

# FlatCAD: Fast Curvature Regularization of Neural SDFs for CAD Models

HAOTIAN YIN\*, New Jersey Institute of Technology, United States

ALEKSANDER PŁOCHARSKI\*, Warsaw University of Technology, Poland and IDEAS NCBR, Poland

MICHAŁ JAN WŁODARCZYK, Warsaw University of Technology, Poland

MIKOŁAJ KIDA, Warsaw University of Technology, Poland

PRZEMYSŁAW MUSIALSKI, New Jersey Institute of Technology, United States and IDEAS Research Institute, Poland

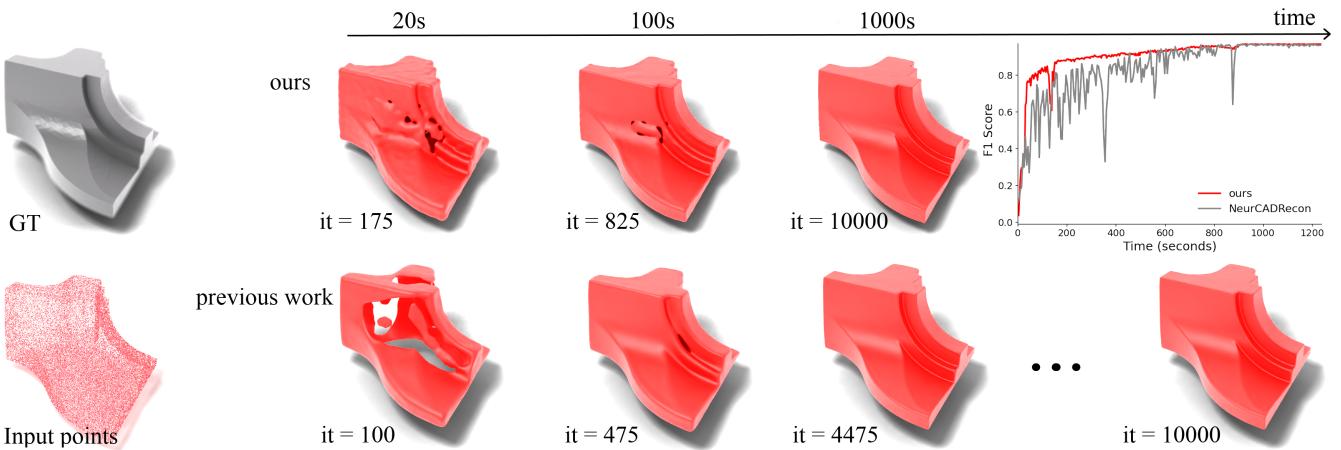


Fig. 1. Comparison of our Weingarten regularizer (top row) and the full-Hessian-based method of previous work [Dong et al. 2024] (bottom row) on a CAD part reconstruction: both achieve similar-quality surfaces, but our method converges in roughly half the time and uses about half the memory.

Neural signed-distance fields (SDFs) are a versatile backbone for neural geometry representation, but enforcing CAD-style developability usually requires Gaussian-curvature penalties with full Hessian evaluation and second-order differentiation, which are costly in memory and time. We introduce an off-diagonal Weingarten loss that regularizes only the mixed shape operator term that represents the gap between principal curvatures and flattens the surface. We present two variants: a finite-difference version using six SDF evaluations plus one gradient, and an auto-diff version using a single Hessian-vector product. Both converge to the exact mixed term and preserve the intended geometric properties without assembling the full Hessian. On the ABC benchmarks the losses match or exceed Hessian-based baselines while cutting GPU memory and training time by roughly a factor of two. The method is drop-in and framework-agnostic, enabling scalable curvature-aware SDF learning for engineering-grade shape reconstruction. Our code is available at <https://flatcad.github.io/>.

**CCS Concepts:** • Computing methodologies → Computer graphics; Shape modeling; Regularization; Machine learning algorithms.

**Additional Key Words and Phrases:** neural implicit representations, neural surface, surface reconstruction, curvature regularization, computer-aided design

\*Authors contributed equally to this work.

Authors' addresses: Haotian Yin, New Jersey Institute of Technology, United States, hy9@njit.edu; Aleksander Płocharski, Warsaw University of Technology, Poland and IDEAS NCBR, Poland, aleksander.plocharSKI@pw.edu.pl; Michał Jan Włodarczyk, Warsaw University of Technology, Poland; Mikolaj Kida, Warsaw University of Technology, Poland; Przemysław Musalski, New Jersey Institute of Technology, United States and IDEAS Research Institute, Poland, przem@njit.edu.

## 1 INTRODUCTION

Neural implicit representations of geometry, specifically *signed-distance fields* (SDFs) encoded by multilayer perceptrons have become a novel and powerful representation in 3D vision and graphics. Early work [Mescheder et al. 2018; Park et al. 2019] showed that a modest MLP could store a watertight shape and support arbitrary resolution through continuous evaluation and differentiation. That promise was broadened by SIREN [Sitzmann et al. 2020], whose sinusoidal activations capture fine geometric detail and high-frequency signals, and positional encodings that brought ReLU-based networks to high detail accuracy [Tancik et al. 2020] and essentially real-time performance [Müller et al. 2022].

However, for accurate geometric representations, a remaining challenge was geometric faithfulness: vanilla SDF fits can satisfy point-wise losses while remaining wildly inconsistent in surface detail or curvature. Lipman et al. addressed this with the Implicit Geometric Regularizer (IGR) [Gropp et al. 2020], adding an Eikonal term  $\|\nabla f\| \approx 1$  that enforces unit-length gradients, followed by second-order methods like DiGS [Ben-Shabat et al. 2021], which stabilizes training using the divergence, and Neural-Singular-Hessian [Wang et al. 2023], which pushes the network's Hessian toward low rank, which discourages spurious curvature and enforces piece-wise smoothness that matches better with classical differential geometry.

That thread naturally intersects with computer-aided design (CAD). Mechanical parts are dominated by plates, cylinders, cones, and developable blends whose Gaussian curvature is zero almost

everywhere, except for sharp feature curves. Encoding these properties in an SDF prior promises more faithful reconstructions from sparse or noisy scans and cleaner patch decomposition for downstream parametric recovery. NeurCADRecon [Dong et al. 2024] recently embodied that idea: it augments a SIREN network with a “developability” loss that minimizes the magnitude of Gaussian curvature  $|K|$  over a thin shell around the surface, thereby encouraging each learned patch to flatten out unless data dictates otherwise.

Unfortunately, such elegance comes at a price. The Gaussian curvature of an implicit surface depends on all nine entries of the Hessian of the network. In automatic differentiation frameworks this means six Jacobian-vector products and thus seven backward passes per sample, plus the retention of second-order computational graphs. Memory footprints balloon, forcing small batches; iteration times rise, limiting practical resolutions.

To address these limitations we propose a novel curvature regularization loss. In particular, **our contributions** are the following:

- **Curvature gap regularization.** We regularize only the off-diagonal Weingarten shape operator term, letting each principal curvature follow the data while flattening hyperbolic and parabolic patches and rounding elliptic ones uniformly. In other words, we minimize the “curvature-gap” which is the difference between the principal curvatures and we denote the resulting regularizer the *off-diagonal Weingarten loss* (ODW-loss). It is described in detail in Section 3.
- **Two implementations.** We propose two variants for computing the regularizer: (i) Finite-differences, which uses forward and backward offsets (six additional SDF queries plus one gradient evaluation) and introduces no second-order graphs; (ii) Auto-differentiation, which employs a single Hessian-vector product obtained with two reverse sweeps, thereby avoiding full-Hessian assembly and running faster in practice. Both are described in detail in Section 4.
- **Efficiency without loss of accuracy.** On the ABC CAD benchmark our proxies match or slightly outperform Hessian-based baselines while roughly halving GPU memory use and training time. The details are presented in Section 5.

Because the method is drop-in and framework-agnostic, it makes curvature-aware neural SDFs practical for large-scale, engineering-grade shape reconstruction. Our implementation is available at <https://flatcad.github.io/>.

## 2 RELATED WORK

*Implicit Surface Representations and Reconstruction.* Implicit representations have emerged as a powerful alternative to explicit surface models due to their continuous and differentiable nature. Reconstructing surfaces from point clouds has long been a fundamental task in computer graphics, and recent years have witnessed a surge in learning-based implicit approaches.

Early implicit methods compute the signed distance to the tangent plane of the closest surface point [Hoppe et al. 1992]. Subsequently, radial basis function (RBF) methods were introduced to fit the zero level-set of the signed distance function, enabling smooth surface reconstruction [Carr et al. 2001; Huang et al. 2019; Li et al. 2016]. Another family of methods formulates reconstruction as solving

partial differential equations, such as Poisson Surface Reconstruction (PSR) [Hou et al. 2022; Kazhdan et al. 2020, 2006; Kazhdan and Hoppe 2013], which computes an occupancy field by solving a Poisson equation. Parametric Gauss Reconstruction (PGR) [Lin et al. 2022] improves normal consistency by leveraging the Gauss formula from potential theory. These methods build surfaces analytically by imposing strong geometric constraints.

In contrast, neural implicit methods learn to map 3D coordinates to continuous scalar fields, representing the surface as the zero level-set of a neural network. DeepSDF [Park et al. 2019] pioneered the use of an auto-decoder that optimizes a latent code per shape using MAP estimation. Supervised learning-based approaches extend this idea by training on datasets with precomputed ground-truth field values, using structured representations such as voxel grids [Chabra et al. 2020; Jiang et al. 2020; Peng et al. 2020], k-nearest neighbor graphs [Boulch and Marlet 2022; Erler et al. 2020], and octrees [Huang et al. 2022; Tang et al. 2021; Wang et al. 2022]. Some methods target open surfaces or unsigned distance fields (UDFs) [Chibane et al. 2020; Ye et al. 2022]. While these methods can reconstruct detailed shapes, they often encode a fixed representation for a family of similar shapes and thus generalize poorly to unseen geometries.

To address generalization and robustness, recent methods directly optimize a neural field from raw point clouds in a self-supervised fashion. These methods typically rely on regularization terms that encode geometric priors. For example, IGR[Gropp et al. 2020] introduces the Eikonal regularization to encourage unit-norm gradients, a defining property of valid distance fields. SAL and SALD[Atzmon and Lipman 2019] adopt sign-agnostic learning to regress signed distances from unsigned data. SIREN [Sitzmann et al. 2020] uses periodic activation functions to preserve high-frequency details.

To further enhance geometric fidelity, higher-order constraints have also been explored. The Hessian matrix of a neural field encodes its second-order derivatives, making it a natural vehicle for expressing differential properties such as surface curvature. DiGS [Ben-Shabat et al. 2021] penalizes the divergence of the gradient field as a soft constraint, encouraging smooth transitions across the surface. Neural-Singular-Hessian [Wang et al. 2023] directly regularizes the Hessian by minimizing the smallest singular value to make it rank-deficient near the surface, promoting developability and reducing spurious oscillations. These second-order constraints have proven effective in restoring fine-grained geometric structures. Concurrently, other curvature-guided regularizers have been explored for surface INRs [Novello et al. 2023; Sang et al. 2025], and rank-minimization schemes have been proposed to approximate developability [Selvaraju 2024].

*Surface Reconstruction of CAD Models.* In the context of CAD model reconstruction, several approaches have focused on fitting predefined geometric primitives. [Kania et al. 2020] and [Sharma et al. 2017] adopt primitive fitting pipelines, while [Li et al. 2018] employ supervised learning to detect primitive types and fit patches such as planes, cylinders, and spheres. [Sharma et al. 2020] extend this idea to B-spline patches using differentiable segmentation, and [Uy et al. 2021] segment point clouds into extrusion cylinders assembled via Boolean operations. However, these methods often

require voxelized and oriented inputs, which makes them less practical for handling complex CAD assemblies that are scanned as raw, unoriented point clouds.

More recently, implicit neural representations have also been applied to CAD reconstruction under a self-supervised setting. [Dong et al. 2024] proposes a novel formulation that integrates curvature-based regularization into the learning process. Specifically, it introduces a Gaussian curvature loss that encourages zero curvature across the reconstructed surface, based on the assumption that CAD models are composed of piecewise-developable patches. This curvature regularization is expressed through a differentiable formulation involving the Hessian of the neural SDF.

### 3 METHOD

Our goal is to learn a neural signed-distance field  $f$  from an unoriented point cloud such that its zero-set reproduces the target CAD surface. We keep the standard supervision recipe used by implicit reconstruction methods and add a minimal curvature regularizer that removes local warp, flattening developable and saddle parts and rounding elliptic ones—without ever building a full Hessian.

First, we inherit standard supervision terms commonly used in implicit learning: a Dirichlet condition on the zero-level set, as proposed by Atzmon et al. [Atzmon and Lipman 2020], and the Eikonal constraint [Gropp et al. 2020], which enforces unit gradient norms in the vicinity of the surface. These terms ensure fidelity to the input data and preserve geometric consistency.

Second, and most importantly, we introduce a novel curvature regularizer that suppresses the difference between the principal curvatures of the learned surface. This regularization is particularly effective for CAD surfaces, as it forces parabolic and hyperbolic surface parts to be planar and elliptic areas to be umbilical (cf. Figure 2). In combination with the surface reconstruction (Dirichlet) and unit gradient field (Eikonal) losses, it promotes flatness where the target is hyperbolic and parabolic, and spherical uniformity where it is doubly curved.

Unlike previous methods that construct the full  $3 \times 3$  Hessian [Dong et al. 2024; Wang et al. 2023], our formulation needs only a single Hessian–vector product per sample, eliminating explicit Hessian assembly and speeding up training. Additionally, we also propose a first-order finite-difference approximation, which avoids Hessian computation entirely, enabling significantly faster training while retaining the geometric benefits of curvature regularization.

#### 3.1 Signed-Distance Fields and Surface Geometry

We represent a surface  $\mathcal{M} \subset \mathbb{R}^3$  implicitly as the zero-level set of a signed-distance field (SDF)

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad \mathcal{M} = \{\mathbf{x} \mid f(\mathbf{x}) = 0\}.$$

Because a true SDF satisfies  $\|\nabla f\| = 1$  in a neighborhood of  $\mathcal{M}$ , its *normal direction is locally linear*:

$$\partial_{nn} f = \mathbf{n}^\top H_f \mathbf{n} = 0, \quad \mathbf{n} = \frac{\nabla f}{\|\nabla f\|},$$

where  $H_f = [\partial_{x_i} \partial_{x_j} f]$  is the Hessian of the SDF. The shape-operator (Weingarten map) is obtained by projecting  $H_f$  into any orthonormal tangent frame  $(\mathbf{u}, \mathbf{v})$ :

$$S = \begin{pmatrix} \mathbf{u}^\top H_f \mathbf{u} & \mathbf{u}^\top H_f \mathbf{v} \\ \mathbf{v}^\top H_f \mathbf{u} & \mathbf{v}^\top H_f \mathbf{v} \end{pmatrix}.$$

That operator (Weingarten map) takes any tangent vector and returns how fast the surface normal rotates in that direction, i.e., the local bending. Its eigenvalues are the *principal curvatures*  $\kappa_1, \kappa_2$ ; their product gives the Gaussian curvature  $K = \kappa_1 \kappa_2 = \det S$ . Together, these values distinguish spherical, planar, parabolic, elliptic, and hyperbolic regimes as illustrated in Fig. 2.

*Rotation of  $S$  and the Mixed Entry.* In the principal frame the shape operator is

$$S_0 = \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix}.$$

We can rotate a tangent basis by an angle  $\theta$  through

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \quad R^\top R = I.$$

The operator in the rotated frame is the orthogonal transform

$$S(\theta) = R^\top S_0 R = \begin{pmatrix} \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta & (\kappa_2 - \kappa_1) \sin \theta \cos \theta \\ (\kappa_2 - \kappa_1) \sin \theta \cos \theta & \kappa_1 \sin^2 \theta + \kappa_2 \cos^2 \theta \end{pmatrix}.$$

The off-diagonal component is

$$S_{12}(\theta) = (\kappa_2 - \kappa_1) \sin \theta \cos \theta = \frac{1}{2}(\kappa_2 - \kappa_1) \sin 2\theta. \quad (1)$$

Thus  $S_{12}$  is a *warp* measure: it vanishes in a principal frame and is proportional to the curvature difference in any other frame.

#### 3.2 Maximum Likelihood Estimation

During training we penalize  $S_{12}^2$  (or  $|S_{12}|$ , see below), in particular, for each point we sample a fresh random frame angle  $\theta$  each iteration (which in practice is equivalent to sampling a fresh random vector  $\mathbf{u}$  orthogonal to  $\mathbf{n}$ , and completing the frame with  $\mathbf{v}$  using the cross product).

Taking the expectation of  $S_{12}^2$  over  $\theta \in [0, 2\pi]$  yields

$$\mathbb{E}_\theta[S_{12}^2(\theta)] = \frac{(\kappa_2 - \kappa_1)^2}{4} \mathbb{E}_\theta[\sin^2 2\theta].$$

Because

$$\mathbb{E}_\theta[\sin^2 2\theta] = \frac{1}{2\pi} \int_0^{2\pi} \sin^2 2\theta d\theta = \frac{1}{2},$$

we obtain

$$\mathbb{E}_\theta[S_{12}^2] = \frac{1}{8}(\kappa_2 - \kappa_1)^2. \quad (2)$$

Equation (2) vanishes if  $\kappa_1 = \kappa_2$ . Hence gradient descent on  $S_{12}^2$  systematically *suppresses the curvature gap* without directly pushing either curvature to zero.

Because the frame angle is resampled independently as  $\theta \sim \mathcal{U}[0, 2\pi]$ , the off-diagonal Weingarten entry  $S_{12}$  is a zero-mean random variable. The mini-batch average of  $S_{12}^2$  (or  $|S_{12}|$ ) is thus an unbiased Monte-Carlo estimate of its expectation over  $\theta$ . If the measurement noise on  $S_{12}(\theta)$  is assumed Gaussian with variance  $\sigma^2$ , the negative log-likelihood reduces—up to a constant—to

$$\frac{1}{2\sigma^2} \sum S_{12}(\theta)^2.$$

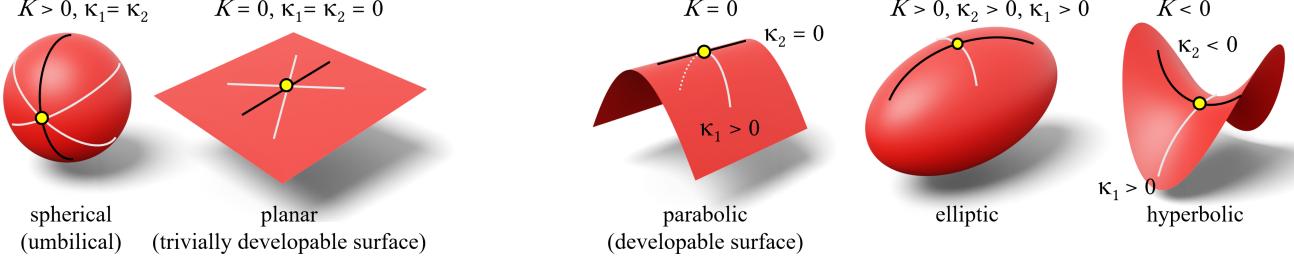


Fig. 2. Principal curvatures  $\kappa_1$  and  $\kappa_2$  are the eigenvalues of the Weingarten map  $S$ . Their product gives the Gaussian curvature  $K = \kappa_1\kappa_2 = \det S$ . Because  $S$  is self-adjoint with respect to the first fundamental form, its eigenvectors (the principal directions) are orthogonal in the surface metric whenever the eigenvalues are distinct. At an umbilic point (e.g. on a sphere) the entire two-dimensional tangent plane is the eigenspace, so no unique pair of principal directions exists. For a plane the second fundamental form vanishes, i.e.,  $S = 0$ .

Minimizing the squared off-diagonal entry is therefore the maximum-likelihood estimator for the zero-warp hypothesis. Consequently, the  $L^2$  formulation possesses a clear maximum-likelihood interpretation under the common Gaussian-noise assumption.

*Expectation of the absolute value.* Adopting a Laplace noise model instead yields the absolute-value, with the same Monte-Carlo interpretation. With  $\theta \sim \text{Uniform}[0, 2\pi]$ ,

$$\begin{aligned} \mathbb{E}_\theta[|S_{12}(\theta)|] &= |\kappa_2 - \kappa_1| \mathbb{E}_\theta[|\sin \theta \cos \theta|] \\ &= |\kappa_2 - \kappa_1| \frac{1}{2\pi} \int_0^{2\pi} |\sin \theta \cos \theta| d\theta. \end{aligned}$$

Evaluating the integral:

$$\int_0^{2\pi} |\sin \theta \cos \theta| d\theta = \frac{1}{2} \int_0^{2\pi} |\sin 2\theta| d\theta = \frac{1}{2} \times 4 = 2$$

yields

$$\mathbb{E}_\theta[|S_{12}|] = \frac{1}{\pi} |\kappa_2 - \kappa_1|. \quad (3)$$

While the squared Weingarten loss has elegant Gaussian-noise and variance interpretations, in a SIREN setting the practical stability, edge preservation, and faster optimization of the absolute-value off-diagonal Weingarten loss generally outweigh the theoretical neatness of  $L^2$ .

SIREN network's high-frequency sine layers can generate very large local gradients; employing the absolute ( $L^1$ ) loss caps each sample's gradient at  $\pm 1$ , preventing exploding updates while preserving sharp detail, and thus trains more stably than the squared ( $L^2$ ) version.

### 3.3 Hessian Deficiency

Evaluating  $S_{12}$  directly on the zero level-set is numerically unstable: the Hessian is rank-deficient because  $\partial_n f = 0$ . In fact, since the Eikonal condition enforces  $\|\nabla f\| = 1$ , differentiating this constraint along the normal direction yields

$$\partial_n \left( \frac{1}{2} \|\nabla f\|^2 \right) = (H_f \mathbf{n}) \cdot \nabla f = \mathbf{n}^\top H_f \mathbf{n} = 0,$$

which means that one eigenvalue of  $H_f$  must vanish, making the Hessian singular on the surface. This property of signed distance fields is also emphasized by Wang et al. [Wang et al. 2023], who use it as the basis for their Neural-Singular-Hessian regularizer.

To address the numerical instability, we follow the sampling strategy of Dong et al. [Dong et al. 2024]. At each iteration, we sample a static shell  $\Omega$  by drawing one point around every input point  $p$  from a 3D Gaussian whose standard deviation equals the distance to its 50-th nearest neighbor, yielding a fixed batch of 15k near-surface points. We then evaluate  $S_{12}$  on  $\Omega$ , which provides stable signals while leaving the standard data and Eikonal terms unchanged.

The off-diagonal Weingarten loss is hence a minimal, orientation-free condition that suppresses the curvature difference  $(\kappa_2 - \kappa_1)^2$ , thereby isolating the spurious “warp” defect and flattening hyperbolic and parabolic areas.

Because the loss depends only linearly on this mixed second derivative, rather than on quadratic combinations of Hessian entries, it yields well-conditioned, low-variance gradients that integrate seamlessly with standard SDF and Eikonal losses, mitigating the exploding sensitivities observed with many quadratic-or-higher curvature regularizers.

### 3.4 Loss Function Components

Our training objective combines four terms that constrain the signed-distance field both on and off the surface. The total loss is

$$\mathcal{L}_{\text{total}} = \lambda_{\text{DM}} \mathcal{L}_{\text{DM}} + \lambda_{\text{DNM}} \mathcal{L}_{\text{DNM}} + \lambda_{\text{EIK}} \mathcal{L}_{\text{EIK}} + \lambda_{\text{ODW}} \mathcal{L}_{\text{ODW}}, \quad (4)$$

with scalar weights  $\lambda_{\text{DM}}$ ,  $\lambda_{\text{DNM}}$ ,  $\lambda_{\text{EIK}}$ , and  $\lambda_{\text{ODW}}$  balancing the contributions of the three regularizers against the data term.

*Manifold Loss.* For the  $N$  input points  $\{\mathbf{x}_i\}_{i=1}^N$  sampled on or very near the target surface, we impose a Dirichlet condition that pulls their SDF values to 0:

$$\mathcal{L}_{\text{DM}} = \frac{1}{N} \sum_{i=1}^N |f(\mathbf{x}_i)|. \quad (5)$$

This term anchors the learned zero-level set to the point cloud and provides first-order geometric fidelity.

*Non-Manifold Loss.* For  $M$  free-space samples  $\{\mathbf{y}_j\}_{j=1}^M$  we adopt the sign-agnostic formulation of Atzmon et al. [Atzmon and Lipman 2020]:

$$\mathcal{L}_{\text{DNM}} = \frac{1}{M} \sum_{j=1}^M \exp(-\alpha |f(\mathbf{y}_j)|), \quad \alpha = 100. \quad (6)$$

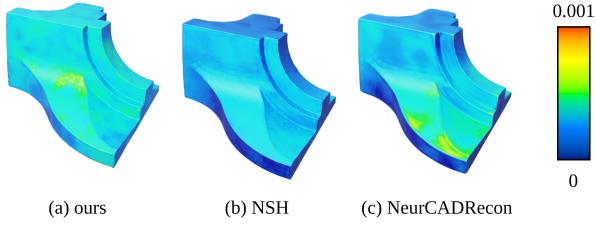


Fig. 3. Hausdorff-distance heat maps (blue → low, yellow → high) for the fandisk model reconstructed with ours, NSH, and NeurCADRecon.

The exponential quickly attenuates the penalty as  $|f|$  grows, suppressing spurious oscillations close to the surface while allowing the SDF to expand smoothly in regions devoid of geometric evidence, thereby stabilizing training and preventing distant artifacts.

*Eikonal Loss.* To preserve the metric property of a true distance field we enforce the Eikonal equation on  $K$  uniformly sampled points  $\{\mathbf{z}_k\}_{k=1}^K$ :

$$\mathcal{L}_{EIK} = \frac{1}{K} \sum_{k=1}^K \left( \|\nabla f(\mathbf{z}_k)\|_2 - 1 \right)^2. \quad (7)$$

This global term regularizes gradient norms, stabilizing optimization and improving extrapolation away from the data [Gropp et al. 2020].

*Off-Diagonal Weingarten Loss (ours).* Let  $S_{12}(\theta)$  denote the off-diagonal entry of the  $2 \times 2$  shape operator obtained by projecting the Hessian onto a tangent frame rotated by a random angle  $\theta \sim \mathcal{U}[0, 2\pi]$ . For  $L$  near-surface samples  $\{\mathbf{p}_\ell\}_{\ell=1}^L$  we penalize the *absolute* curvature gap,

$$\mathcal{L}_{ODW} = \frac{1}{L} \sum_{\ell=1}^L |S_{12}(\theta_\ell)|. \quad (8)$$

The mixed second derivative  $S_{12}$  is evaluated either with a single Hessian–vector product or via the symmetric finite-difference approximation, avoiding explicit Hessian assembly.

Minimizing  $|S_{12}|$  drives the principal curvatures toward equality, flattening parabolic and hyperbolic regions and rounding elliptic regions uniformly. In all benchmarks the absolute-value variant yields the most stable optimization and the lowest validation error, so we use it by default in all experiments.

**Weights.** We keep most parameter weights identical to the settings recommended in NeurCADRecon:  $\lambda_{DM} = 7000$ ,  $\lambda_{DNM} = 600$ ,  $\lambda_{EIK} = 50$ , and  $\lambda_{ODW} = 10$ .

## 4 COMPUTATIONAL PATHWAYS

The mixed-curvature term  $S_{12}$  can be evaluated with or without explicit second-order auto-differentiation, giving two practical routes that trade accuracy for running cost. The finite-difference minimizes memory use, whereas the auto-diff variant offers higher accuracy. Both achieve the curvature-gap regularization without assembling the full Hessian.

### 4.1 Off-Diagonal Weingarten Loss via Taylor Expansion

When raw throughput or framework simplicity is paramount we approximate the mixed derivative with a symmetric finite-difference stencil. This stencil evaluates the SDF at forward and backward offsets of magnitude  $h$ , averages the two estimates, and therefore incurs a truncation error of order  $h^2$ . If only the one-sided (forward) stencil is used, the truncation error is  $O(h)$  and the estimate is biased. Our default uses the symmetric average which cancels the odd powers and yields  $O(h^2)$  accuracy.

Let  $x_\Omega$  be any point in the near-surface shell  $\Omega$ , and let  $\mathbf{u}, \mathbf{v}$  form an orthonormal tangent frame at  $x_\Omega$ . For a small step  $h > 0$  define the forward offsets

$$f_{00} = f(x_\Omega), \quad f_{\mathbf{u}}^+ = f(x_\Omega + h \mathbf{u}), \\ f_{\mathbf{v}}^+ = f(x_\Omega + h \mathbf{v}), \quad f_{\mathbf{uv}}^+ = f(x_\Omega + h (\mathbf{u} + \mathbf{v})),$$

and the backward offsets

$$f_{\mathbf{u}}^- = f(x_\Omega - h \mathbf{u}), \quad f_{\mathbf{v}}^- = f(x_\Omega - h \mathbf{v}), \\ f_{\mathbf{uv}}^- = f(x_\Omega - h (\mathbf{u} + \mathbf{v})).$$

Define the one-sided mixed differences and their symmetric average

$$D_{\mathbf{uv}}^{(+)} = \frac{f_{\mathbf{uv}}^+ - f_{\mathbf{u}}^+ - f_{\mathbf{v}}^+ + f_{00}}{h^2}, \\ D_{\mathbf{uv}}^{(-)} = \frac{f_{\mathbf{uv}}^- - f_{\mathbf{u}}^- - f_{\mathbf{v}}^- + f_{00}}{h^2}, \\ D_{\mathbf{uv}}^{(c)} = \frac{1}{2} \left( D_{\mathbf{uv}}^{(+)} + D_{\mathbf{uv}}^{(-)} \right).$$

**Taylor justification (order and bias).** Assume  $f \in C^3$  near  $x_\Omega$ . Using

$$f(x_\Omega + ha) = f_{00} + h \nabla f^\top \mathbf{a} + \frac{1}{2} h^2 \mathbf{a}^\top H_f \mathbf{a} + O(h^3),$$

with  $\mathbf{a} \in \{\mathbf{u}, \mathbf{v}, \mathbf{u} + \mathbf{v}\}$ , we obtain

$$f_{\mathbf{uv}}^+ - f_{\mathbf{u}}^+ - f_{\mathbf{v}}^+ + f_{00} = \frac{1}{2} h^2 [(\mathbf{u} + \mathbf{v})^\top H_f (\mathbf{u} + \mathbf{v}) - \mathbf{u}^\top H_f \mathbf{u} - \mathbf{v}^\top H_f \mathbf{v}] \\ + O(h^3) \\ = h^2 \mathbf{u}^\top H_f \mathbf{v} + O(h^3),$$

hence  $D_{\mathbf{uv}}^{(+)} = \mathbf{u}^\top H_f \mathbf{v} + O(h)$ . Replacing  $h \rightarrow -h$  gives  $D_{\mathbf{uv}}^{(-)} = \mathbf{u}^\top H_f \mathbf{v} + O(h)$ . Averaging cancels odd powers:

$$D_{\mathbf{uv}}^{(c)} = \mathbf{u}^\top H_f(x_\Omega) \mathbf{v} + O(h^2). \quad (9)$$

The off-diagonal entry of the  $2 \times 2$  shape operator in the  $(\mathbf{u}, \mathbf{v})$  frame is

$$S_{12}(x_\Omega) = \frac{\mathbf{u}^\top H_f(x_\Omega) \mathbf{v}}{\|\nabla f(x_\Omega)\|}.$$

Using the finite-difference approximation gives the off-diagonal Weingarten loss

$$|S_{12}(x_\Omega)| \approx \frac{|D_{\mathbf{uv}}^{(c)}(x_\Omega)|}{\|\nabla f(x_\Omega)\|} + O(h^2), \quad (10)$$

and the quantity used in the squared loss becomes

$$S_{12}^2(x_\Omega) \approx \frac{(D_{\mathbf{uv}}^{(c)}(x_\Omega))^2}{\|\nabla f(x_\Omega)\|^2} + O(h^2). \quad (11)$$

In practice we may omit the division by  $\|\nabla f(x_\Omega)\|$  and minimize  $|D_{\mathbf{uv}}^{(c)}|$  directly: because the Eikonal loss enforces  $\|\nabla f\| \approx 1$  in

the shell, both definitions coincide up to a small scale factor. This symmetric stencil converts the curvature-gap loss into a first-order computation.

#### 4.2 Off-Diagonal Weingarten Loss via Auto-Diff

For applications that favor numerical precision we exploit reverse-mode auto-diff to obtain a Hessian–vector product

$$\mathbf{u}^\top H_f \mathbf{v} = \mathbf{u}^\top \nabla_{\mathbf{x}} [(\nabla_{\mathbf{x}} f) \cdot \mathbf{v}].$$

From a shell point  $x_\Omega$  with orthonormal tangents  $\mathbf{u}, \mathbf{v}$  (Sec. 4.1) we first run reverse mode on  $f(x_\Omega)$  to obtain the gradient  $\mathbf{g} = \nabla f(x_\Omega)$ . Treating the scalar  $g_v = \mathbf{g}^\top \mathbf{v}$  as a new objective and invoking reverse mode a second time differentiates  $g_v$  with respect to  $x_\Omega$ ; the result is the Hessian–vector product  $H_f(x_\Omega)\mathbf{v}$ . An inner product with  $\mathbf{u}$  then yields  $\mathbf{u}^\top H_f(x_\Omega)\mathbf{v}$ , and dividing by  $\|\nabla f(x_\Omega)\|$  gives the exact off-diagonal entry  $S_{12}(x_\Omega)$ .

The first backward sweep delivers  $\nabla f$ ; a second sweep differentiates the scalar inner product  $(\nabla f) \cdot \mathbf{v}$  with respect to the input coordinates, giving the exact mixed second derivative without materializing the full  $3 \times 3$  Hessian. This doubles the backward sweeps for the Weingarten loss samples and avoids the prohibitive memory footprint of full Hessian assembly.

This Hessian–vector product therefore requires only two reverse-mode sweeps and stores no matrix entries, so its cost grows linearly with the number of shell samples and is independent of the ambient dimension. Dividing by  $\|\nabla f(x_\Omega)\|$  provides the exact  $S_{12}(x_\Omega)$  value used in the off-diagonal loss. Although more expensive than the first-order finite-difference loss, this second-order route offers higher accuracy and serves as our reference implementation for ablation studies.

## 5 EXPERIMENTS

In this section, we first describe the implementation details and evaluation metrics used in our validation. We then present a comprehensive comparison of our proposed off-diagonal Weingarten based loss functions (ODW-AD and ODW-FD) against three baseline methods—DiGS [Ben-Shabat et al. 2021], NSH [Wang et al. 2023], and the state-of-the-art NeurCADRecon [Dong et al. 2024]—on two subsets of the ABC dataset [Koch et al. 2019]. Finally, we analyze convergence behavior and runtime performance.

### 5.1 Experimental Setup

**Datasets.** All experiments are conducted on two 100-example subsets of the ABC dataset [Koch et al. 2019]. The first subset (1 MB set) is a pseudo-random selection of 100 CAD models (each  $\approx 1$  MB in file size), while the second subset (5 MB set) comprises 100 more complex human-chosen models (between 3.5 MB and 9.5 MB, in average  $\approx 5$  MB) selected for their clean topology and well-defined features. For each mesh, we uniformly sample a total of 30,000 points to form the input point cloud. This preprocessing strategy ensures that all methods receive identical input distributions. After that, at each iteration 20,000 points are randomly chosen from the 30,000 points. Additional 20,000 non-manifold points are drawn via spatial sampling within the mesh’s bounding volume.

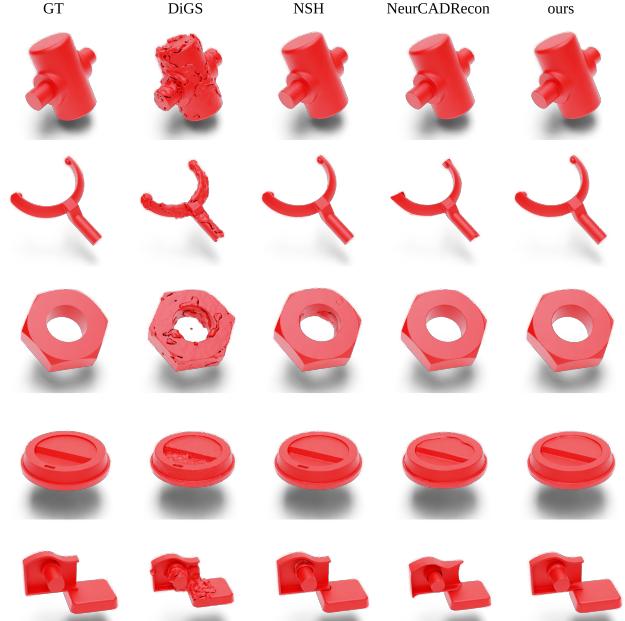


Fig. 4. Comparison with state-of-the-art surface reconstruction. Our method and NeurCADRecon demonstrate superior performance in recovering clean, complete, and geometrically accurate surfaces, benefiting from curvature-aware regularization that encourages developability. Notably, our approach achieves comparable reconstruction quality while requiring only half the training time of NeurCADRecon.

**Methods and Experimental Setup.** We compare two variants of our off-diagonal loss: ODW-AD (Auto-Diff) and ODW-FD (Finite-Diff) against three baselines: DiGS [Ben-Shabat et al. 2021], Neural-Singular-Hessian (NSH) [Wang et al. 2023] and NeurCADRecon (NCR) [Dong et al. 2024]. All methods have been implemented within a unified comparison framework to eliminate discrepancies arising from differing network architectures or training pipelines. Each method employs a SIREN MLP architecture [Sitzmann et al. 2020] with four hidden layers of 256 units and sine activations. All models use a standard SIREN initialization, except DiGS, which uses the MFGI initialization. Training is performed using the Adam optimizer [Kingma and Ba 2017] with a learning rate of  $5 \times 10^{-5}$  for up to 10,000 iterations. Early stopping is triggered if the Chamfer-Distance metric plateaus for 1500 consecutive iterations. All other hyperparameters, such as weight coefficients for curvature or normal losses, and other method specific parameters follow the settings from the original implementations.

**Hardware.** Runtime and convergence experiments on the 1MB set are performed on an NVIDIA H100 GPU. Additional qualitative and quantitative experiments (e.g., on the 5 MB set) are run on NVIDIA A100 and T4 GPUs, but only H100 results are included in the time-comparison analysis. All machines are equipped with at least 32GB of RAM.

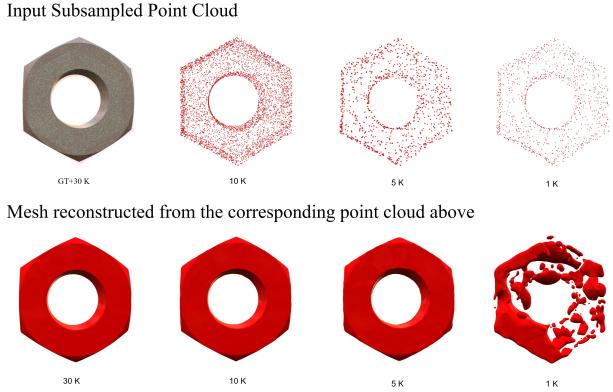


Fig. 5. Qualitative results on point clouds with varying levels of sparsity. We use the original input with 30k points sampled from the ground truth mesh, and generate subsampled versions with 10k, 5k, and 1k points. The top row shows the input point clouds, and the bottom row shows the corresponding reconstructed meshes.

## 5.2 Evaluation Metrics

*Quantitative Metrics.* We evaluate reconstruction accuracy using three standard metrics. Chamfer Distance (CD), scaled by  $10^3$ , which measures the similarity between two surfaces; lower values indicate better accuracy. F1 Score (F1) is the harmonic mean of precision and recall, computed using a distance threshold of  $5 \times 10^{-3}$  between the predicted and ground-truth point sets. The results are scaled by  $10^2$ , with higher values indicating a better overlap. Normal Consistency (NC) captures the mean cosine similarity between predicted and ground-truth normals, scaled by  $10^2$ ; higher values denote better alignment. For each of these metrics, we report the mean and standard deviation across all 100 shapes in each data subset. Additionally, for the 1MB subset, we include additional efficiency measures: GPU VRAM usage (in GB), Time per iteration (in milliseconds), and Best Iteration achieved during evaluation—the iteration count at which the best CD metric is reached. We utilize the time measurements and best iteration count to calculate the convergence time.

## 5.3 Qualitative Results

Beyond aggregate scores, the spatial distribution of reconstruction error is inspected in Figure 3 on a Hausdorff-distance heat map. Our method markedly reduces peak deviations and suppresses local error spikes compared with the baselines. Other qualitative results are shown in Figure 1 and in the collection in Figure 10. We also compare our method with other state-of-the-art reconstruction methods, depicted in Figure 4, which shows our approach achieves comparable reconstruction quality with NeurCADRecon [Dong et al. 2024].

## 5.4 Input Data Experiments

*Data Sparsity.* To assess the robustness of our method under reduced sampling density with sparse input, we simulate sparse inputs by randomly subsampling 10k, 5k, and 1k of the originally sampled 30k point cloud data. This scenario mimics real-world settings such as thin-walled plates, tubes, or surfaces with large

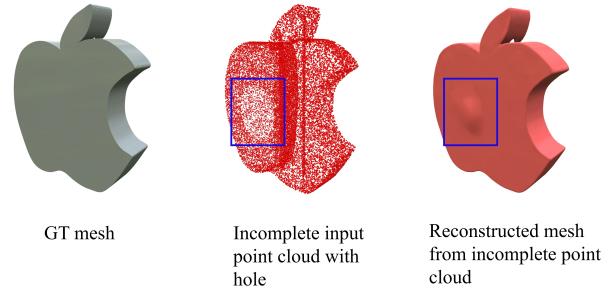


Fig. 6. Using incomplete point cloud as input, our method is able to plausibly fill in missing planar regions while preserving global surface continuity and topological correctness.

inter-point gaps, where predicting accurate SDF values becomes challenging.

As shown in both Figure 5 and quantitative Table 1 evaluations, our method successfully reconstructs geometry of comparable quality to that obtained from the full-resolution input. Despite the significant reduction in point density, the reconstructed shapes remain smooth, topologically correct, and well-aligned with the ground truth, without introducing spurious genus or holes.

Our method maintains high-fidelity reconstruction at 10k and 5k, while severe degradation appears at 1k due to extreme sparsity.

*Incomplete Point Clouds.* To evaluate robustness to missing regions, a circular patch on the surface is excised from each input cloud while keeping the remaining samples unchanged. As reported in Table 2 and Figure 6, the Weingarten loss-based reconstruction degrades gracefully: Chamfer distance increases by only 64 % and normal consistency drops by 0.7 %, yet global topology and salient features are preserved, however, a visible bump remains.

*Generalization to Non-CAD Shapes.* Our method is specifically designed for CAD data, where the Weingarten loss discourages

Table 1. Quantitative results on sparse point-cloud data with 10k, 5k, and 1k points sub-sampled from the original 30k-point cloud generated from the ground-truth mesh.

Input Size	NC $\uparrow$	CD $L_1$	F1 $\uparrow$
30k	0.9936044	0.0024629	0.9693
10k	0.9926978	0.0024399	0.970385
5k	0.9917300	0.0024912	0.9693145
1k	0.7255813	0.0143693	0.3399991

Table 2. Quantitative comparison on incomplete data with hole for apple shape.

input	NC $\uparrow$	CD $L_1$ $\downarrow$	F1 $\uparrow$
Full	0.9897844	0.0029810	0.9334532
Hole	0.9824422	0.0048818	0.5045011

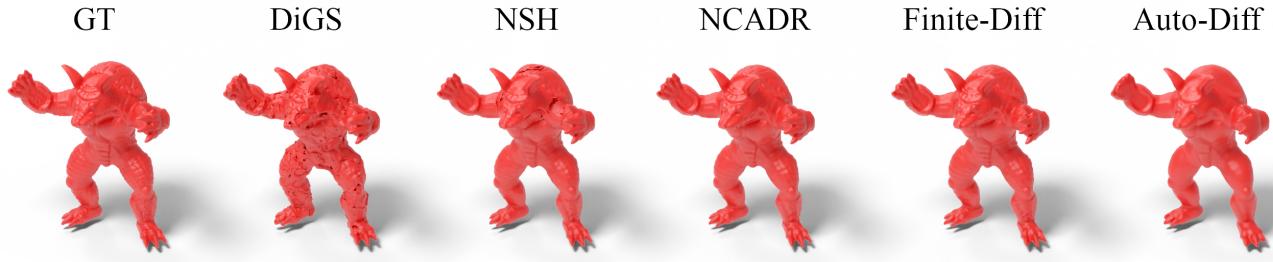


Fig. 7. Reconstruction results on a non-CAD shape (Armadillo). Our method and NeurCADRecon achieve smooth and complete surfaces with fewer artifacts than DiGS and NSH. While fine details are slightly smoothed, our approach offers a good balance between quality and efficiency, with only half the training time of NeurCADRecon.

the reconstructed surface to be locally warped and twisted. To understand how this inductive bias affects performance on general, non-CAD geometry, we evaluate our model on the standard Armadillo shape, which contains rich organic, but also high frequency, details.

As shown in Figure 7 and Table 3, our method produces smooth and topologically consistent reconstructions, comparable to those generated by NeurCADRecon [Dong et al. 2024], but at approximately half the training time. Compared to DiGS [Ben-Shabat et al. 2021] and Neural-Singular-Hessian (NSH) [Wang et al. 2023], both our method and NeurCADRecon yield superior global shape recovery, exhibiting fewer holes and more coherent topology. However, we observe that fine surface details are partially smoothed out—an expected consequence of the curvature regularization, which favors larger planar or uniformly curved regions.

This experiment illustrates that, while our method is tailored for CAD reconstruction, it remains robust on general shapes and offers a favorable trade-off between geometric fidelity, smoothness, and computational efficiency.

### 5.5 Ablation Studies

*Ablation of Weingarten Loss Weight.* To assess the impact of our proposed off-diagonal Weingarten loss regularization, we conduct an ablation study by varying the weight  $\lambda_{\text{ODW}}$  applied to the loss term. Specifically, we test settings with  $\lambda_{\text{ODW}} \in \{0.1, 1, 10, 100\}$  while keeping all other parameters fixed.

As shown in the quantitative results in Table 4 and qualitative comparisons in Figure 8, our method consistently produces smooth and structurally faithful reconstructions across a wide range of weights. Importantly, when the curvature term is removed entirely

( $\lambda_{\text{curv}} = 0$ ), the reconstruction quality drops significantly—exhibiting surface noise, distorted geometry, or incorrect topology in challenging regions. These observations confirm that even a small weight on the curvature loss is sufficient to improve reconstruction quality. The results highlight the importance of our Weingarten loss formulation in guiding the implicit surface toward CAD-compliant characteristics.

*Comparison of Finite-Diff and Auto-Diff.* Our method supports two implementations of the curvature Weingarten loss: an *Auto-Diff* version that computes the mixed second derivative analytically via backward-mode auto-differentiation, and a *Finite-Diff* version that approximates the same quantity using a symmetric finite-difference stencil.

To compare these alternatives, we evaluate both their reconstruction quality and runtime performance. As shown in Table 5, Figure 9, the two variants produce nearly identical reconstruction accuracy, suggesting that the finite-difference approximation is sufficiently precise when the neighborhood resolution we choose is small. This validates the use of our discrete formulation as a numerically stable and effective off-diagonal Weingarten loss.

In terms of runtime, we observe comparable training speeds between the two variants. While the Auto-Diff approach incurs additional backward passes (e.g., two extra reverse sweeps per batch), the Finite-Diff version relies on multiple forward evaluations (six additional SDF queries for the symmetric stencil; seven total including  $f_{00}$  if it is not reused). On networks of our scale, the difference in wall-clock performance remains negligible, further supporting the practicality of our lightweight first-order implementation.

Table 3. Quantitative results on armadillo reconstruction with different methods.

input	NC $\uparrow$	CD $L_1$ $\downarrow$	F1 $\uparrow$	time (s)
Our (ODW-AD)	0.9726746	0.0025855	0.9724751	877.15
Our (ODW-FD)	0.9799343	0.0023597	0.9877244	915.50
NeurCADRecon	0.9789649	0.0022634	0.9898299	1891.00
NSH	0.9615119	0.0056421	0.8860692	231.30
DiGS	0.9337240	0.0045840	0.6731318	714.34

Table 4. Quantitative results with different weight settings.

	NC $\uparrow$	CD $L_1$ $\downarrow$	F1 $\uparrow$
Weight = 100	0.9776159	0.0035313	0.854874
Weight = 10	0.9939355	0.0024900	0.969770
Weight = 1	0.9940296	0.0024305	0.9708199
Weight = 0.1	0.9920184	0.0025440	0.9596963

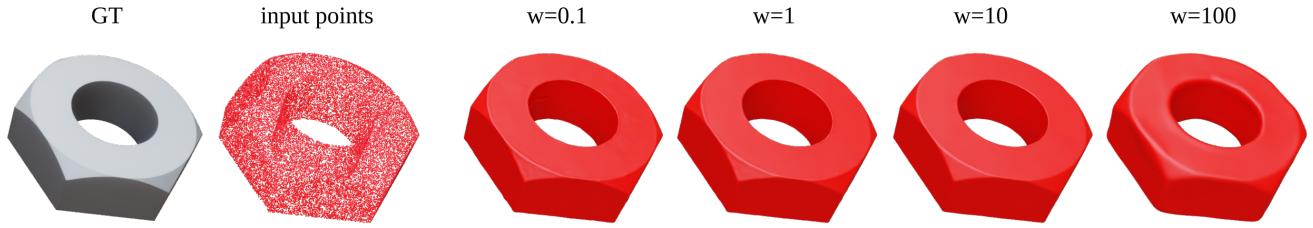


Fig. 8. We compare the reconstructed shape with different Weingarten-loss weight. From our result, our method consistently produces smooth and structurally faithful reconstructions across a wide range of weights.

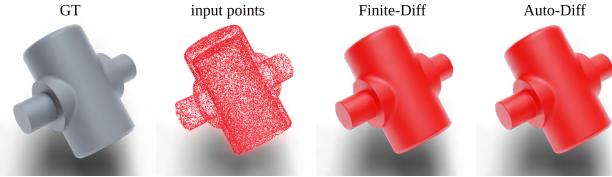


Fig. 9. Ablation between Auto-Diff and Finite-Diff versions of our Weingarten loss. Both methods produce equally high-quality reconstructions, validating the effectiveness of the finite-difference approximation.

## 5.6 Quantitative Results

*Human-selected Examples.* On the cleaner 5 MB subset, see Table 7, both Weingarten variants exhibit strong performance across all metrics. ODW-FD achieves the lowest Chamfer Distance (CD), outperforming NeurCADRecon, while still keeping the Normal Consistency (NC) value very competitive. ODW-AD achieves an F1 score only second to NeurCADRecon, while keeping the variance lower. These results highlight the ability of both Weingarten loss methods to generalize well on well-curated data, with ODW-FD showing the most consistent geometric accuracy, and ODW-AD providing the highest detection fidelity. Figure 10 depicts a few selected models of the 5 MB set.

*Pseudo-Random Examples.* In the more varied 1 MB subset, see Table 7, where input meshes are less structured, ODW-FD leads in geometric accuracy with the lowest CD and a strong NC, closely trailing NeurCADRecon. F1 scores remain competitive, with ODW-FD and ODW-AD closely rivaling NeurCADRecon. These findings demonstrate that even with varied input point clouds, both ODW-FD and ODW-AD deliver reconstructions that are either on par with or superior to NCR, and significantly outperform DiGS and NSH across all metrics.

Table 5. Comparison between Auto-Diff and Finite-Diff methods.

	NC ↑	CD <sub>L<sub>1</sub></sub> ↓	F1 ↑	time (s)
ODW-AD	0.9944662	0.0029025	0.9127199	875.19
ODW-FD	0.9939950	0.0029577	0.9105444	909.76

Table 6. Comparison of iteration time (ms), convergence time (s)—computed as mean iteration time × number of iterations, scaled to seconds—and GPU memory usage. Within each column, the best is **bold and underlined**, second-best is **bold**.

	Iter. time (ms)		iter	time (s)	(GB)
	mean	std.			
DiGS [Ben-Shabat et al. 2021]	4.14	<b>0.03</b>	8769	363.04	3.89
NSH [Wang et al. 2023]	6.11	0.06	8249	504.01	6.10
NCR [Dong et al. 2024]	4.11	<b>0.02</b>	8129	334.10	6.16
<b>ODW-AD</b>	<b>2.50</b>	<b>0.03</b>	7552	<b>188.80</b>	<b>3.46</b>
<b>ODW-FD</b>	<b>3.09</b>	0.05	<b>5519</b>	<b>170.54</b>	<b>3.70</b>

*Efficiency.* In terms of computational efficiency, see Table 6, both of our methods clearly outperform the baselines in terms of iteration time ( $\approx 2\times$  speedup over NCR). The difference is even more apparent when considering the overall training convergence time ( $\approx 2\times$  speedup over NCR). ODW-AD offers a strong trade-off with the second-lowest iteration count and runtime, while consuming modest GPU memory. Compared to NeurCADRecon’s long runtime and high memory footprint, both Weingarten methods are significantly more efficient. This demonstrates that ODW-FD and ODW-AD not only offer high-quality reconstructions but do so with favorable resource demands, making them well suited for scalable and time-sensitive applications.

## 6 DISCUSSION AND CONCLUSIONS

*Discussion.* The proposed off-diagonal Weingarten loss term minimizes the expectation  $\mathcal{L}_{\text{ODW}} = \mathbb{E}_\theta[S_{12}^2] \propto (\kappa_2 - \kappa_1)^2$ , so gradient descent drives the two principal curvatures to coincide. Where one curvature is already zero (parabolic zones) or of opposite sign (hyperbolic zones) the only viable limit is  $\kappa_1 = \kappa_2 = 0$ , hence those patches relax to planes. Where both curvatures share a sign (elliptic zones) equality produces an umbilic state and the patch rounds into a spherical cap whose radius is fixed by the data term. In this way the loss selectively flattens developable and saddle parts while regularizing doubly-curved areas without erasing their overall shape.

Because the Weingarten loss penalty acts only on the *curvature difference*, it complements rather than competes with the Dirichlet and Eikonal constraints: the Dirichlet term fits the sampled points exactly, the Eikonal term keeps the normal lines straight, and the

Table 7. Quantitative results on the ABC dataset [Koch et al. 2019]. Evaluation is conducted on two resolution subsets (1 MB and 5 MB) using three metrics: Normal Consistency (NC), Chamfer Distance (CD), and F1 score (F1). For each metric’s mean value, the best result is **bold and underlined**, and the second-best is **bold**. NC and F1  $\times 10^2$  and CD  $\times 10^3$

	1 MB set						5 MB set					
	NC ↑		CD ↓		F1 ↑		NC ↑		CD ↓		F1 ↑	
	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.
DiGS [Ben-Shabat et al. 2021]	92.30	5.84	7.21	4.20	47.91	30.25	94.01	5.41	8.47	6.91	62.63	29.47
NSH [Wang et al. 2023]	94.84	6.09	6.73	6.00	64.94	34.32	97.44	2.54	5.27	4.85	86.67	14.46
NCR [Dong et al. 2024]	<b>95.43</b>	<b>4.92</b>	<b>3.92</b>	<b>3.72</b>	<b>87.74</b>	<b>16.33</b>	<b>97.59</b>	<b>2.35</b>	<b>4.99</b>	<b>4.17</b>	<b>88.29</b>	15.65
ODW-AD	93.79	6.80	4.59	3.98	83.80	18.28	97.23	2.96	5.27	4.60	<b>86.82</b>	<b>13.81</b>
ODW-FD	<b>94.69</b>	<b>5.41</b>	<b>3.84</b>	<b>3.44</b>	<b>87.65</b>	<b>16.02</b>	<b>97.46</b>	<b>2.48</b>	<b>4.93</b>	<b>4.38</b>	86.61	<b>13.92</b>

Weingarten loss term eliminates the residual tangential “twist” that is unconstrained by first-order information.

Compared to the Gaussian-curvature loss introduced in previous work [Dong et al. 2024], we showed in our results that, under identical shapes and training settings, our Weingarten loss delivers essentially the same reconstruction accuracy and boundary sharpness while requiring only one Hessian–vector product per sample instead of the multiple second-order evaluations and quadratic combination needed for the Gaussian term, resulting in a markedly lower compute and memory footprint during training.

*Limitations and Future Work.* The off-diagonal Weingarten loss is lightweight and effective, yet several boundaries remain. Because it equalizes the two principal curvatures, it assigns a non-zero penalty to developable but curved primitives such as cylinders and cones; with a balanced weight the data term keeps the correct radius or slope, but an overly strong weight can nudge these patches toward planar or spherical limits. The regularizer acts purely locally in the off-surface shell and carries no notion of long-range patch coherence, so global fairness constraints—keeping opposite faces of a thin sheet parallel, for example—are not guaranteed.

The exact Hessian–vector implementation also doubles the backward pass for Weingarten samples, so the finite-difference stencil remains preferable on memory-constrained hardware, with a truncation error  $O(h^2)$  for the symmetric stencil (or  $O(h)$  if only the one-sided stencil is used). Finally, the Weingarten loss has no explicit mechanism for preserving sharp feature lines: narrow chamfers are retained only when they are faithfully sampled and reproduced by the decoder, suggesting that an edge-aware weighting scheme is a useful direction for future work.

Our off-diagonal Weingarten loss treats curvature reduction purely locally; an immediate extension is to learn an adaptive weight  $\lambda_{\text{ODW}}$  that depends on local feature probability, so planar zones flatten faster while highly curved zones remain expressive. A second avenue is joint optimization with mesh extraction: coupling the Weingarten loss with differentiable DMC/dual-contouring could avoid the Marching-Cubes voxel bias.

*Conclusions.* We introduced the off-diagonal Weingarten loss that measures the curvature gap and needs only one Hessian–vector product, giving  $\approx 2\times$  faster convergence than full Gaussian-curvature regularization and using roughly half the GPU memory, while

matching or surpassing state-of-the-art accuracy on the ABC benchmark. Two proposed implementations (auto-diff and finite-diff) perform similarly; the finite-difference stencil is preferable when back-propagation memory is scarce. The technique is drop-in for any SIREN-based reconstruction pipeline and our code is available at <https://flatcad.github.io/>.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for valuable inputs and our institutions for supporting the project.

## REFERENCES

- Matan Atzmon and Yaron Lipman. 2019. SAL: Sign Agnostic Learning of Shapes From Raw Data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 2562–2571. <https://api.semanticscholar.org/CorpusID:208267630>
- Matan Atzmon and Yaron Lipman. 2020. SAL++: Sign Agnostic Learning with Derivatives. *ArXiv* abs/2006.05400 (2020). <https://api.semanticscholar.org/CorpusID:219558695>
- Yizhak Ben-Shabat, Chamin Pasidu Hewa Koneputugodage, and Stephen Gould. 2021. DiGS : Divergence guided shape implicit neural representation for unoriented point clouds. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 19301–19310. <https://api.semanticscholar.org/CorpusID:235490568>
- Alexandre Boulch and Renaud Marlet. 2022. POCO: Point Convolution for Surface Reconstruction. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 6292–6304. <https://api.semanticscholar.org/CorpusID:245769792>
- Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. 2001. Reconstruction and representation of 3D objects with radial basis functions. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001). <https://api.semanticscholar.org/CorpusID:11863186>
- Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, S. Lovegrove, and Richard A. Newcombe. 2020. Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction. In *European Conference on Computer Vision*. <https://api.semanticscholar.org/CorpusID:214623267>
- Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. 2020. Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 6968–6979. <https://api.semanticscholar.org/CorpusID:211817828>
- Qiuqie Dong, Rui Xu, Pengfei Wang, Shuangmin Chen, Shiqing Xin, Xiaohong Jia, Wenping Wang, and Changhe Tu. 2024. NeurCADRecon: Neural Representation for Reconstructing CAD Surfaces by Enforcing Zero Gaussian Curvature. *ACM Transactions on Graphics (TOG)* 43 (2024), 1 – 17. <https://api.semanticscholar.org/CorpusID:269293976>
- Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy Jyoti Mitra, and Michael Wimmer. 2020. Points2Surf Learning Implicit Surfaces from Point Clouds. In *European Conference on Computer Vision*. <https://api.semanticscholar.org/CorpusID:226298441>
- Amos Gropp, Lior Yaniv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit Geometric Regularization for Learning Shapes. In *International Conference on Machine Learning*. <https://api.semanticscholar.org/CorpusID:211259068>
- Hugues Hoppe, Tony DeRose, Tom Duchamp, John Alan McDonald, and Werner Stuetzle. 1992. Surface reconstruction from unorganized points. *Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (1992). <https://api.semanticscholar.org/CorpusID:1122241>



Fig. 10. Results of our method, each set shows the ground truth, the used point clouds, and the rendering of our reconstruction.

- Fei Hou, Chiyu Wang, Wencheng Wang, Hong Qin, Chen Qian, and Ying He. 2022. Iterative poisson surface reconstruction (iPSR) for unoriented points. *ACM Transactions on Graphics (TOG)* 41 (2022), 1 – 13. <https://api.semanticscholar.org/CorpusID:250956751>
- Jiahui Huang, Haoxiang Chen, and Shihui Hu. 2022. A Neural Galerkin Solver for Accurate Surface Reconstruction. *ACM Transactions on Graphics (TOG)* 41 (2022), 1 – 16. <https://api.semanticscholar.org/CorpusID:254097072>
- Zhiyang Huang, Nathan A. Carr, and Tao Ju. 2019. Variational implicit point set surfaces. *ACM Transactions on Graphics (TOG)* 38 (2019), 1 – 13. <https://api.semanticscholar.org/CorpusID:196834922>
- Chiayi Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas A. Funkhouser. 2020. Local Implicit Grid Representations for 3D Scenes. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 6000–6009. <https://api.semanticscholar.org/CorpusID:214606025>
- Kacper Kania, Maciej Zieba, and Tomasz Kajdanowicz. 2020. UCSG-Net - Unsupervised Discovering of Constructive Solid Geometry Tree. *ArXiv* abs/2006.09102 (2020). <https://api.semanticscholar.org/CorpusID:219708841>
- Misha Kazhdan, Ming Chuang, Szymon Rusinkiewicz, and Hugues Hoppe. 2020. Poisson Surface Reconstruction with Envelope Constraints. *Computer Graphics Forum* 39 (2020). <https://api.semanticscholar.org/CorpusID:220116886>
- Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*. <https://api.semanticscholar.org/CorpusID:14224>
- Michael M. Kazhdan and Hugues Hoppe. 2013. Screened poisson surface reconstruction. *ACM Trans. Graph.* 32 (2013), 29:1–29:13. <https://api.semanticscholar.org/CorpusID:1371704>
- Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. (2017). arXiv:1412.6980 [cs.LG]
- Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, L. Yi, and Leonidas J. Guibas. 2018. Supervised Fitting of Geometric Primitives to 3D Point Clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 2647–2655. <https://api.semanticscholar.org/CorpusID:53715802>
- Manyi Li, Falai Chen, Wenping Wang, and Changhe Tu. 2016. Sparse RBF surface representations. *Comput. Aided Geom. Des.* 48 (2016), 49–59. <https://api.semanticscholar.org/CorpusID:35948521>
- Siyou Lin, Dong Xiao, Zuqiang Shi, and Bin Wang. 2022. Surface Reconstruction from Point Clouds without Normals by Parametrizing the Gauss Formula. *ACM Transactions on Graphics* 42 (2022), 1 – 19. <https://api.semanticscholar.org/CorpusID:251257800>
- Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2018. Occupancy Networks: Learning 3D Reconstruction in Function Space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 4455–4465. <https://api.semanticscholar.org/CorpusID:54465161>
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)* 41 (2022), 1 – 15. <https://api.semanticscholar.org/CorpusID:246016186>
- Tiago Novello, Vinicius da Silva, Guilherme Schardong, Luiz Schirmer, Helio Lopes, and Luiz Velho. 2023. Neural Implicit Surface Evolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 14279–14289.
- Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and S. Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 165–174. <https://api.semanticscholar.org/CorpusID:58007025>
- Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. 2020. Convolutional Occupancy Networks. *ArXiv* abs/2003.04618 (2020). <https://api.semanticscholar.org/CorpusID:212646575>
- Lu Sang, Abhishek Saroha, Maolin Gao, and Daniel Cremers. 2025. Enhancing Surface Neural Implicit with Curvature-Guided Sampling and Uncertainty-Augmented Representations. In *Pattern Recognition: 46th DAGM German Conference, DAGM GCPR 2024, Munich, Germany, September 10–13, 2024. Proceedings, Part I* (Munich, Germany). Springer-Verlag, Berlin, Heidelberg, 312–328. [https://doi.org/10.1007/978-3-031-85181-0\\_20](https://doi.org/10.1007/978-3-031-85181-0_20)
- Pratheba Selvaraju. 2024. Developability Approximation for Neural Implicit Through Rank Minimization. In *2024 International Conference on 3D Vision (3DV)*. 780–789. <https://doi.org/10.1109/3DV62453.2024.00041>
- Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. 2017. CSGNet: Neural Shape Parser for Constructive Solid Geometry. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 5515–5523. <https://api.semanticscholar.org/CorpusID:4939249>
- Gopal Sharma, Difan Liu, Evangelos Kalogerakis, Subhransu Maji, Siddhartha Chaudhuri, and Radomir Mvech. 2020. ParSeNet: A Parametric Surface Fitting Network for 3D Point Clouds. *ArXiv* abs/2003.12181 (2020). <https://api.semanticscholar.org/CorpusID:214692999>
- Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. 2020. Implicit Neural Representations with Periodic Activation Functions. *ArXiv* abs/2006.09661 (2020). <https://api.semanticscholar.org/CorpusID:219720931>
- Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *ArXiv* abs/2006.10739 (2020). <https://api.semanticscholar.org/CorpusID:219720931>

- [org/CorpusID:219791950](https://api.semanticscholar.org/CorpusID:219791950)
- Jiapeng Tang, Weikai Chen, Jie Yang, Bo Wang, Songrun Liu, Bo Yang, and Lin Gao. 2021. OctFields: Hierarchical Implicit Functions for 3D Modeling. In *Neural Information Processing Systems*. <https://api.semanticscholar.org/CorpusID:240354073>
- Mikaela Angelina Uy, Yen-Yu Chang, Minhyuk Sung, Purvi Goel, J. Lambourne, Tolga Birdal, and Leonidas J. Guibas. 2021. Point2Cyl: Reverse Engineering 3D Objects from Point Clouds to Extrusion Cylinders. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 11840–11850. <https://api.semanticscholar.org/CorpusID:245329808>
- Peng-Shuai Wang, Yang Liu, and Xin Tong. 2022. Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Transactions on Graphics (TOG)* 41 (2022), 1 – 15. <https://api.semanticscholar.org/CorpusID:248525003>
- Zixiong Wang, Yunxiao Zhang, Rui Xu, Fan Zhang, Peng Wang, Shuangmin Chen, Shiqing Xin, Wenping Wang, and Changhe Tu. 2023. Neural-Singular-Hessian: Implicit Neural Representation of Unoriented Point Clouds by Enforcing Singular Hessian. *ACM Transactions on Graphics (TOG)* 42 (2023), 1 – 14. <https://api.semanticscholar.org/CorpusID:261529950>
- Jianglong Ye, Yuntao Chen, Naiyan Wang, and X. Wang. 2022. GIFS: Neural Implicit Function for General Shape Representation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 12819–12829. <https://api.semanticscholar.org/CorpusID:248177781>