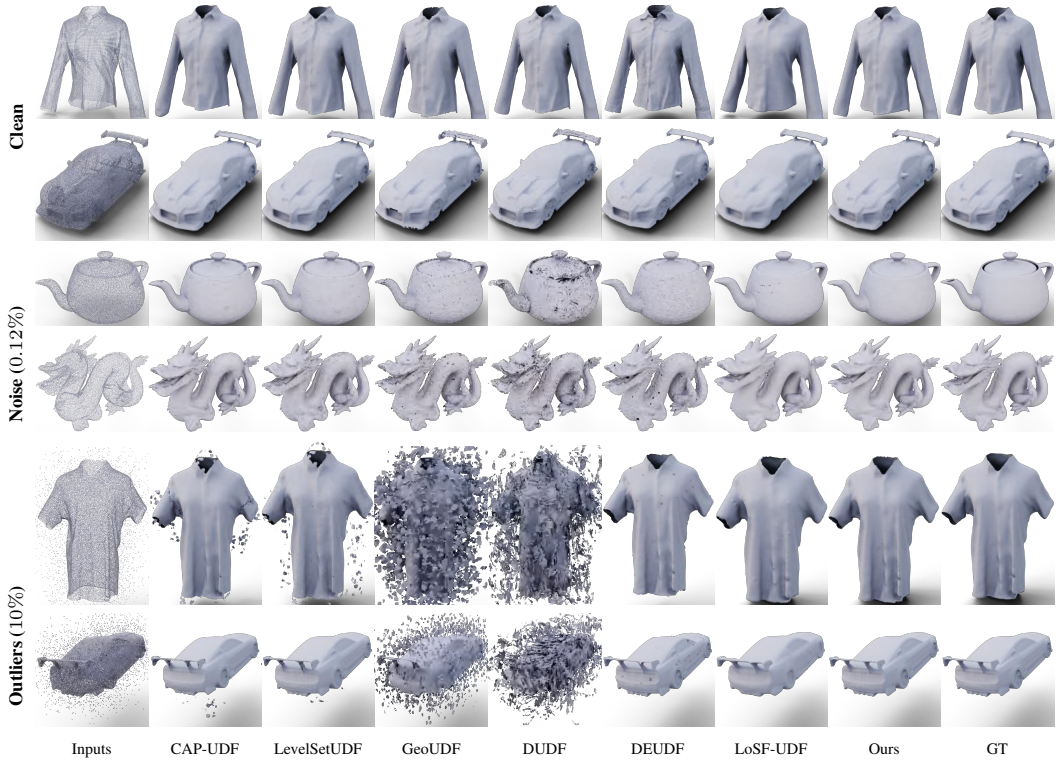


Figure 3: Qualitative comparison of UDF-based surface reconstruction methods on clean point clouds, point clouds with 0.12% noise, and point clouds with 10% outliers.

Table 2: Comparison of RMSE for unoriented normal, Mean (Abs) and Gaussian curvature on PCPNet dataset. "NA" denotes unsupported outputs. **Best** and Second Best are highlighted.

Methods	Unoriented Normal RMSE (degree) ↓				Mean Curvature (Abs) rectified RMSE ↓				Gaussian Curvature rectified RMSE ↓			
	Noise				Noise				Noise			
	Clean	0.12%	0.6%	1.2%	Clean	0.12%	0.6%	1.2%	Clean	0.12%	0.6%	1.2%
PCA	12.5221	12.9583	18.1304	26.6987	NA	NA	NA	NA	NA	NA	NA	NA
PCPNet	10.2220	11.7753	19.0375	23.5449	1.1896	1.2163	1.1256	1.2011	4.8985	4.5932	4.8581	5.4636
DeepFit	6.5825	9.2568	16.7343	23.2601	1.0134	1.2164	1.3255	1.4010	6.4846	5.6767	5.8067	5.8868
AdaFit	5.1638	9.0562	16.4479	21.9536	NA	NA	NA	NA	NA	NA	NA	NA
GraphFit	4.6047	8.8226	16.0969	21.6876	NA	NA	NA	NA	NA	NA	NA	NA
HSurfNet	4.3188	8.9508	16.2652	21.6221	NA	NA	NA	NA	NA	NA	NA	NA
NeAF	4.2661	9.2252	16.3617	21.7015	NA	NA	NA	NA	NA	NA	NA	NA
SHSNet	6.7103	8.7174	16.2030	21.5750	NA	NA	NA	NA	NA	NA	NA	NA
CMGNet	4.0762	8.4840	16.0814	21.8909	NA	NA	NA	NA	NA	NA	NA	NA
MSECNet	4.6496	8.9443	17.3698	22.3377	NA	NA	NA	NA	NA	NA	NA	NA
NGLO	4.0900	8.7278	16.5383	21.6420	NA	NA	NA	NA	NA	NA	NA	NA
NeuralGF	4.7258	9.7271	17.3710	23.4282	NA	NA	NA	NA	NA	NA	NA	NA
LGSF	7.7776	9.7670	18.3521	23.7596	NA	NA	NA	NA	NA	NA	NA	NA
PFF-Net	**3.3278**	**8.2251**	**15.6634**	**20.9886**	NA	NA	NA	NA	NA	NA	NA	NA
ASO	NA	NA	NA	NA	0.8947	0.9877	1.2662	1.9223	4.8915	5.9888	> 50	> 50
CurvCNC	NA	NA	NA	NA	0.8792	0.8945	0.9820	0.9833	3.2321	3.6131	3.9134	3.9647
OscuFit	4.0808	8.4629	16.3681	21.1667	1.0420	**0.5035**	0.6999	0.7324	2.5019	0.8982	**0.5862**	**0.5554**
Ours	4.0553	9.0162	17.1825	22.7097	**0.7099**	0.6215	**0.6091**	**0.6072**	**0.7871**	**0.7771**	0.6993	0.6568

Appendix C, Fig 8). We report Chamfer Distance (CD), normal consistency (NC), and F-scores. Tab. 1 presents quantitative results, and Fig. 3 provides visual comparisons.

Normal and curvature estimation We evaluate local differential properties on PCPNet, including unoriented normals, absolute mean curvature, and Gaussian curvature. Following OscuFit, normal accuracy is measured by orientation-invariant angular RMSE in degrees. We include curvature specified methods ASO (Lejemble et al., 2021) and CurvCNC (Lachaud et al., 2023). Following OscuFit, both curvatures are evaluated using rectified RMSE (Guerrero et al., 2018) defined

as $L(\tilde{\kappa}) = |\tilde{\kappa} - \kappa| / \max(|\kappa|, 1)$. Tab. 2 reports quantitative results. We provide visual results in Appendix C and Fig. 15.

4.2 Ablation Study

We conduct experiments on the main design choices of PatchGF, with complete quantitative and qualitative results provided in Appendix. Compared with LoSF-UDF, our model produces more accurate and spatially consistent UDF predictions, substantially reducing both on-surface and spatial errors. Replacing the fixed-radius neighborhood with a K-NN-derived adaptive radius improves robustness to variations in sampling density, with stable reconstruction observed for $K \geq 16$. We further find that the anchor ratio provides a flexible trade-off between efficiency and fidelity: 3% of the input points is sufficient for efficient evaluation, whereas increasing the ratio to 20% better preserves fine geometric details. More discussion can be viewed in Appendix B.

4.3 Results

Detailed mesh reconstruction. The anchor count controls UDF coverage and resolution: more anchors yield denser predictions and finer geometry at greater computational and memory cost. Learning UDF values and spatial gradients enables detailed mesh extraction comparable to per-shape optimization methods such as DEUDF. We select the count according to input density and desired detail. For ThreeDScans (Laric, 2012), we sample each surface to 300K points and use 65,536 anchors to cover small-scale structures. Fig. 11 in Appendix C provides the comparison. In addition, PatchGF can reconstruct inputs with complex topology from the NestPC (Li et al., 2025b) dataset, with qualitative results shown in Fig. 12 in Appendix C.

Robustness to noise and outliers. Anchor selection also improves outlier robustness. Neighborhood support excludes isolated points from the anchor set, while attention emphasizes informative inliers and down-weights near-surface anomalies within each patch. As shown in Fig. 9 in Appendix C, PatchGF remains robust to contamination ratios up to 30%. We compare PatchGF with LoSF-UDF on PCPNet under noise levels of up to 0.12%. Our results shown in Fig. 10 in Appendix C indicate that LoSF-UDF's apparent robustness is largely attributable to the strong Laplacian regularization employed by DCUDF (Hou et al., 2023), rather than to the inherent stability of its learned UDF. When the Laplacian weight is reduced, or when LoSF-UDF's predicted UDF is processed using PatchGF's direct projection method, its reconstruction quality degrades substantially and becomes unstable with certain noise levels. In contrast, PatchGF produces consistently stable reconstructions under the same conditions, demonstrating stronger intrinsic noise resistance and more reliable near-surface UDF learning at 0.12% noise level.

5 Conclusion

We presented PatchGF, a local learning framework that jointly predicts unsigned distances, unoriented normals, mean-curvature magnitude, and Gaussian curvature from unoriented point clouds. Trained entirely on synthetic quadratic patches, PatchGF uses paired queries with curvature-adaptive offsets to learn near-surface distance variation. At inference, fixed anchor patches support local field assembly, while the predicted differential quantities guide adaptive extraction, surface fitting, and remeshing. Experiments on synthetic and real-world datasets demonstrate competitive reconstruction quality, accurate normal and curvature estimation, and robustness to noise and outliers. The trained network supports parallel feed-forward inference without per-shape parameter optimization.

Future work will explore broader applications of the predicted differential quantities in geometry processing and physics-based simulation. We also plan to extend PatchGF to large-scale indoor scenes through hierarchical and adaptive anchor allocation, and to investigate temporally consistent reconstruction from sequential scans of dynamic geometry.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology and experiments, and assist with paper writing and proofreading papers. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, implement methods, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. We have reviewed all AI-assisted work, including evaluating methodological feedback, and checking AI-assisted texts for accuracy. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

To the best of our knowledge, the research presented in this paper complies with the applicable Code of Ethics. We have considered the ethical implications of our data, methodology, and potential applications, including privacy, security, fairness, potential harm, and research integrity, and have not identified any significant ethical concerns. Any relevant ethical approvals, data usage restrictions, funding sources, and potential conflicts of interest are disclosed in the paper where applicable.

REPRODUCIBILITY STATEMENT

We will make the project's source code and the datasets used in our experiments available to enable independent reproduction and verification of the empirical results.

REFERENCES

P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, pp. 39–48, 2007.

Nina Amenta and Marshall Bern. Surface reconstruction by voronoi filtering. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, pp. 39–48, 1998.

Yizhak Ben-Shabat and Stephen Gould. Deepfit: 3d surface fitting via neural network weighted least squares. In *European conference on computer vision*, pp. 20–34. Springer, 2020.

Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Matthew Berger, Joshua A Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T Silva. A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(2):1–17, 2013.

Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, 31(5):1765–1774, 2012.

Yohann Béarzi, Julie Digne, and Raphaëlle Chaine. Wavejets: A local frequency framework for shape details amplification. *Computer Graphics Forum*, 37(2):13–24, 2018.

Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer aided geometric design*, 22(2):121–146, 2005.

Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pp. 608–625. Springer, 2020.

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

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

Xin Deng, Bo Yang, Yifei Shi, and Bing Wang. Primitiveudf: Primitive-based unsigned distance fields for surface reconstruction from point clouds. In *European Conference on Computer Vision*, September 2026.

Marion Dunyach, David Vanderhaeghe, Loïc Barthe, and Mario Botsch. Adaptive remeshing for real-time mesh deformation. In *34th Annual Conference of the European Association for Computer Graphics (Eurographics 2013)*. The Eurographics Association, 2013.

Miguel Fainstein, Viviana Siless, and Emmanuel Iarussi. Dudf: Differentiable unsigned distance fields with hyperbolic scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4484–4493, June 2024.

Rao Fu, Qian Li, Liang Yu, and Jianmin Zheng. Oscufit: Learning to fit osculating implicit quadrics for point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4022–4030, 2026.

Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. Pcpnet learning local shape properties from raw point clouds. *Computer Graphics Forum*, 37(2):75–85, 2018.

Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 71–78, 1992.

Fei Hou, Chiyu Wang, Wencheng Wang, Hong Qin, Chen Qian, and Ying He. Iterative poisson surface reconstruction (ipsr) for unoriented points. *ACM Transactions on Graphics (TOG)*, 41(4), 2022.

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

Jian Hu, Yanggeng Li, Fei Hou, Junhui Hou, Zhen Zhang, Shiqing Wang, Na Lei, and Ying He. A lightweight UDF learning framework for 3D reconstruction based on local shape functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1297–1307, 2025.

Hanwen Jiang, Haitao Yang, Georgios Pavlakos, and Qixing Huang. Cofie: learning compact neural surface representations with coordinate fields. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.

Evangelos Kalogerakis, Patricio Simari, Derek Nowrouzezahrai, and Karan Singh. Robust statistical estimation of curvature on discretized surfaces. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, pp. 13–22, 2007.

Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):1–13, 2013.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pp. 61–70, 2006.

Jiayi Kong, Xuhui Chen, Chen Zong, Fei Hou, Junhui Hou, Wenping Wang, and Ying He. Metric—phase fields: Decoupling distance and sign for thin-structure reconstruction from unoriented point clouds. In *ICML 2026*, 2026.

J.-O. Lachaud, D. Coeurjolly, C. Labart, P. Romon, and B. Thibert. Lightweight curvature estimation on point clouds with randomized corrected curvature measures. *Computer Graphics Forum*, 42 (5):e14910, 2023.

Oliver Laric. Three D scans. https://threedscans.com/, 2012.

T. Lejemble, D. Coeurjolly, L. Barthe, and N. Mellado. Stable and efficient differential estimators on oriented point clouds. *Computer Graphics Forum*, 40(5):205–216, 2021.

Jan Eric Lenssen, Christian Osendorfer, and Jonathan Masci. Deep iterative surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11244–11253, June 2020.

Keqiang Li, Mingyang Zhao, Huaiyu Wu, Dong-Ming Yan, Zhen Shen, Fei-Yue Wang, and Gang Xiong. Graphfit: Learning multi-scale graph-convolutional representation for point cloud normal estimation. In *European Conference on Computer Vision*, pp. 651–667. Springer, 2022a.

Qing Li, Yu-Shen Liu, Jin-San Cheng, Cheng Wang, Yi Fang, and Zhizhong Han. HSurf-Net: Normal estimation for 3D point clouds by learning hyper surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 4218–4230, 2022b.

Qing Li, Huifang Feng, Kanle Shi, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Neural gradient learning and optimization for oriented point normal estimation. In *SIGGRAPH Asia 2023 Conference Papers*, 2023a.

Qing Li, Huifang Feng, Kanle Shi, Yue Gao, Yi Fang, Yu-Shen Liu, and Zhizhong Han. NeuralGF: Unsupervised point normal estimation by learning neural gradient function. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023b.

Qing Li, Huifang Feng, Kanle Shi, Yue Gao, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Learning signed hyper surfaces for oriented point cloud normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9957–9974, 2024.

Qing Li, Huifang Feng, Xun Gong, and Yu-Shen Liu. Learning normals of noisy points by local gradient-aware surface filtering. In *International Conference on Computer Vision (ICCV)*, 2025a.

Qing Li, Huifang Feng, Kanle Shi, Yue Gao, Yi Fang, Yu-Shen Liu, and Zhizhong Han. PFF-Net: Patch feature fitting for point cloud normal estimation. 2025b.

Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Neaf: Learning neural angle fields for point normal estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 1396–1404, 2023c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance function from point clouds by learning to pull space onto surface. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 7246–7257, 2021.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. In *special issue of International Journal of Computational Geometry and Applications*, volume 14, pp. 261–276, 2004.

Quentin Mérigot, Maks Ovsjanikov, and Leonidas J. Guibas. Voronoi-based curvature and feature estimation from point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 17 (6):743–756, 2011.

Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics (TOG)*, 22(3):463–470, July 2003.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Siyu Ren and Junhui Hou. Shape as line segments: Accurate and flexible implicit surface representation. In *International Conference on Learning Representations*, volume 2025, pp. 82842–82862, 2025.

Siyu Ren, Junhui Hou, Xiaodong Chen, Ying He, and Wenping Wang. Geoudf: Surface reconstruction from 3d point clouds via geometry-guided distance representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14214–14224, 2023.

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision (ECCV)*, 2020.

Zixiong Wang, Yunxiao Zhang, Rui Xu, Fan Zhang, Peng-Shuai Wang, Shuangmin Chen, Shiqing Xin, Wenping Wang, and Changhe Tu. Neural-singular-hessian: Implicit neural representation of unoriented point clouds by enforcing singular hessian. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.

Yingrui Wu, Mingyang Zhao, Keqiang Li, Weize Quan, Tianqi Yu, Jianfeng Yang, Xiaohong Jia, and Dong-Ming Yan. Cmg-net: Robust normal estimation for point clouds via chamfer normal distance and multi-scale geometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6171–6179, 2024.

Haoyi Xiu, Xin Liu, Weimin Wang, Kyoung-Sook Kim, and Masashi Matsuoka. Msecnet: Accurate and robust normal estimation for 3d point clouds by multi-scale edge conditioning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 2535–2543, 2023.

Cheng Xu, Fei Hou, Wencheng Wang, Hong Qin, Zhebin Zhang, and Ying He. Details enhancement in unsigned distance field learning for high-fidelity 3d surface reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8806–8814, 2025.

Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Gifs: Neural implicit function for general shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, and Zhizhong Han. Learning a more continuous zero level set in unsigned distance fields through level set projection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

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

Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *European Conference on Computer Vision*, pp. 512–530. Springer, 2020.

Runsong Zhu, Yuan Liu, Zhen Dong, Yuan Wang, Tengping Jiang, Wenping Wang, and Bisheng Yang. Adafit: Rethinking learning-based normal estimation on point clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6098–6107, 2021.

A TRAINING DETAILS

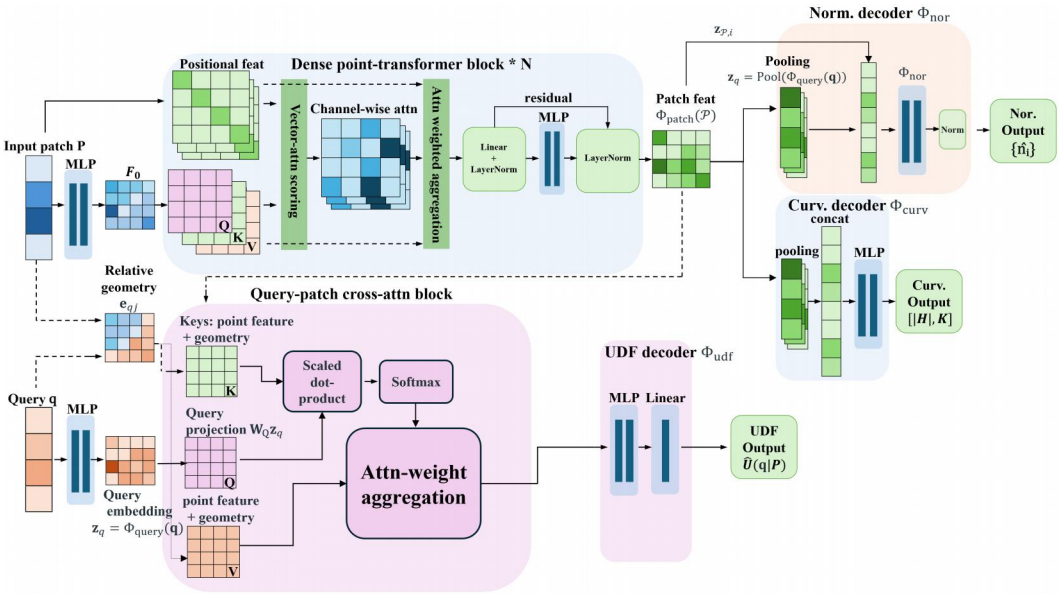


Figure 4: A point MLP and dense point transformers encode the local point patch. Query–point cross-attention combines query embeddings with geometric context for UDF prediction through a decoder, while two separate attention-pooled heads estimate unoriented normal, mean and Gaussian curvatures.

Training objectives. We jointly supervise distance, normal, and curvature predictions. Let $\mathcal{B}$ denote a batch of input patches, $\mathcal{Q}_\mathcal{P}$ the off-surface training queries associated with patch $\mathcal{P}$, and $N_\mathrm{q} = \sum_{\mathcal{P}\in\mathcal{B}} |\mathcal{Q}_\mathcal{P}|$. For brevity, $\sum_{\mathcal{P},\mathbf{q}}$ denotes summation over $\mathcal{P} \in \mathcal{B}$ and $\mathbf{q} \in \mathcal{Q}_\mathcal{P}$. Let ρ_β denote the Smooth-L_1 penalty with transition parameter β, and define the distance residual

$$r_{\mathcal{P},\mathbf{q}} = \widehat{U}_\theta(\mathbf{q} \mid \mathcal{P}) - U_\mathcal{P}(\mathbf{q}).$$

We combine global regression with a near-surface weighted term:

$$\mathcal{L}_\mathrm{global} = \frac{1}{N_\mathrm{q}} \sum_{\mathcal{P},\mathbf{q}} \rho_{\beta_\mathrm{d}}(r_{\mathcal{P},\mathbf{q}}), \quad \mathcal{L}_\mathrm{near} = \frac{\sum_{\mathcal{P},\mathbf{q}} w_{\mathcal{P},\mathbf{q}}\, \rho_{\beta_\mathrm{d}}(r_{\mathcal{P},\mathbf{q}})}{\sum_{\mathcal{P},\mathbf{q}} w_{\mathcal{P},\mathbf{q}} + \varepsilon},$$

$$w_{\mathcal{P},\mathbf{q}} = \exp\left(-\frac{U_\mathcal{P}(\mathbf{q})}{\sigma_\mathrm{d}}\right), \quad \mathcal{L}_\mathrm{udf} = (1-\alpha_\mathrm{d})\mathcal{L}_\mathrm{global} + \alpha_\mathrm{d}\mathcal{L}_\mathrm{near},$$

where $\alpha_\mathrm{d} \in [0,1]$ balances global and near-surface supervision, and $\sigma_\mathrm{d} > 0$ controls the spatial weighting.

For unit predicted and target normals, we use

$$\begin{aligned}\ell_\mathrm{n}(\widehat{\mathbf{n}}, \mathbf{n}) &= a_\mathrm{n} \min_{s\in\{-1,1\}} \|\widehat{\mathbf{n}} - s\mathbf{n}\|_2^2 + b_\mathrm{n}\|\widehat{\mathbf{n}} \times \mathbf{n}\|_2 \\ &= 2a_\mathrm{n}\left(1 - |\widehat{\mathbf{n}}^\top \mathbf{n}|\right) + b_\mathrm{n}\|\widehat{\mathbf{n}} \times \mathbf{n}\|_2.\end{aligned}$$

Both terms are invariant to reversing either normal and vanish when the two normal lines coincide. The point-normal objective is

$$\mathcal{L}_\mathrm{normal} = \frac{\displaystyle\sum_{\mathcal{P}\in\mathcal{B}} \sum_{i=1}^{N_\mathcal{P}} m_{\mathcal{P},i}\eta_{\mathcal{P},i}\, \ell_\mathrm{n}(\widehat{\mathbf{n}}_{\mathcal{P},i}, \mathbf{n}_{\mathcal{P},i})}{\displaystyle\sum_{\mathcal{P}\in\mathcal{B}} \sum_{i=1}^{N_\mathcal{P}} m_{\mathcal{P},i}\eta_{\mathcal{P},i} + \varepsilon},$$

where $m_{\mathcal{P},i} \in \{0,1\}$ indicates a valid normal label, $\eta_{\mathcal{P},i} > 0$ is a radial weight within the patch, and $a_\mathrm{n}, b_\mathrm{n} \geq 0$ balance the regression and sine penalties.

Direct normal supervision is complemented by the off-surface Eikonal loss, which regularizes the distance field:

$$\mathcal{L}_{\text{eik}} = \frac{1}{N_{\text{q}}} \sum_{\mathcal{P},\mathbf{q}} \left(\left\| \nabla_{\mathbf{q}} \widehat{U}_\theta(\mathbf{q} \mid \mathcal{P}) \right\|_2 - 1 \right)^2 .$$

This gradient is used to constrain the UDF rather than to define the directly predicted surface normals.

Mean-curvature magnitude and signed Gaussian curvature are supervised separately:

$$\mathcal{L}_{\text{curv}} = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{P}\in\mathcal{B}} \left[a_H \rho_{\beta_{\text{c}}}(\widehat{h}_{\mathcal{P}} - h_{\mathcal{P}}) + a_K \rho_{\beta_{\text{c}}}(\widehat{K}_{\mathcal{P}} - K_{\mathcal{P}}) + \lambda_{\text{valid}} \tau_{\text{c}} \, \text{softplus}\left(\frac{\widehat{K}_{\mathcal{P}} - \widehat{h}^2_{\mathcal{P}}}{\tau_{\text{c}}} \right) \right],$$

where $a_H, a_K \geq 0$ balance the curvature regression terms, $\tau_{\text{c}} > 0$ controls the smoothness of the consistency penalty. Applying separate regression penalties prevents cancellation between the curvature residuals. The final term softly penalizes violations of $\widehat{K}_{G,\mathcal{P}} \leq \widehat{h}^2_{\mathcal{P}}$.

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{udf}} + \lambda_{\text{normal}} \mathcal{L}_{\text{normal}} + \lambda_{\text{eik}} \mathcal{L}_{\text{eik}} + \lambda_{\text{curv}} \mathcal{L}_{\text{curv}} + \mathcal{L}_{\text{aux}}.$$

Here $\mathcal{L}_{\text{aux}}$ collects the weighted off-surface anti-collapse penalty, patch-center normal and coplanarity supervision, normal-variation supervision, surface-directed UDF-gradient supervision, and consistency losses between paired transformed patches. Their definitions and training schedules are specified separately. All loss coefficients are nonnegative.

$$\begin{aligned} \mathcal{L}_{\text{aux}} = {} & \lambda_{\text{off}} \mathcal{L}_{\text{off}} + \lambda_{\text{center}} \mathcal{L}_{\text{center}} + \lambda_{\text{cop}} \mathcal{L}_{\text{cop}} \\ & + \lambda_{\text{var}} \mathcal{L}_{\text{var}} + \lambda_{\text{dir}} \mathcal{L}_{\text{dir}} + \lambda_{\text{pair}} \mathcal{L}_{\text{pair}}. \end{aligned}$$

Here, $\mathcal{L}_{\text{off}}$ discourages off-surface UDF collapse, $\mathcal{L}_{\text{center}}$ supervises patch-center normals, and $\mathcal{L}_{\text{cop}}$ supervises coplanarity. The remaining terms supervise normal variation ($\mathcal{L}_{\text{var}}$), surface-directed UDF gradients ($\mathcal{L}_{\text{dir}}$), and consistency between paired transformed patches ($\mathcal{L}_{\text{pair}}$). All coefficients are nonnegative; setting a coefficient to zero disables the corresponding term.

Implementation and training process We train PatchGF on synthetic local patches sampled from smooth quadratic surfaces, where each patch contains $K = 128$ surface points and 20 surrounding queries supervised with UDF and curvature values. The patches are mean-centered, scale-normalized, randomly rotated, and augmented with noise and outliers to improve geometric invariance and robustness. A dense point-transformer encoder captures the local surface geometry. Query-to-patch cross-attention predicts the UDF at spatial queries, while a separate curvature head uses the pooled patch descriptor to predict mean-curvature magnitude $|H|$ and signed Gaussian curvature K at the patch center. A dedicated normal head combines pointwise features with pooled patch context to predict unoriented normals at the input points. We optimize the network for 300 epochs using AdamW (Loshchilov & Hutter, 2017), with a batch size of 32 and a learning rate cosine-annealed from 10^{-3} to 10^{-5} on a single NVIDIA RTX 4090 GPU. At inference, we normalize the input point cloud, sample anchor points, and discard neighborhoods with insufficient support. Valid predictions from overlapping patches are aggregated into a global UDF field, from which the final surface is extracted using the one-step projection method, and for detailed mesh reconstruction, we use the geometry field guidance meshing.

B Ablation Study

Model design comparison with LoSF-UDF. We isolate the new model design on a randomly selected testing patch, uniformly sampling queries along the z-axis and evaluating both LoSF-UDF and our method. As shown in Fig. 5, our prediction more closely matches the ground-truth UDF, $U(z) = |z|$, with lower errors than LoSF-UDF. These errors remain small and uniform across the query interval, indicating improved local accuracy and spatial consistency.

Patch radius and K neighbors. LoSF-UDF constructs patches using a fixed radius R, whose optimal value varies with sampling density and query location. Our anchor-based strategy instead

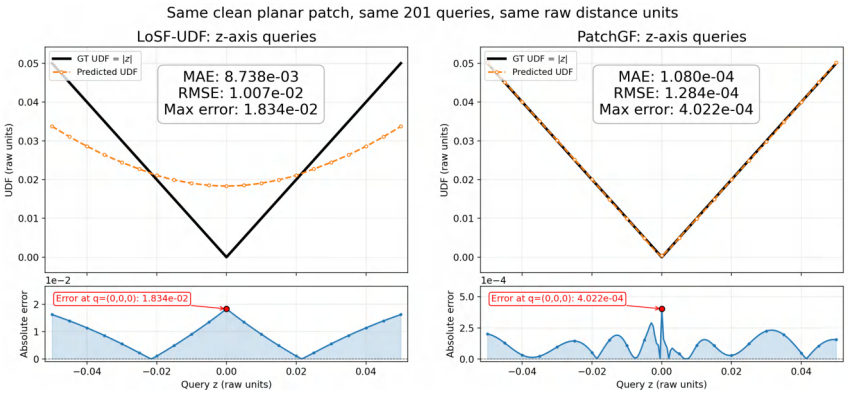
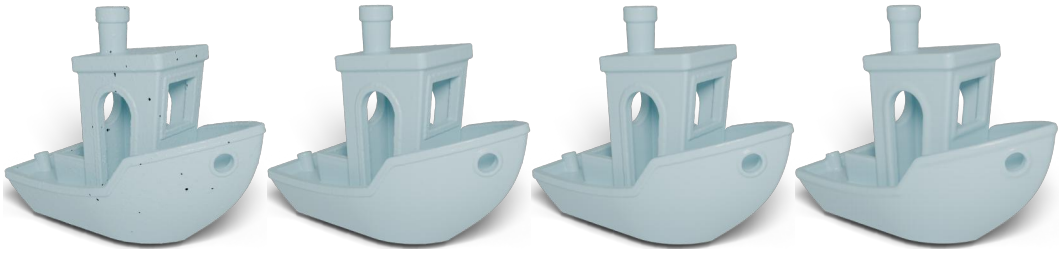


Figure 5: Evaluation on a testing patch. At the on-surface query $\mathbf{q} = (0, 0, 0)$, LoSF-UDF produces the UDF error approximately 45 times larger than our method, and approximately 80 times larger for MAE on whole space.

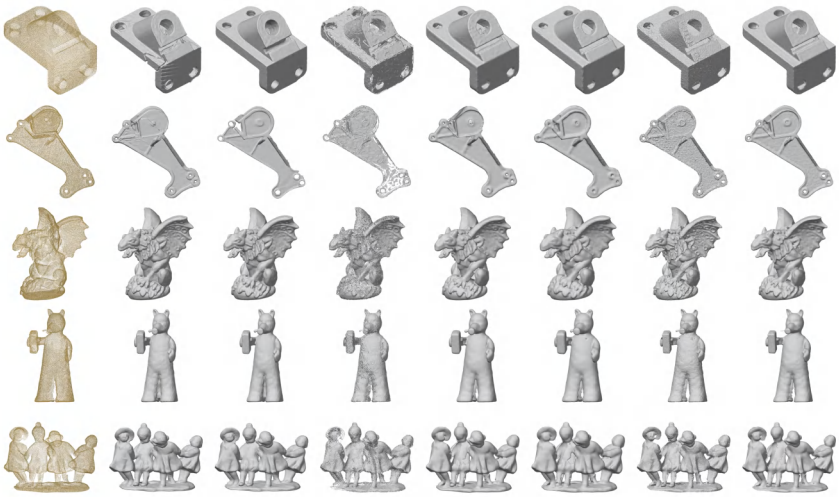
associates each anchor with its K nearest input points and sets the radius to the K-th-neighbor distance. For approximately uniform samples from a two-dimensional surface with density ρ, $K \approx \rho\pi R^2$. Thus, $R \approx \sqrt{\frac{K}{\pi\rho}}$, implying $R \propto \sqrt{K}$ at fixed density. Specifying K is therefore more intuitive and density-adaptive than manually choosing R. We cap both K and the resulting radius to preserve locality. Fig. 6 shows that performance is stable when each patch contains at least 16 points.



$K = 8$ $K = 16$ $K = 32$ $K = 128$

Figure 6: Qualitative reconstruction for different choices of K, demonstrating robustness when $K \geq 16$.

Choice of anchor points. The anchor ratio trades computational efficiency for reconstruction fidelity. Empirically, 3% of the input points suffices for resource-constrained evaluation, while 20% better preserves fine details. Unlike iterative per-shape methods such as DEUDF, our feed-forward model evaluates all anchors in parallel.



Input	CAP-UDF	LevelSetUDF	DUDF	GeoUDF	LoSF-UDF	DEUDF	PatchGF
Method	CAP-UDF	LevelSetUDF	DUDF	GeoUDF	LoSF-UDF	DEUDF	PatchGF
CD (×100) ↓	0.8825	0.7398	0.4965	0.5889	0.7302	0.4476	**0.3960**
F1 @ 0.005 ↑	0.3848	0.4378	0.6902	0.5461	0.4008	0.7987	**0.7990**
F1 @ 0.01 ↑	0.6581	0.7276	0.9108	0.8825	0.8137	0.9587	**0.9670**
NC ↑	0.9079	0.9222	0.8862	0.9389	0.9363	0.9461	**0.9515**

Figure 8: SRB dataset reconstruction comparison. The table reports averages over the five shapes shown above.



Figure 7: Qualitative reconstruction for different A.

C VISUALIZATION RESULTS

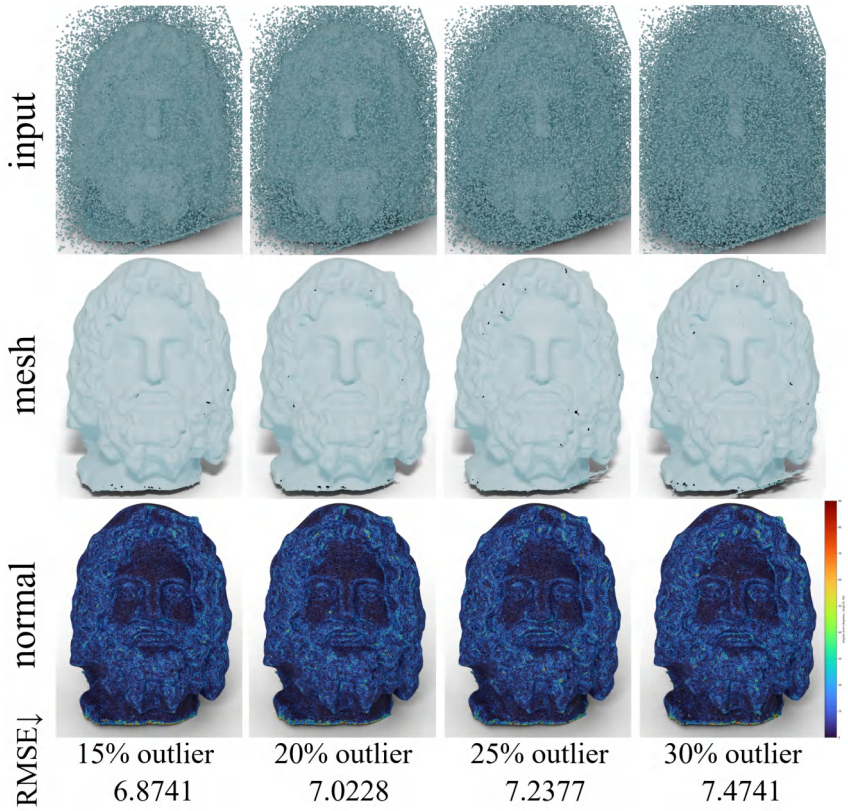


Figure 9: PatchGF reconstructs surfaces and estimates normals with outlier ratios of up to 30%.

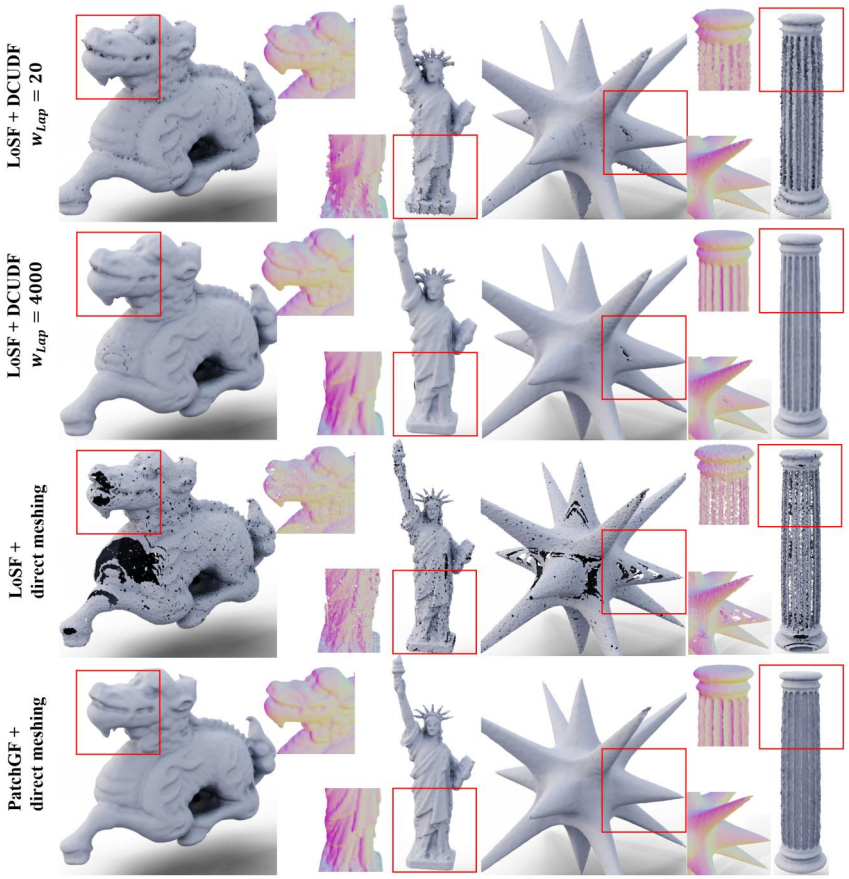


Figure 10: Qualitative comparison between LoSF-UDF and PatchGF with different Laplacian weight of DCUDF under PCPNet with 100K inputs and 0.12% Noise.

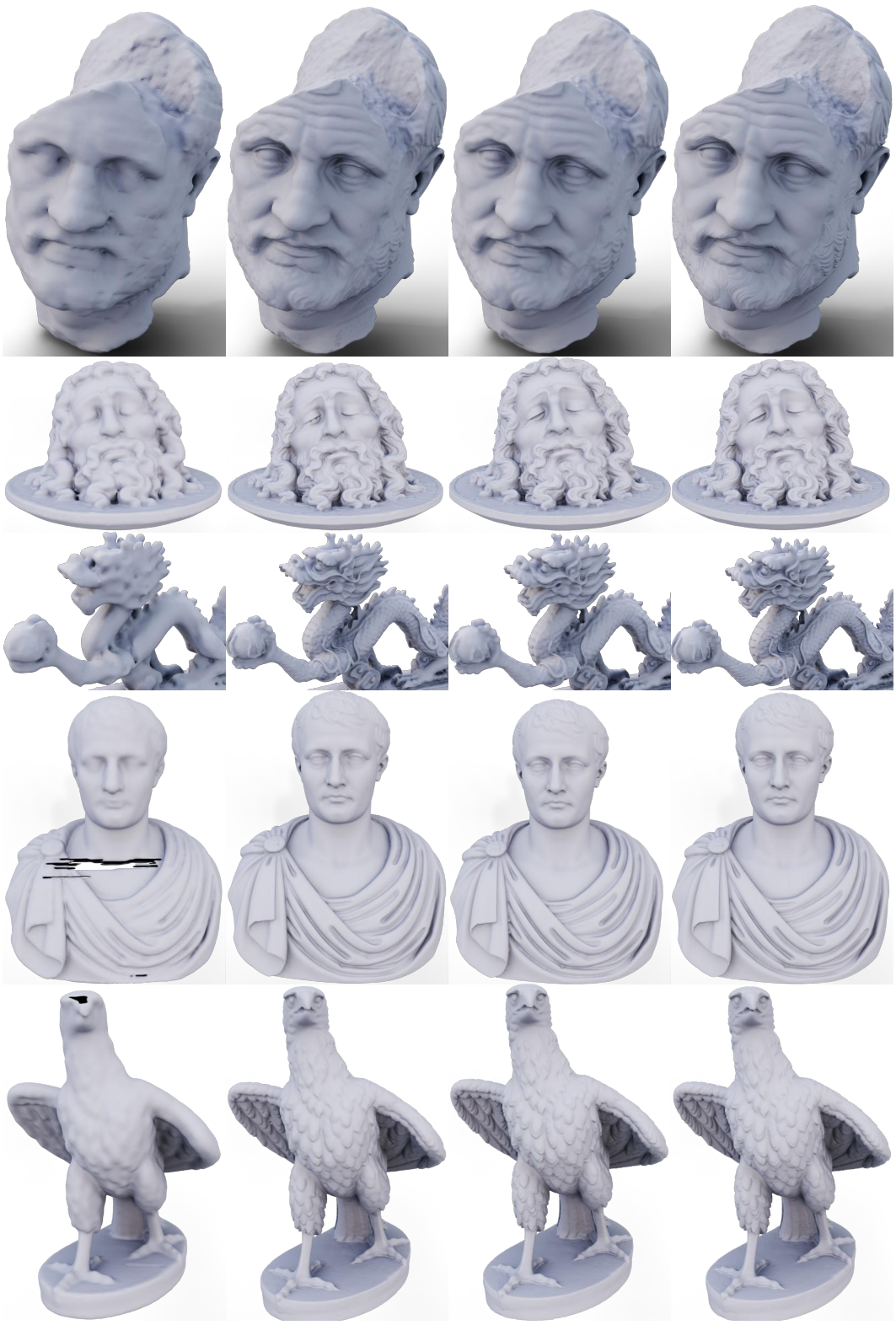


Figure 11: Results on the ThreeDScans dataset with 300K input points.

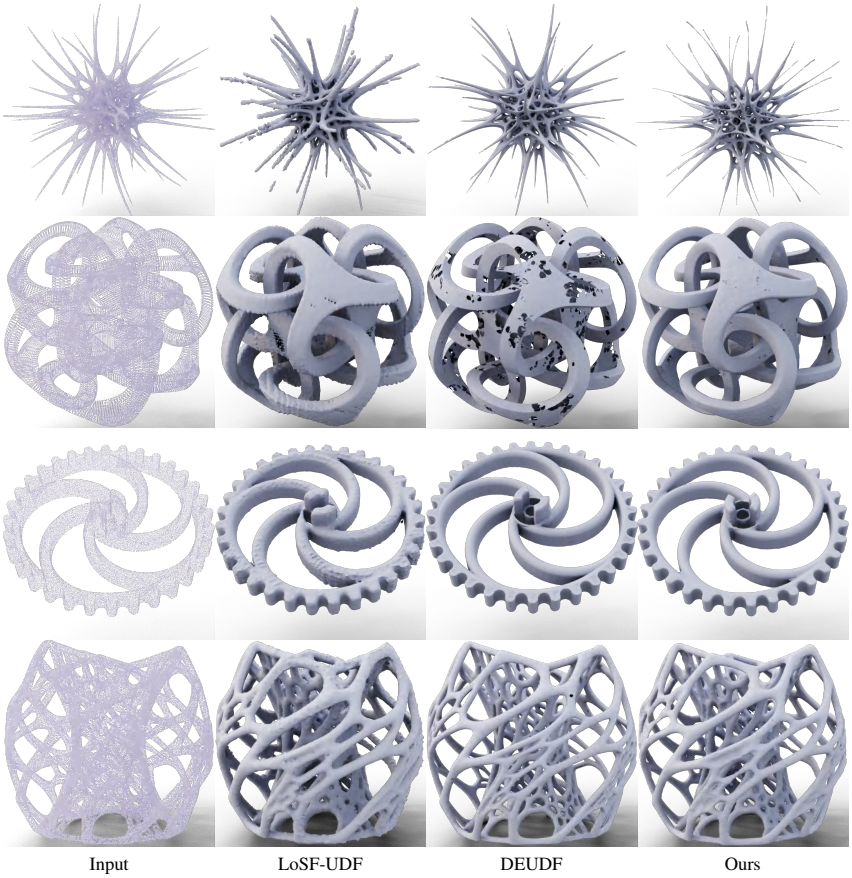


Figure 12: Results on the NestPC dataset.

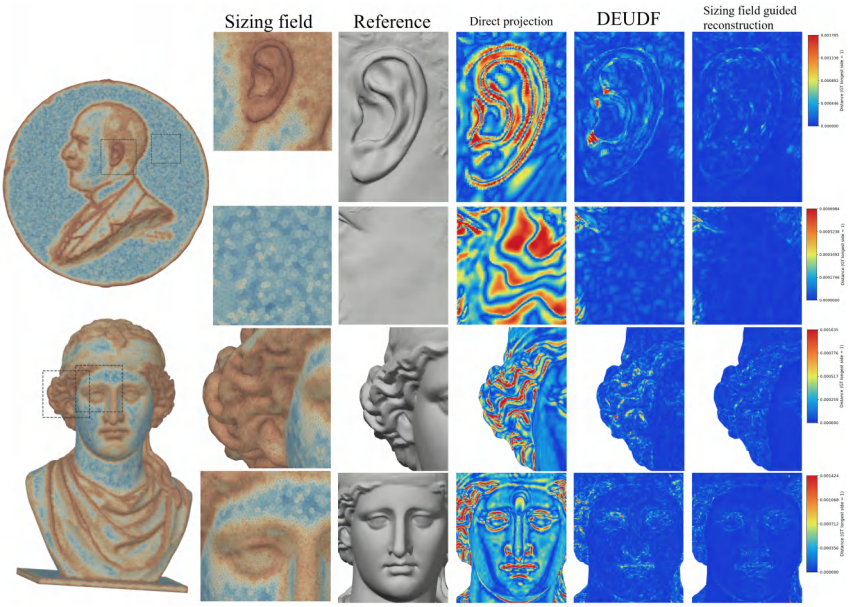


Figure 13: With guidance from the sizing field, PatchGF is able to reconstruct greater detail than DEUDF.

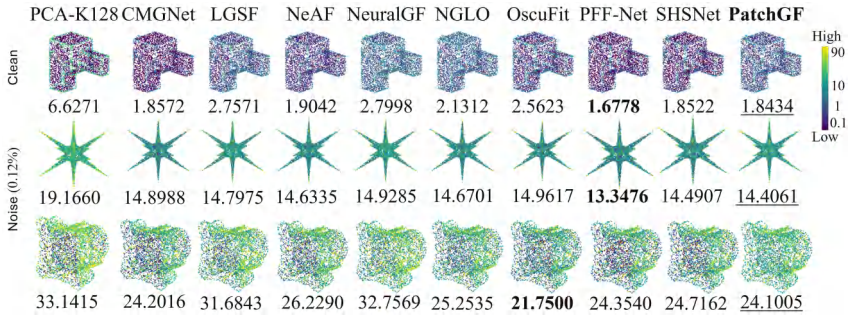


Figure 14: Visual comparison results of normal prediction. PatchGF achieves comparable results. Input shapes are color-coded by the error in logarithmic scale, with blue/yellow indicating low/high error.

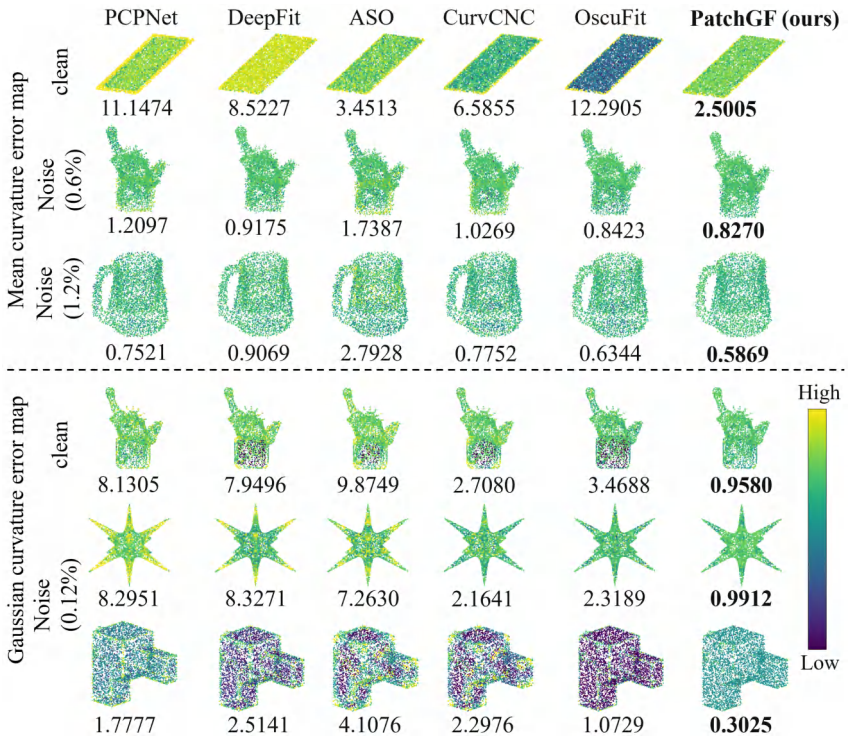


Figure 15: Visual comparison results of mean and Gaussian curvature estimation using rectified errors on the PCPNet dataset. Input shapes are color-coded by the error in logarithmic scale, with blue/yellow indicating low/high error.