NeuroGF: A Neural Representation for Fast Geodesic Distance and Path Queries

Qijian Zhang[1], Junhui Hou[1*], Yohanes Yudhi Adikusuma[2], Wenping Wang[3], Ying He[2]

[1]Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China

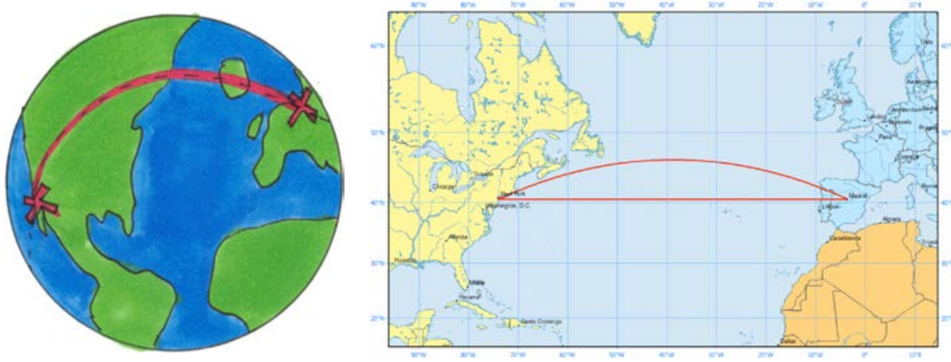
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore

[3]Department of Computer Science and Engineering, Texas A&M University, Texas, USA

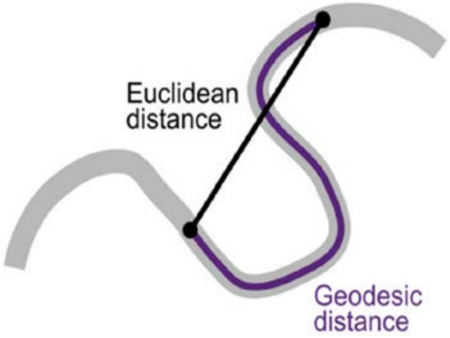
Background Introduction

Geodesics in 3D Geometry Processing

- A curve representing the shortest path between two points on a surface



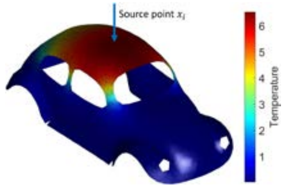
(a) flight paths travel along an arc

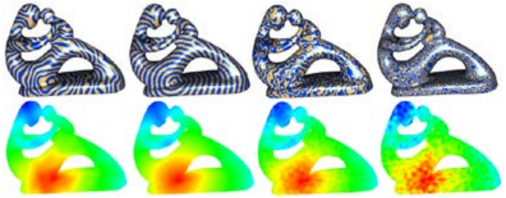


(b) shortest path along the surface

Wide Applications

- Shape analysis
- Correspondence
- Deformation
- Texture mapping
- …





Background Introduction

- **Previous Works**
  - Methods based on discrete wavefront propagation or geodesic graphs
    - ✓ Advantages: high-quality geodesics; arbitrary mesh triangulation
    - ✓ Disadvantages: computational inefficiency; cumbersome pre-computation
  - Methods based on partial differential equation
    - ✓ Advantages: flexibility; efficiency; ease of implementation
    - ✓ Disadvantages: sensitive to the quality of mesh triangulation
- **Task Objectives**
  - Encode geodesic distance and path fields using neural implicit representations
    - ✓ Compact storage
    - ✓ Fast query speed
    - ✓ Generalizable to unseen shapes/categories
    - ✓ Flexible for various data formats (e.g., mesh, point cloud)

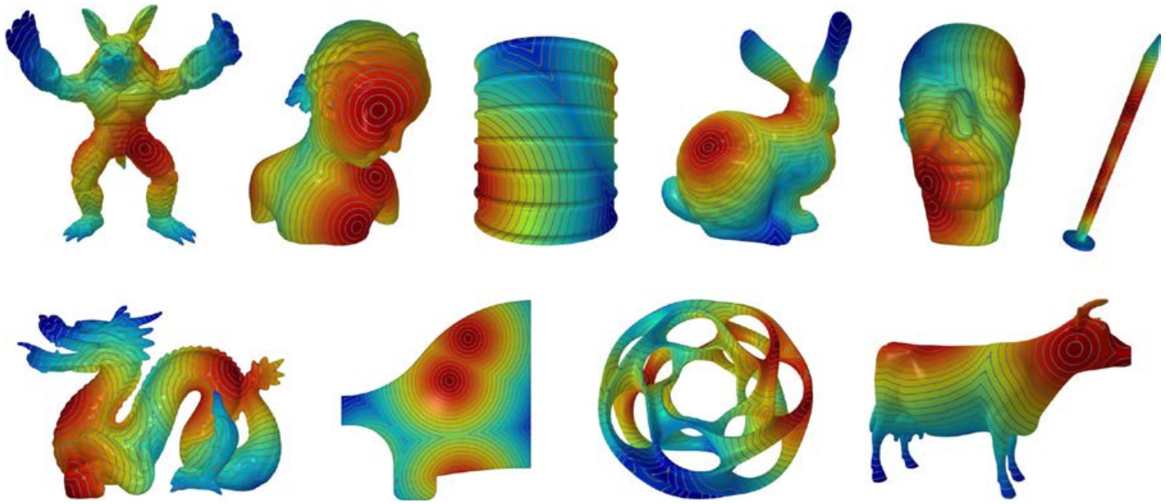
NEURAL INFORMATION PROCESSING SYSTEMS

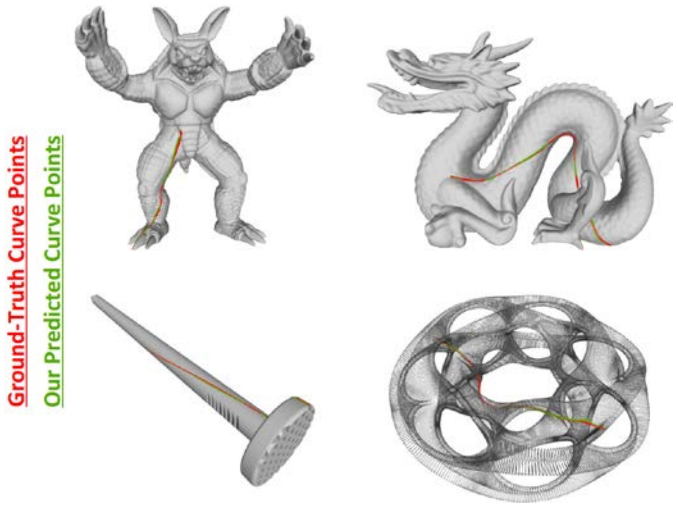
Problem Formulation

- Given a pair of source and target query points, the neural network is trained to output their geodesic distance and shortest path.

$$\{d, \mathbf{c}_{s\to t}\} = \mathcal{N}_\Theta(\mathbf{q}_s; \mathbf{q}_t)$$

- ✓ paired input queries: 3D points located on the underlying surface
- ✓ geodesic distance: a scalar value $d \in \mathbb{R}^+$
- ✓ shortest path: discretized as an ordered sequence of 3D points $\mathbf{c}_{s\to t} \in \mathbb{R}^{M\times 3}$ $\mathbf{c}_{s\to t} = \{\mathbf{p}_m\}_{m=0}^{M-1}$

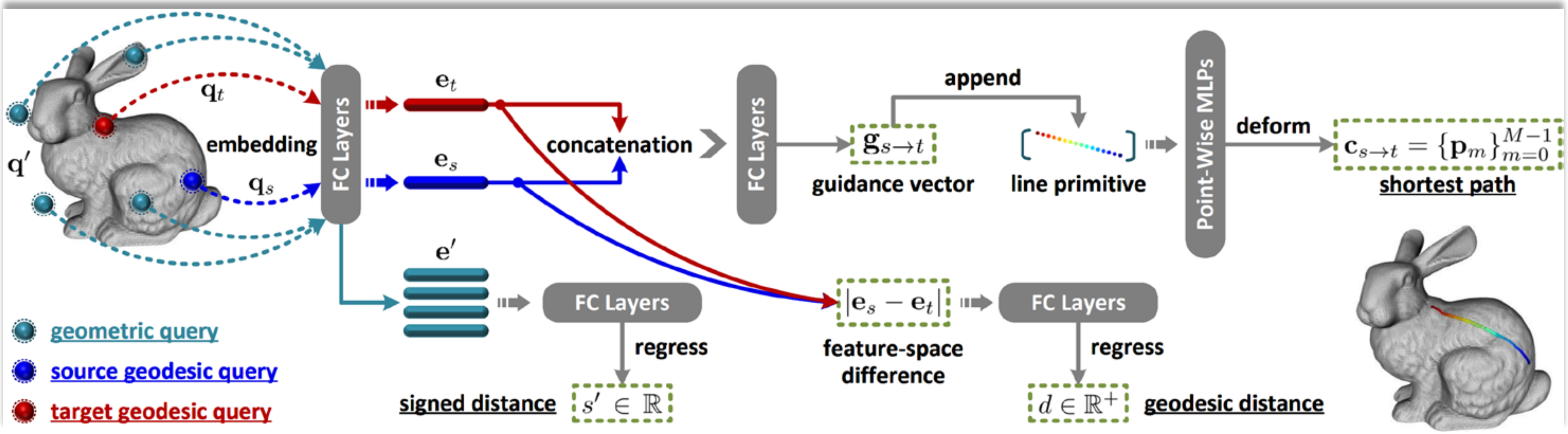




NEURAL INFORMATION PROCESSING SYSTEMS

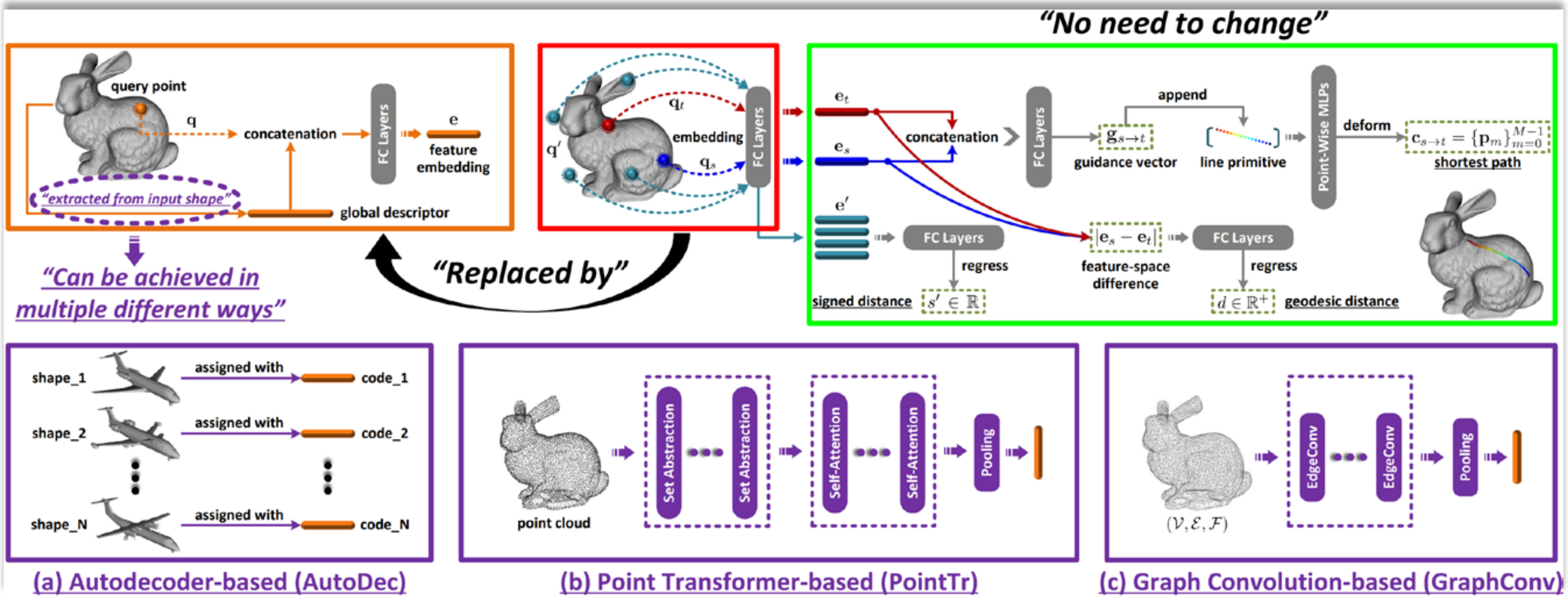
Overfitted Working Mode

- (1) Embed each 3D query point into the high-dimensional latent space
- (2) Regress the geodesic distance value from the absolute feature difference
- (3) Generate shortest path points through curve deformation
  - ✓ (a) initial line segment
  - ✓ (b) feature guidance
  - ✓ (c) curve points generation
- (4) Regress the signed distance value from each query point embedding



NEURAL INFORMATION PROCESSING SYSTEMS

- **Generalizable Working Mode**
  - Replace the original query point embedding with a feature extractor
    - ✓ (a) autodecoder-based (i.e., DeepSDF-like)
    - ✓ (b) transformer-based
    - ✓ (c) graph-based



NEURAL INFORMATION PROCESSING SYSTEMS

Loss Functions

- Supervision of signed and geodesic distances

$$\ell_{\text{sdist}} = \|s' - \tilde{s}'\|_1 \qquad \ell_{\text{gdist}} = \left\| d - \tilde{d} \right\|_1$$

- Supervision of shortest paths

$$\ell_{\text{spath}} = \|\mathbf{c}_{s\to t} - \tilde{\mathbf{c}}_{s\to t}\|_1$$

- Consistency constraint of curve lengths

$$\ell_{\text{ccl}} = \left\| \sum_{m=1}^{M-1} (\|\tilde{\mathbf{p}}_m - \tilde{\mathbf{p}}_{m-1}\|_2) - \sum_{m=1}^{M-1} (\|\mathbf{p}_m - \mathbf{p}_{m-1}\|_2) \right\|_1$$

- Distribution constraint of curve points

$$\ell_{\text{dcp}} = \frac{1}{M} \sum_{m=0}^{M-1} |\mathcal{N}_\phi(\mathbf{p}_m)|$$

where $\mathcal{N}_\phi : \mathbb{R}^3 \to \mathbb{R}$ represents an independent neural model overfitted on the given shape for the fitting of signed distance fields in advance, whose network parameters are fixed. Given an arbitrary spatial query, $\mathcal{N}_\phi$ outputs a scalar of the corresponding signed distance value, offering a natural way of constraining the generated curve points in a differentiable manner.

Results of Single-Source All-Destination (SSAD) Querying and SDF Fitting

Table 1: Comparison of geodesic representation accuracy and time efficiency for SSAD querying.

Mesh	*#V (K)*	τ	*Running Time (ms) of SSAD Query*				*Mean Relative Error (%)*		
			VTP [34]	HM [10]	fDGG [2]	NeuroGF	HM [10]	fDGG [2]	NeuroGF
armadillo	173	1.3	1778	194	59	0.5	1.03	0.59	0.51
bimba	75	1.1	985	82	20	0.5	0.67	0.57	0.46
bucket	35	14.1	500	18	16	0.5	3.35	0.96	0.18
bunny	35	1.4	374	29	10	0.5	0.87	0.58	0.44
cow	46	1.6	593	28	11	0.5	2.19	0.57	0.51
dragon	436	12.7	6209	246	145	0.7	10.6	0.46	0.68
fandisk	20	1.4	359	14	4	0.5	0.88	0.66	0.35
heptoroid	287	2.6	5789	212	86	0.6	1.75	0.48	0.87
maxplanck	49	1.2	797	33	11	0.5	0.79	0.57	0.39
nail	2.4	4.6	16	1.4	0.6	0.4	2.71	0.42	0.50

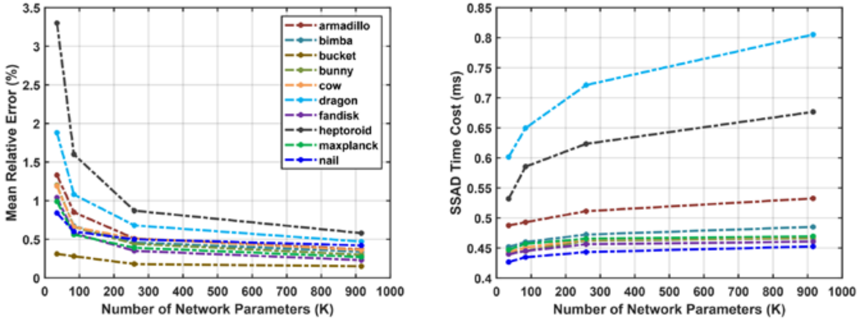


Figure 6: Statistics of SSAD geodesic distance querying with different network complexity.

Mesh	*Chamfer-L_1 ($\times 10^{-2}$)*
armadillo	1.366
bimba	1.301
nail	0.354
bunny	1.559
cow	0.941
dragon	1.319
fandisk	0.822
heptoroid	2.244
maxplanck	1.434
bucket	1.183

Table 2: Chamfer-L_1 errors between ground-truth and our predicted shortest path points.

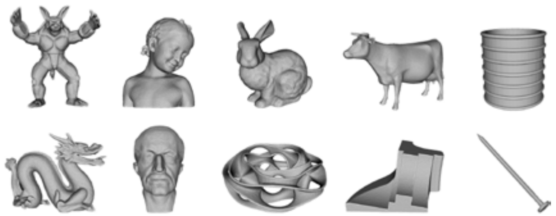


Figure 13: Visualization of mesh reconstruction deduced from NeuroGFs.

Table 5: Chamfer-L_1 ($\times 10^{-2}$) errors between ground-truth and our predicted shortest path points after post-processing.

armadillo	bimba	nail	bunny	cow	dragon	fandisk	heptoroid	maxplanck	bucket	**average**
1.131	1.126	0.347	1.325	0.804	1.070	0.766	1.994	1.248	1.079	**1.09**

Table 9: Mean L_1 errors between our predicted and ground-truth signed distances.

Mesh	armadillo	bimba	bucket	bunny	cow	dragon	fandisk	heptoroid	maxplanck	nail
L_1 ($\times 10^{-3}$)	1.22	0.97	0.68	0.97	0.86	1.28	0.67	1.01	1.03	0.45

Figure 12: Experimental evaluations on large-scale real-world meshes (with up to 2 million vertices).

NEURAL INFORMATION PROCESSING SYSTEMS

Performances of Generalizable NeuroGF

Table 3: MRE (%) performances of generalizable NeuroGF learning frameworks equipped with different global shape feature extractors, including: (a) AutoDec, (b) PointTr, and (c) GraphConv.

(a)	SN-Airplane	SN-Chair	SN-Car	(b)	SN-8x50	SN-5x50	(c)	SN-8x50	SN-5x50
AutoDec	3.03	3.91	2.78	*PointTr*	3.28	4.16	*GraphConv*	2.94	3.55

Ablation Studies

Table 4: Influences of different learning components and supervision objectives, where the results are averaged on all testing shapes. The right two columns show the representation accuracy of our predicted geodesic distances and shortest paths (the lower, the better). The averaged statistics of our full implementation in terms of the two metrics are 0.49% and 1.25×10^{-2}. In particular, we mark the relative change within each bracket to facilitate comparison.

$\mathcal{B}_{\text{gdist}}$	$\mathcal{B}_{\text{spath}}$	$\mathcal{B}'_{\text{sdist}}$	ℓ_{ccl}	ℓ_{dcp}	*Mean Relative Error (%)*	*Chamfer-L_1 ($\times 10^{-2}$)*
✗					-	1.46 (↑ 0.21)
	✗				0.61 (↑ 0.12)	-
		✗			0.57 (↑ 0.08)	1.31 (↑ 0.06)
			✗		0.52 (↑ 0.03)	1.35 (↑ 0.10)
				✗	0.50 (↑ 0.01)	1.38 (↑ 0.13)

Table A7: Ablation studies on different variants of our technical implementations, where "variant (1)" means adding position encoding before fed into FC layers for query point feature embedding, "variant (2)" means replacing L_1 loss with L_2 loss for supervisions, and "variant (3)" means computing Chamfer distance between generated and ground-truth curve points of shortest paths $\mathbf{c}_{s \to t}$ and $\tilde{\mathbf{c}}_{s \to t}$ for the formulation of ℓ_{spath} (Eq. (8) in the paper).

Implementation Variants	*dragon*		*heptoroid*	
	MRE	*Chamfer-L_1*	*MRE*	*Chamfer-L_1*
(0) Original Implementation	0.68	1.319	0.87	2.244
(1) Adding PosEnc	0.62	1.253	0.79	1.921
(2) L_2 Supervisions	0.71	1.345	0.88	2.227
(3) Chamfer Loss for ℓ_{spath}	0.66	1.286	0.82	2.149

THANKS FOR LISTERNING!