Learning Geometry-Aware Embedding Fields for Intrinsic Riemannian Mappings

Bo Pang[1,2], Simone Foti[1], and Tolga Birdal[1] ⋆

[1] Imperial College London, United Kingdom
[2] Peking University, China

Abstract. Computing exponential and logarithmic maps on raw geometries is a fundamental yet challenging task in geometry processing, often hindered by the lack of explicit connectivity and the presence of noise. We present a novel learning-based framework that computes these maps efficiently by learning a continuous geometric embedding field directly from discrete point sets. Our key insight is to construct an embedding field that encapsulates the intrinsic local geometry governing exponential and logarithmic maps. Specifically, we encode the input into a high-dimensional feature volume using a sparse Octree-based CNN. For any arbitrary query point, we retrieve its corresponding embedding to modulate two specialized triplane-based neural networks, which then predict the maps in a single forward pass. By formulating the problem as learning a query-able latent field, our method bypasses the need for mesh connectivity while ensuring robustness against irregular sampling and noise. Our results demonstrate that the proposed framework achieves competitive accuracy and runtime over previous methods, while maintaining robustness on challenging geometric structures.

Keywords: Geometry Processing · Geometric Learning · Neural Fields

1 Introduction

Differential geometry, and in particular Riemannian geometry [5,12] provides the conceptual and computational foundation for reasoning about curved spaces. It furnishes the language through which distances, directions, and intrinsic structure on manifolds are rigorously defined. On smooth and well-behaved manifolds such as spheres, tori, or SO(3), these geometric objects admit well-understood analytical expressions. In contrast, the geometric data encountered in modern graphics or vision rarely conforms to such idealized settings. Instead, surfaces are typically observed as noisy, discretized samples, or irregular triangulations, yielding point clouds or meshes that only approximate an underlying manifold, rendering the classical apparatus of Riemannian geometry unavailable. The challenge is further amplified by the differentiability requirements of modern machine learning pipelines.

⋆ Corresponding author.

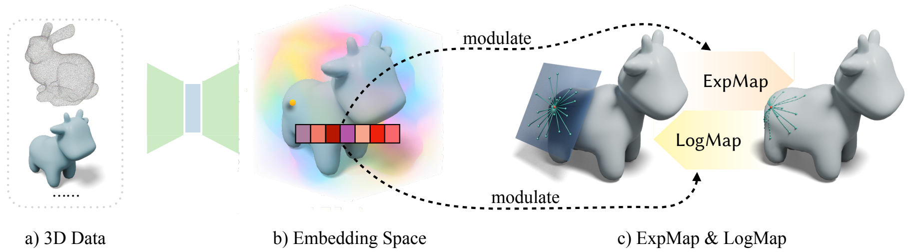


Fig. 1: Method Overview. We propose a field-based framework for learning intrinsic mappings. **(a)** Our method takes raw point clouds as input, which can be easily sampled from diverse sources such as SDFs or low-quality meshes, without requiring manifold connectivity. **(b)** A backbone network encodes this geometry into a continuous *Geometric Embedding Field*, where each point's embedding captures its local intrinsic properties. **(c)** For any query, the local embedding modulates lightweight decoders (ExpNet/LogNet) to instantly predict *Exponential and Logarithmic Maps*, enabling efficient and robust navigation on manifolds.

In this work, we adopt a data-driven perspective toward reconstructing two key operators of Riemannian geometry directly from geometric observations. Our focus lies on two fundamental constructs: the exponential map (Exp) and its inverse, the logarithmic map (Log). Exp transports vectors from the tangent space at a local point along geodesics onto the manifold, while Log performs the inverse operation, recovering tangent vectors that point toward a given surface location. Together, these maps establish a correspondence between the Euclidean tangent space and the intrinsically curved surface, forming the backbone of numerous geometric algorithms, enabling geodesic interpolation, texture parameterization, deformation transfer, and statistical analysis of shapes.

Despite their fundamental importance, computing Exp and Log maps on discrete geometry remains a challenging problem. Classical methods in discrete differential geometry have achieved remarkable success on well-formed polygonal meshes, where explicit connectivity allows geodesic information to be propagated through the mesh structure [9, 43]. Yet these approaches are inherently tied to mesh topology and degrade when applied to raw geometric data where connectivity is incomplete, unreliable, or entirely absent. Point clouds, triangle soups, and implicit surface representations provide only partial information about the underlying manifold, making the direct application of traditional algorithms difficult. Even reconstructing a clean manifold mesh from such data can be a challenging problem in its own right.

For implicit surface representations, navigating the manifold typically requires iterative projection [19, 62] or geodesic tracing procedures [15, 27]. While theoretically principled, these methods can be computationally expensive and susceptible to numerical instability, particularly in the presence of high curvature or sharp geometric features. Moreover, purely geometric heuristics [51] often lack the ability to reason about higher-level shape structure. As a result, they may incorrectly traverse opposing sheets of thin surfaces or fail to bridge gaps in sparsely sampled or noisy regions.

These challenges motivate the central question of this work: *can we learn the fundamental operators of Riemannian geometry directly from data, enabling robust geometric reasoning even when classical assumptions about smoothness and connectivity no longer hold*? In response, we propose a learning-based framework for computing Exp and Log maps directly from raw geometric observations. Illustrated in Fig. 1, our central idea is to learn a continuous *geometric embedding field* (GEF) that captures the intrinsic structure of the underlying manifold and enables efficient evaluation of these mappings at arbitrary locations.

We are motivated by two key properties of Exp and Log: *locality* and *regularity*. At a given point, the behaviour of Exp and Log is primarily determined by the local geometry. At the same time, the maps vary smoothly across most regions of the manifold: neighboring points typically induce similar local maps (except the cut locus). This observation suggests that the operators themselves form a structured, spatially coherent field over the surface. Building on this insight, our continuous embedding field [24, 30, 59] encodes the intrinsic geometry of the entire manifold. Rather than solving a geometric problem independently for each query, our method performs a one-time geometric analysis of the shape and stores the resulting structure in this field representation.

Once constructed, at query time, given a source point on the surface, we first retrieve its corresponding embedding from the geometric field via spatial interpolation. This embedding acts as a local geometric descriptor that conditions the evaluation of two neural modules: ExpNet and LogNet. Rather than using fixed network parameters, we employ a dynamic conditioning mechanism inspired by HyperNetworks [18] and feature-wise modulation techniques such as FiLM [34]. The retrieved embedding dynamically generates a compact triplane field representation [17, 45] tailored to the current query location.

To evaluate our Exp or Log maps, we query this generated triplane field using the relative coordinates between the source point and the query point. The resulting features are then passed through a lightweight decoder to produce the final result. This design yields a highly efficient evaluation pipeline. The core operations only consist of simple interpolations and small matrix multiplications, enabling large batches of queries to be processed in parallel on modern GPUs.

Conceptually, our approach can be interpreted as learning a *neural atlas* of local geometric behavior across the manifold. The GEF captures the global distribution of local geometries, while the dynamically generated triplane fields instantiate the specific mapping rules associated with each source point. Together, they provide a flexible and scalable mechanism for approximating the exponential and logarithmic maps on complex, irregular geometric data. To summarize, our main contributions are:

- We propose a novel framework that learns a continuous geometric embedding field from discrete point sets to compute Exp and Log maps efficiently.

– We introduce a geometry-aware conditioning mechanism and a series of geometric regularizations, enabling a single network to adapt to diverse local geometries without explicit connectivity.
– We demonstrate that our method achieves competitive accuracy and reasonably good speed compared to previous methods, while exhibiting strong robustness on challenging geometric structures.

We make our implementation publicly available under:
https://circle-group.github.io/research/LearningRiemannianGeometry.

2 Related Work

We review relevant literature in differential geometry for 3D shapes, neural shape representations, and the emerging field of learning geometric operators.

2.1 Differential Geometry of 3D Shapes

Geodesic Distance Computation. Finding the shortest path on discrete meshes is a classical problem in geometry processing. Early exact methods, such as the MMP algorithm [29] and its variants [49,61], provide exact geodesic paths by propagating wavefronts across mesh edges. However, these exact methods typically suffer from high computational complexity and are difficult to parallelize. To improve efficiency, heat-flow based methods were introduced. The *Heat Method* [8, 9] solves a linear Poisson system to approximate geodesic distances based on short-time heat diffusion. Subsequent works have extended this to broader domains, including point clouds [41] and non-manifold geometry [42]. GeGnn [31] and the work by Huberman et al. [21] might be the most similar works to our method. They also precompute a field, which is then used to predict certain geometry-related quantities.

Exponential and Logarithmic Maps. The exponential map projects tangent vectors onto the surface along geodesics, while the logarithmic map performs the inverse [39, 47]. The Vector Heat Method [43] and the recent Affine Heat Method [48] extend the heat diffusion framework to parallel transport vectors and compute logarithmic maps efficiently. Lichtenstein et al. [26] solves the Eikonal equation, which potentially enables the geodesic-related operations. Such methods usually require a triangular mesh with good triangle quality to ensure the accuracy of the computed maps. Madan et al. [27] enables logarithm map on implicit surfaces by tracing geodesic paths. Genest et al. [15] propose to compute logarithm map via approximated geodesic distance. These methods tend to rely on explicit tracing of geodesic paths, which is not always possible or efficient. Also, some methods [26, 44, 48] have to solve an equation for the entire shape, and cannot just give evaluations in local regions. Concurrent to our work, [51] proposed differentiable *straightest geodesic* algorithms, to enable learning over 3D meshes. Yet, these algorithms are not data driven and cannot be used on noisy point scans.

2.2 Neural Shape Learning and Geometric Operators

We now review deep learning on unstructured data representations, which has revolutionized 3D geometry processing.

Neural Shape Representations. Point-based methods [25, 35, 36, 56] process raw point clouds directly but often lack the connectivity information required for intrinsic geometric analysis. Implicit representations, such as Signed Distance Functions (SDFs) and Occupancy Fields [6, 28, 32, 33], excel at representing continuous geometry and handling topology changes. Recent works have further improved these representations with periodic activation functions [46], surface smoothness regularization [1,2,16], or hybrid structures like Octree-based CNNs [37, 52, 53] and Triplanes [17, 45].

Learning Geometric Operators. Apart from shape understanding and reconstruction, recent research has begun to explore learning intrinsic geometric operators directly from data. DiffusionNet [40] learns discretization-agnostic diffusion features for discriminative tasks, via a simulated heat diffusion process. For geodesic computation, GeodesicEmbedding [60], GeGnn [31], and NeuroGF [63] learn high-dimensional embeddings or neural fields to approximate geodesic distances. Recent advances extend this to solving PDEs on neural representations [58] or generating UV-free textures via geodesic heat diffusion [14]. However, currently, few works address the explicit learning of the vector-valued Exponential and Logarithmic maps.

3 Geometric Embedding and Hyper-Modulation

Our goal is to learn neural approximations for the Exp and Log maps on Riemannian manifolds in 3D. Our pipeline consists of two components: (1) a *backbone network* that converts input into a continuous feature field; and (2) a pair of lightweight, hyper-modulated decoder networks, ExpNetand LogNet, that perform the mappings, dynamically conditioned on the local geometric context.

Problem Formulation. Let $\mathcal{M} \subset \mathbb{R}^3$ be a 2-manifold surface embedded in 3D Euclidean space. For any source point $p \in \mathcal{M}$, let $T_p\mathcal{M}$ denote the tangent plane at p. The *Exponential Map*, $\mathrm{Exp}_p : T_p\mathcal{M} \to \mathcal{M}$, maps a tangent vector $v \in T_p\mathcal{M}$ to a point $q \in \mathcal{M}$ such that q is reached by traveling along the geodesic path starting at p with initial direction of v. Conversely, the *Logarithmic Map*, $\mathrm{Log}_p : \mathcal{M} \to T_p\mathcal{M}$, is the inverse operation, mapping a point $q \in \mathcal{M}$ back to the tangent vector $v \in T_p\mathcal{M}$. While Exp_p is locally diffeomorphic around the origin, it is not globally injective due to the cut locus. Log_p is well-defined only within the injectivity radius. Given a discrete input geometry (e.g., a point cloud $\mathcal{P}$) and a source point p, our aim is to learn two continuous and differentiable functions, **ExpNet** and **LogNet**, that predict the Exp and Log maps, respectively.

3.1 Geometric Embedding Field

A key challenge in 3D learning is the diversity of data representations. Our method only requires a set of coarse boundary points, which can be easily obtained from SDFs, occupancy fields, point clouds, triangle soups, or Functa [13].

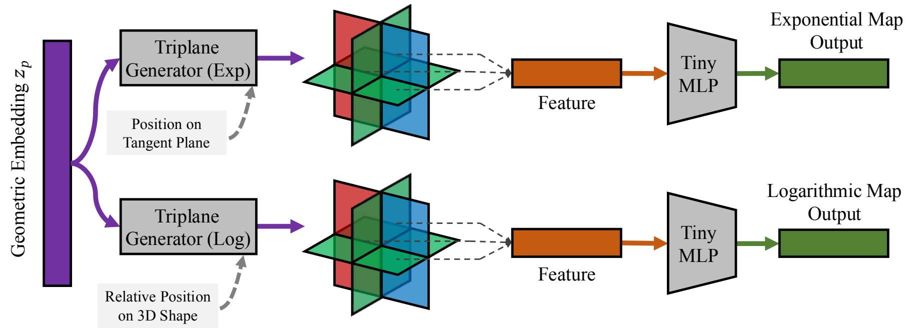


Fig. 2: Hyper-Modulation Mechanism. To decode the mapping at a specific point p, we first retrieve its geometric embedding z_p. The embedding, after a matrix multiplication, synthesizes a local triplane feature field, which is then queried by relative coordinates. The retrieved features are decoded by a tiny MLP to output the exp or log map result. This design ensures that the mapping function is dynamically adapted to the local geometry while maintaining translation invariance and high inference speed.

Backbone Network and Feature Field. We employ an octree-based convolutional neural network (OCNN) with a 3D U-Net architecture as our backbone [38,53]. The network consists of an encoder that progressively downsamples the sparse octree features, followed by a decoder with skip connections that restores spatial resolution. This design allows the backbone to effectively aggregate both global shape context and local geometric details. The final output is a continuous volumetric feature field, from which we can retrieve the local geometric embedding z_p for any arbitrary source point p via trilinear interpolation. Since the OCNN operates purely on relative spatial structures and is fully convolutional, the resulting embedding field is inherently translation-invariant.

3.2 Hyper-Modulated Triplane Representations

As discussed before, while the global geometry of $\mathcal{M}$ varies, the properties of exponential and logarithmic maps are governed by local geometric shape. Thus, a local geometric embedding z_p is actually sufficient to capture the behavior of the map at p. To this end, we propose a *hyper-modulation* mechanism that dynamically adapts the networks to the local geometry at p.

For a specific query consisting of a source point p and an input, either a 3D tangent vector v or a 3D manifold point q, our pipeline proceeds as follows.

1. **Dynamic Triplane Generation:** Following FiLM [34], the retrieved embedding z_p is used to generate the parameters of triplane representations for ExpNet and LogNet, respectively. Specifically, a linear layer maps z_p to a feature volume that is reshaped into three orthogonal planes, XY, XZ, YZ. Each plane captures the local geometric characteristics at p.
2. **Field-Based Sampling:** For a query coordinate x, we sample features from the generated triplane via bilinear interpolation on the three planes. By using relative coordinates and a fully convolutional backbone, our entire mapping process remains translation-invariant.

3. **Lightweight Decoding:** The sampled triplane features are concatenated together and passed through a tiny MLP decoder (ExpNet or LogNet).

Crucially, our ExpNet and LogNet operate directly in the ambient 3D coordinate system using relative coordinates, rather than relying on local 2D tangent charts. This design choice is deliberate: defining a local 2D basis on a tangent plane is inherently ambiguous (up to a rotation), which can confuse the network and hinder convergence. By processing 3D coordinates directly, we bypass this ambiguity, ensuring that our network learns a consistent and robust mapping function across the entire manifold. This process is very lightweight. In our implementation, it involves only approximately 5 matrix multiplications and 3 linear interpolations per query, and is highly parallelizable.

3.3 Training

We now describe our data preparation and filtering, followed by loss functions.

Synthesizing Training Data. Training a robust neural embedding field requires high-quality geometric data that is manifold, watertight, and exhibits rich local geometric variations. Standard datasets like ShapeNet [7], Thing10K [65] or Objaverse [10, 11] often fail to meet these criteria. For example, ShapeNet is dominated by planar structures, e.g., tables, cabinets, introducing a strong inductive bias that limits the network's generalization to curved surfaces.

To address this, we construct a custom synthetic dataset using a generative pipeline. We first select a subset of ShapeNet models and render them into 2D images. These images are fed into a large multimodal model, Google Gemini 3 Flash [50], to generate descriptive text prompts. Using these prompts, we employ Flux Dev [23] to synthesize diverse 2D images, which are then lifted to 3D geometries using Hunyuan3D 2.0 [64]. Then, we follow the processing method from GeGnn [31] to ensure mesh quality. Since our method learns local geometric rules rather than global shape, we use a compact training set of approximately 1,500 such meshes.

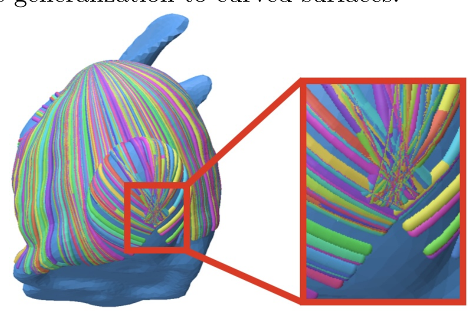


Fig. 3: Visualization of the cut locus on a Stanford Bunny, where the non-injectivity of the exponential map arises from intersecting geodesics. Our data filtering strategy avoids such cases affecting the training of LogNet.

After that, we use the open-sourced library POTPOURRI3D to generate the ground truth. Specifically, for each mesh, we randomly sample M source points, and for each source point p, we generate N vectors on the tangent plane at p. We then use the library to compute the exponential map at p for each vector. For the logarithm map, we simply use the inverse of the exponential map.

Filtering Training Samples. A theoretical challenge in learning differential geometric operators is that they are not globally bijective. The exponential map is locally surjective but not injective, that is, multiple tangent vectors can map to the same point due to periodicity or cut locus, and the logarithm map is well-defined only if the cut locus is not reached, as illustrated in Fig. 3.

Training a neural network on raw, unfiltered data often involves processing a large number of redundant or ambiguous samples. This not only increases the computational burden but also slows down convergence. To address this and improve efficiency, we filter our training data. Specifically, if a target point on the manifold corresponds to multiple distinct vectors in the tangent plane (due to geodesics wrapping around), we treat it as being on or beyond the cut locus, and therefore not valid for training . We refer to our supplementary materials for the specifics of the data filtering process. By focusing on valid, bijective regions, the network can learn the operators more efficiently, avoiding the overhead of processing ambiguous or non-informative regions.

Loss Functions. We train our network via a multi-objective loss function, composed of reconstruction terms, geometric regularization, and cycle consistency constraints, in order to learn accurate exponential and logarithmic maps while preserving the intrinsic geometric properties of the two maps.

$$\mathcal{L}_{\text{total}} = \lambda_{\text{rec}} \underbrace{(\mathcal{L}_{\text{Exp}} + \mathcal{L}_{\text{Log}})}_{\text{Reconstruction loss } \mathcal{L}_{\text{rec}}} + \lambda_{\text{geo}} \underbrace{(\mathcal{L}_{\text{iso}} + \mathcal{L}_{\text{conf}})}_{\text{Geometric loss } \mathcal{L}_{\text{geo}}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cycle}} \tag{1}$$

where $\lambda_{rec} = 1.0$, $\lambda_{geo} = 0.01$, and $\lambda_{cyc} = 0.01$.

1. **Reconstruction Loss** $\mathcal{L}_{\text{rec}}$: The primary supervision comes from the L_1 distance between the predicted maps and the ground truth generated on high-quality ground truth meshes. For the Exp, we minimize the error in the 3D embedding space, while for the Log, we minimize the error in the tangent space:
$$\mathcal{L}_{\text{Exp}} = \|\hat{q} - q_{\text{gt}}\|_1 \qquad \mathcal{L}_{\text{Log}} = \|\hat{v} - v_{\text{gt}}\|_1. \tag{2}$$
We observe that incorporating ground truth supervision is crucial for stabilizing the early stages of training. Relying solely on self-supervised constraints often leads to trivial solutions or mode collapse.
2. **Geometric Losses** $\mathcal{L}_{\text{geo}}$: Inspired by the optimizing objectives proposed by [27, 47], to enforce that the learned maps have good intrinsically geometric properties, we impose regularization on the Jacobian of the neural fields. Ideally, both maps should be locally isometric (distance-preserving) and conformal (angle-preserving). Note that our pipeline is fully differentiable, and thus the Jacobian matrix J of our mapping can be obtained via auto-differentiation. Since both mappings operate in 3D space ($\mathbb{R}^3 \to \mathbb{R}^3$), the Jacobian J is a 3×3 matrix. We compute $M = J^T J$ and define two regularization terms using a robust L_1 loss: (i) **Isometric Loss** $\mathcal{L}_{\text{iso}}$, where we encourage the diagonal elements of M, squared norms of basis vectors, to be close to 1, and (ii) **Conformal Loss** $\mathcal{L}_{\text{conf}}$, where we penalize the off-diagonal elements of M to ensure that orthogonal vectors remain orthogonal after mapping:
$$\mathcal{L}_{\text{iso}} = \sum_i |M_{ii} - 1| \qquad \mathcal{L}_{\text{conf}} = \sum_{i \neq j} |M_{ij}|. \tag{3}$$

These losses are applied to both ExpNet and LogNet predictions.

3. **Cycle Consistency Loss** $\mathcal{L}_{\text{cycle}}$: Finally, to enforce the duality between the forward and backward maps, we introduce a self-supervised cycle consistency constraint. Let $g(\cdot; z_p)$ denote the ExpNet and $h(\cdot; z_p)$ denote the LogNet. Mapping a vector v to the manifold via ExpMap and then back to the tangent space via LogMap should recover the original vector:

$$\mathcal{L}_{\text{cycle}} = \|h(g(v; z_p); z_p) - v\|_1 \tag{4}$$

This term acts as a regularizer that enables the two networks to mutually correct each other.

Overall, the reconstruction loss dominates the optimization, while the geometric and cycle consistency losses serve as auxiliary regularizers to ensure smoothness and topological correctness.

3.4 Stitching Log Maps via Weighted Rigid Alignment

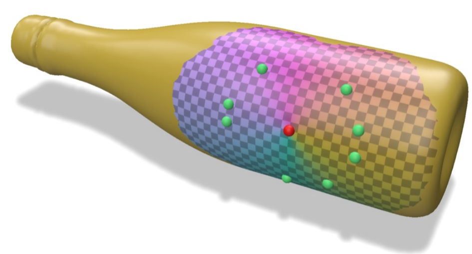


Fig. 4: Example of large-scale log map stitching on a bottle. Local charts from multiple source points (green dots) are aligned and blended to form a continuous parameterization over a significant portion of the manifold.

To parameterize a larger region of the surface $\mathcal{M}$, similar to [15], we propose a method that stitches together multiple overlapping local log maps. Our approach relies on a greedy propagation strategy combined with robust weighted rigid alignment. Specifically, we first perform a breadth-first search to cover the surface with overlapping patches. For a new patch k centered at x_k that overlaps with the already visited region $\mathcal{R}_{\text{visited}}$, we compute the optimal 2D rigid transformation that aligns the local coordinates of patch k with the current global coordinates in the overlapping area. Once aligned, the global coordinates of vertices in the new patch are updated using a weighted blending scheme.This blending process effectively suppresses discontinuities and noise in the transition regions. Please refer to our supplementary material for detailed algorithms and implementation.

4 Experiments

We conduct a series of experiments across different 3D shapes, including quantitative and qualitative evaluations to ensure a comprehensive assessment. Detailed explanations of our experiments can be found in the supplementary.

Experimental Setup. For controlled, synthetic evaluations, we use the ShapeNet [7] dataset. We randomly select 1,000 samples across 18 categories. ShapeNet meshes are usually non-manifold and of bad topology. To allow for calculating the ground truth, we first convert them into watertight manifolds following [55], and subsequently, we apply isotropic explicit remeshing [3, 20] to regularize the mesh connectivity. On each shape, we randomly sample 10 points as the source points and evaluate the exp and log maps on them. We trained the model for 400 epochs using the AdamW optimizer [22] with a learning rate of 10^{-3} and weight decay of 10^{-4}. Both of them are default settings of PyTorch. The training process takes approximately 8 hours on four NVIDIA RTX 4090 GPUs. All input geometries are normalized to the range $[-1, 1]$.

To test the generalization of our method, we also conduct evaluations on non-rigid shapes of DFAUST [4] and highly noisy scans of real-world objects of Kinect-v2 [54], using the same checkpoint from the above-mentioned training, without further fine-tuning.

Metrics. To evaluate our performance, following the idea of [47], we employ two primary metrics for Log map, and one for Exp map. Please note that most metrics are impossible to reach 0 on any non-developable surface.

- **Mean Angular Error:** This metric evaluates the conformality of the Log mapping from the 3D surface to the tangent plane. We compute the angular distortion on valid triangles, for each valid triangle T_k, we calculate its three internal angles in both 3D space, $\theta^{3\text{D}}$, and in tangent space after the Log map, $\theta^{\tan}$. The distortion for an angle is the absolute difference $\varDelta\theta = |\theta^{3\text{D}} - \theta^{\tan}|$. The final metric is the average angular error across all angles of all N_{valid} valid triangles:
$$\mathcal{E}_\theta = \frac{1}{3N_{valid}} \sum_{k=1}^{N_{\text{valid}}} (\varDelta\theta_{A,k} + \varDelta\theta_{B,k} + \varDelta\theta_{C,k}) \tag{5}$$
A lower value indicates better angular preservation.
- **Mean Distance Distortion** ($\mathcal{E}_{\text{MDD}}$): This metric measures the isometric quality, i.e., the discrepancy between the distances before and after the log map. Similar to the angular distortion, we consider only valid edges that satisfy the filtering criteria. For each valid edge e, we compute the relative distortion as $\delta_e = |\frac{L_{3\text{D}}}{L_{\tan}} - 1|$, where $L_{3\text{D}}$ is the edge length on the 3D mesh and $L_{\tan}$ is the corresponding length in the tangent plane after the log map. The final metric is the average of these relative distortions over all valid edges.
- **Exponential Map Error**: The exponential map maps a tangent vector v to a point q on the 3D surface. We generate a set of query vectors $\mathcal{V} = \{v_1, \dots, v_M\}$ on the tangent plane. Let $\mathcal{Q}_{GT} = \{q_1, \dots, q_M\}$ be the ground truth endpoints obtained via geodesic tracing, and $\mathcal{Q}_{Pred} = \{p_1, \dots, p_M\}$ be the network predictions. We measure the average Euclidean error in 3D space:
$$\mathcal{E}_{\text{Exp}} = \frac{1}{M} \sum_{j=1}^{M} \|q_j - p_j\|_2 \tag{6}$$
This metric directly evaluates how close our prediction is to the ground truth. A lower value indicates better accuracy.

4.1 Quantitative Evaluation

In this section, we report the quantitative results of our method on the testing set. We compare our approach with a few baselines. For the log map, we consider

Table 1: Quantitative evaluation of Logarithmic and Exponential maps on the test sets. For Log, we report the Mean Distance Distortion ($\mathcal{E}_{\text{MDD}}$) and the Mean Angular Error ($\mathcal{E}_{\theta}$) in degrees. We report the Average Euclidean Error, and we also report the average runtime per query *Time* in seconds, for both methods. Across both operators, our method achieves the best balance between accuracy and efficiency, outperforming baselines, especially on challenging inputs.

Category	N	**#Verts**	**Logarithmic Map**				**Exponential Map**		
				$\mathcal{E}_{\text{MDD}} \downarrow$	$\mathcal{E}_{\theta} \downarrow$	Time↓		$\mathcal{E}_{\text{Exp}} \downarrow$	Time↓
ShapeNet	1000	16.6k	Heat	0.250	16.7	0.178	Tracing	0.036	0.396
			Proj.	0.115	14.6	**0.001**	Proj.	0.049	**0.003**
			Ours	**0.110**	**11.1**	0.012	Ours	**0.026**	0.019
RawShapeNet	200	10.0k	Heat	0.314	21.3	0.101	Tracing	0.050	0.374
			Proj.	**0.166**	17.1	**0.001**	Proj.	0.048	**0.003**
			Ours	0.169	**15.4**	0.013	Ours	**0.036**	0.020
Non-uniform	200	15.0k	Heat	0.339	22.4	0.051	Tracing	0.043	0.415
			Proj.	0.155	16.3	**0.0005**	Proj.	0.047	**0.003**
			Ours	**0.152**	**14.0**	0.011	Ours	**0.030**	0.019
Noisy	200	16.6k	Heat	0.326	21.1	0.101	Tracing	0.060	0.374
			Proj.	0.203	19.4	**0.001**	Proj.	0.055	**0.003**
			Ours	**0.202**	**17.8**	0.013	Ours	**0.038**	0.020

two baselines. First, Heat Method [43] which solves the heat diffusion equation to recover geodesic information is used as the primary baseline. And we set another simple baseline that assumes local flatness by directly projecting the difference vector onto the tangent plane. For the Exponential Map, we compare against a heuristic walking method that iteratively projects steps onto the manifold to simulate geodesic tracing, and a simple linear projection baseline that naively extends the tangent vector in 3D Euclidean space and snaps to the surface via a projection. Detailed implementation descriptions of these baselines are provided in the supplementary material. The results are shown in Tab. 1.

Another thing to notice is our speed. Similar to GeGnn, our backbone network only requires a single forward pass to compute the embeddings, which on average takes less than 0.5 second for an input point cloud with 15k vertices with a batch size of 8 on a NVIDIA RTX 3090 GPU. Then, the *ExpNet* and *LogNet* predict the target coordinate in $O(1)$ time after the initial embedding computation. As discussed in Sec. 3.2, an inference only includes less than 10 matrix multiplications or linear interpolations. The results reported above verify the efficiency of our method. Note that for fairness, all time measurements are on an Apple M4 Max CPU, without using any parallelism of GPU.

Raw ShapeNet Geometry. To evaluate the capability of our method in handling low-quality and unstructured data, we construct a test set based on raw, unprocessed ShapeNet meshes [7]. These models usually have poor geometric quality, often riddled with self-intersections, non-manifold edges, and discon-

Table 2: Results on DFAUST and Kinect-v2. Lower is better for all metrics.

Log map

Method	DFAUST		Kinect-v2	
	$\mathcal{E}_{\mathrm{MDD}} \downarrow$	$\mathcal{E}_{\theta} \downarrow$	$\mathcal{E}_{\mathrm{MDD}} \downarrow$	$\mathcal{E}_{\theta} \downarrow$
Heat	0.23	25.47	0.12	8.45
Proj.	0.17	19.04	0.11	9.29
Ours	0.17	16.51	0.11	8.37

Exp map ($\mathcal{E}_{\mathrm{Exp}} \downarrow$)

Method	DFAUST	Kinect-v2
Tracing	0.033	0.063
Proj.	0.071	0.064
Ours	0.043	0.060

nected components, essentially resembling a "triangle soup". Such defects make mesh-based algorithms hardly usable. We generate input point clouds by sampling points directly from these raw surfaces using Poisson Disk Sampling [57]. We then feed these point clouds into our network and baseline methods to compute the exponential and logarithmic maps. This experiment directly tests the "in-the-wild" applicability of our approach. See Tab. 1 for results.

Robustness Analysis. We challenge our method with two types of data imperfections common in real-world scenarios: non-uniform sampling and noise.

- **Non-uniform Sampling:** We synthesize point clouds with spatially varying densities, where some regions are densely populated while others are sparse. This tests whether our learned embedding field can maintain consistent geometric understanding despite irregular discretization.
- **Noisy Inputs:** We introduce random uniform noise to the input point coordinates. Specifically, for each point, we add a displacement vector with a magnitude uniformly sampled from $[0, 0.015]$. This tests the resilience of our method against sensor noise and reconstruction artifacts.

The results of robustness analysis are at the bottom half of Tab. 1.

Noisy Scans and Generalization. We additionally provide evaluations on non-rigid shapes of DFAUST [4] and highly noisy scans of real-world objects of Kinect-v2 [54], as in Tab. 2. These experiments use the same checkpoint as in previous sections, without any retraining or fine-tuning. The results suggest that our data-driven method is stable even for unseen shapes, open boundaries, missing holes, and in-the-wild scans.

Ablation Study. To evaluate the key components of our method, we conduct an ablation study to evaluate their individual contributions. In Tab. 3, we demonstrate the experiment results on 100 ShapeNet models of three variants of our model. The "No Filtering" variant does not apply data filtering described in Sec. 3.3, while the "Simple Loss" variant uses only the supervised loss function described in Eq. (2). The results show that such variant suffers from a noticeable performance drop, highlighting the importance of our geometric priors for learning accurate mappings. While the "No Filtering" variant achieves accuracy comparable to our proposed method, it incurs a much higher computational burden. By removing redundant and ambiguous training samples (approx. 29.6%), our filtering strategy significantly reduces the training cost without compromising geometric accuracy. This demonstrates that our data filtering strategy

Table 3: Ablation study on different variants of our method. Note that the "No Filtering" variant consumes significantly more computational resources due to redundant training samples.

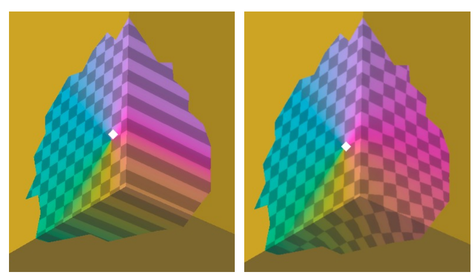
Variant	$\mathcal{E}_{\mathrm{MDD}}$ (Log)	$\mathcal{E}_{\theta}$ (Log)	$\mathcal{E}_{\mathrm{Exp}}$ (Exp)	Rel. Cost ↓
No Filtering	**0.112**	11.3	0.027	1.42×
Simple Loss	0.162	11.5	**0.025**	**1.00**×
Full	**0.112**	**11.2**	0.028	**1.00**×

effectively streamlines the learning process. Please refer to our supplementary material for more results.

4.2 Qualitative Evaluation

Synthetic Benchmarks. We now present qualitative results on a variety of shapes, as shown in Fig. 5 and Fig. 6. An interesting observation is that our method is able to handle thin structures much more accurately than previous methods. Thin structures pose a challenge for traditional methods due to "leakage" across boundaries. Most previous methods cannot distinguish between the upper and lower surfaces of a thin structure, leading to relatively inaccurate results. In contrast, our method leverages the geometric priors encoded by the backbone network, and accurately recovers the underlying manifold topology. Our network correctly learns that the path must wrap around the edge of the thin plate. As visualized in the first row of Fig. 5, heat methods fail to distinguish between the upper and lower surfaces.

Real-World Noisy Scans To validate our model's robustness on noisy or incomplete geometries, we visualize the results of different methods on Kinect-v2 dataset [54], as shown in Fig. 7. Kinect-v2 contains real-world scanned point clouds that are often noisy, incomplete, and geometrically irregular. Despite these challenges, our method produces more coherent and visually pleasing checkerboard patterns, while other methods exhibit stronger distortions and discontinuities, especially near boundaries and holes.



(a) Simple Baseline **(b)** Ours

Fig. 8: The simple projection baseline causes severe distortion if the normal of the underlying surface changes dramatically, while our method remains coherent.

Where does the Baseline Fail? The simple *projection* baselines for both Exp and Log maps rely on a linear projection. While fast, their accuracies degrade severely when the normal vector changes drastically, as shown in Fig. 8. This behavior is expected, since projection only performs a local linear mapping and does not model the underlying surface

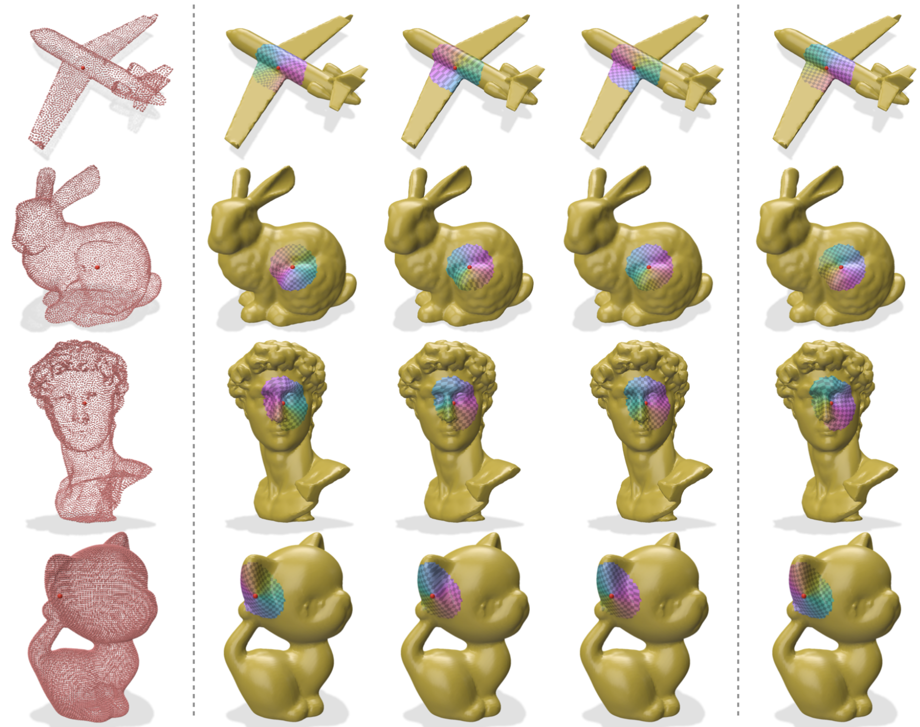


Fig. 5: Qualitative comparison of logarithmic map on point clouds. From left to right: input point cloud, heat method result, simple projection result, our method result, and reference ground truth. Our method faithfully recovers the intrinsic UV coordinates on complex shapes where baselines struggle. Meshes are visualized only for clarity; all inputs are raw point clouds.

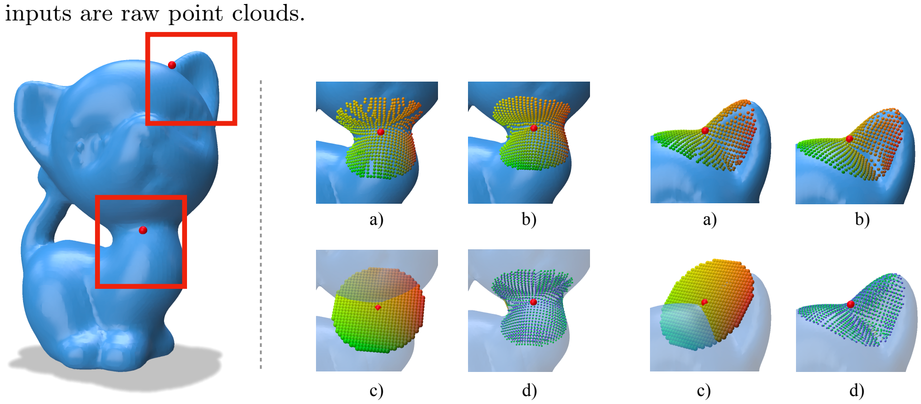


Fig. 6: Exponential Map visualization. We show results for two source points (red dots) located on the ear and neck. For each example, we visualize: **(a)** the heuristic method result, **(b)** our result, **(c)** the tangent plane, and **(d)** the error visualization (blue: prediction, green: ground truth).

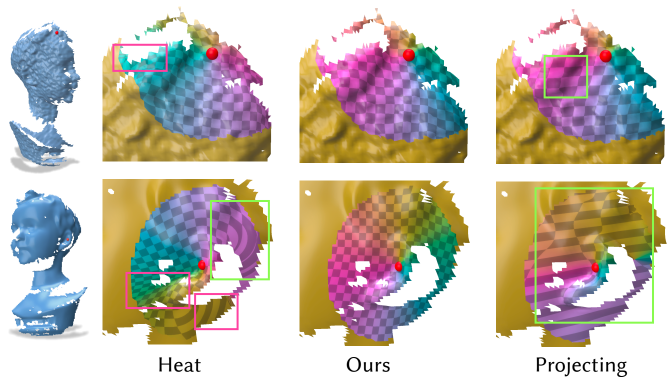


Fig. 7: Qualitative comparison of logarithmic maps on Kinect-v2 scans. Our method yields more coherent patterns with fewer distortions (highlighted boxes). Meshes are visualized only for clarity; all inputs are raw point clouds.

geometry beyond the tangent space. As a result, regions with sharp changes in surface orientation can introduce strong stretching and distortion.

5 Conclusion

In this paper, we presented a novel learning-based framework and regularization loss functions for computing exponential and logarithmic maps on 3D surfaces. By encoding geometry into a continuous embedding field and employing a geometry-aware triplane modulation mechanism, our method effectively decouples geometric inference from explicit mesh connectivity. This allows for robust and efficient computation of intrinsic mappings on raw point clouds and unstructured data. Our method demonstrates good accuracy, speed, and robustness in various experiments.

Limitations and Future Work. Our model currently relies on training data generated by classical numerical methods, meaning its accuracy is inherently bounded by the supervision signal. Moreover, Octree and triplane resolution may smooth out fine-scale details. Finally, while our method efficiently computes local mappings, extending them to large-scale parameterizations currently requires a separate stitching algorithm in Sec. 3.4, which may not be the optimal end-to-end solution. Future work could explore geometry-informed unsupervised learning to bypass data dependencies and integrate a global consistency directly into the learning framework.

References

1. Atzmon, M., Lipman, Y.: SAL: Sign agnostic learning of shapes from raw data. In: CVPR (2020)

2. Atzmon, M., Lipman, Y.: SALD: Sign agnostic learning with derivatives. In: ICLR (2021)
3. Bhat, P., Ingram, S., Turk, G.: Geometric texture synthesis by example. In: Symp. Geom. Proc. (2004)
4. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: CVPR (2017)
5. Boumal, N.: An introduction to optimization on smooth manifolds. Cambridge University Press (2023)
6. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In: ECCV (2020)
7. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
8. Crane, K., de Goes, F., Desbrun, M., Schröder, P.: Digital geometry processing with discrete exterior calculus. In: ACM SIGGRAPH 2013 courses (2013)
9. Crane, K., Weischedel, C., Wardetzky, M.: Geodesics in heat: A new approach to computing distance based on heat flow. ACM Trans. Graph. **32**(5) (2013)
10. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36**, 35799–35813 (2023)
11. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13142–13153 (2023)
12. Do Carmo, M.P., Flaherty Francis, J.: Riemannian geometry, vol. 393. Springer (1992)
13. Dupont, E., Kim, H., Eslami, S., Rezende, D., Rosenbaum, D.: From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204 (2022)
14. Foti, S., Zafeiriou, S., Birdal, T.: Uv-free texture generation with denoising and geodesic heat diffusion. Advances in Neural Information Processing Systems (2024)
15. Genest, B., Gueth, P., Levallois, J., Wang, S.: Implicit uvs: Real-time semi-global parameterization of implicit surfaces. In: Computer Graphics Forum. Wiley Online Library (2025)
16. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: ICML (2020)
17. Gupta, A., Xiong, W., Nie, Y., Jones, I., Oğuz, B.: 3DGen: Triplane latent diffusion for textured mesh generation. arXiv preprint arXiv:2303.05371 (2023)
18. Ha, D., Dai, A., Le, Q.V.: HyperNetworks. In: ICLR (2017)
19. He, Y., Tiwari, G., Birdal, T., Lenssen, J.E., Pons-Moll, G.: Nrdf: Neural riemannian distance fields for learning articulated pose priors. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
20. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Mesh optimization. In: SIGGRAPH (1993)
21. Huberman, S., Bracha, A., Kimmel, R.: Deep accurate solver for the geodesic problem. In: International Conference on Scale Space and Variational Methods in Computer Vision. Springer (2023)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2014)

23. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025), `https://arxiv.org/abs/2506.15742`
24. Li, H., Zhu, H., Zhong, S., Wang, N., Lin, C., Guo, X., Xin, S., Wang, W., Hua, J., Zhong, Z.: Nasm: Neural anisotropic surface meshing. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–12 (2024)
25. Li, J., Chen, B.M., Lee, G.H.: SO-Net: Self-organizing network for point cloud analysis. In: CVPR (2018)
26. Lichtenstein, M., Pai, G., Kimmel, R.: Deep eikonal solvers. In: International conference on scale space and variational methods in computer vision. Springer (2019)
27. Madan, A., Levin, D.: Local surface parameterizations via smoothed geodesic splines. ACM Transactions on Graphics (2025)
28. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: CVPR (2019)
29. Mitchell, J.S.B., Mount, D.M., Papadimitriou, C.H.: The discrete geodesic problem. SIAM Journal on Computing **16**(4) (1987)
30. Pang, B., Zheng, Z., Li, Y., Wang, G., Wang, P.S.: Neural laplacian operator for 3d point clouds. ACM Trans. Graph. **43**(6) (Nov 2024)
31. Pang, B., Zheng, Z., Wang, G., Wang, P.S.: Learning the geodesic embedding with graph neural networks. ACM Trans. Graph. (SIGGRAPH ASIA) **42**(6) (2023)
32. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. In: CVPR (2019)
33. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: ECCV (2020)
34. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence (2018)
35. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: CVPR (2017)
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
37. Riegler, G., Ulusoy, A.O., Geiger, A.: OctNet: Learning deep 3D representations at high resolutions. In: CVPR (2017)
38. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention (2015)
39. Schmidt, R., Grimm, C., Wyvill, B.: Interactive decal compositing with discrete exponential maps. In: SIGGRAPH (2006)
40. Sharp, N., Attaiki, S., Crane, K., Ovsjanikov, M.: DiffusionNet: Discretization agnostic learning on surfaces. ACM Trans. Graph. **41**(3) (2022)
41. Sharp, N., Crane, K.: A Laplacian for nonmanifold triangle meshes. Comput. Graph. Forum (SGP) **39**(5) (2020)
42. Sharp, N., Crane, K.: You can find geodesic paths in triangle meshes by just flipping edges. ACM Trans. Graph. (SIGGRAPH ASIA) **39**(6) (2020)
43. Sharp, N., Soliman, Y., Crane, K.: Navigating intrinsic triangulations. ACM Trans. Graph. (SIGGRAPH) **38**(4) (2019)
44. Sharp, N., Soliman, Y., Crane, K.: The vector heat method. ACM Transactions on Graphics (ToG) (2019)

45. Shue, J.R., Chan, E.R., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3D neural field generation using triplane diffusion. In: CVPR (2023)
46. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: NeurIPS (2020)
47. Smith, J., Schaefer, S.: Bijective parameterization with free boundaries. ACM Trans. Graph. (2015)
48. Soliman, Y., Sharp, N.: The affine heat method. In: Computer Graphics Forum. Wiley Online Library (2025)
49. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. ACM Trans. Graph. **24**(3) (2005)
50. Team, G., Georgiev, P., Lei, V.I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
51. Verninas, H., Korkmaz, C., Zafeiriou, S., Birdal, T., Foti, S.: Parallelised differentiable straightest geodesics for 3d meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14637–14647 (2026)
52. Wang, P.S.: OctFormer: Octree-based transformers for 3D point clouds. ACM Trans. Graph. (SIGGRAPH) **42**(4) (2023)
53. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-CNN: Octree-based convolutional neural networks for 3D shape analysis. ACM Trans. Graph. (SIGGRAPH) **36**(4) (2017)
54. Wang, P.S., Liu, Y., Tong, X.: Mesh denoising via cascaded normal regression. ACM Trans. Graph. (SIGGRAPH ASIA) **35**(6) (2016)
55. Wang, P.S., Liu, Y., Tong, X.: Dual octree graph networks for learning adaptive volumetric shape representations. ACM Trans. Graph. (SIGGRAPH) **41**(4) (2022)
56. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**(5) (2019)
57. Wei, L.Y.: Parallel poisson disk sampling. Acm Transactions On Graphics (ToG) (2008)
58. Welschinger, L., Liu, Y., Wang, Z., Mitra, N.: Learning to solve pdes on neural shape representations. arXiv preprint arXiv:2512.21311 (2025)
59. Williams, F., Gojcic, Z., Khamis, S., Zorin, D., Bruna, J., Fidler, S., Litany, O.: Neural fields as learnable kernels for 3D reconstruction. In: CVPR (2022)
60. Xia, Q., Zhang, J., Fang, Z., Li, J., Zhang, M., Deng, B., He, Y.: GeodesicEmbedding (GE): a high-dimensional embedding approach for fast geodesic distance queries. IEEE. T. Vis. Comput. Gr. (2021)
61. Xin, S.Q., Wang, G.J.: Improving Chen and Han's algorithm on the discrete geodesic problem. ACM Trans. Graph. (SIGGRAPH) **28**(4) (2009)
62. Yu, Z., Foti, S., Zhang, L., Zhao, A., Keskin, C., Zafeiriou, S., Birdal, T., et al.: Geometric neural distance fields for learning human motion priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2232–2242 (2026)
63. Zhang, Q., Hou, J., Adikusuma, Y.Y., Wang, W., He, Y.: NeuroGF: A neural representation for fast geodesic distance and path queries. arXiv preprint arXiv:2306.00658 (2023)
64. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. arXiv preprint arXiv:2501.12202 (2025)
65. Zhou, Q., Jacobson, A.: Thingi10k: A dataset of 10,000 3d-printing models. arXiv preprint arXiv:1605.04797 (2016)