Geo2Vec: Shape- and Distance-Aware Neural Representation of Geospatial Entities

Chen Chu, Cyrus Shahabi

University of Southern California
Los Angeles, CA USA
chenchu@usc.edu, shahabi@usc.edu

Abstract

Spatial representation learning is essential for GeoAI applications such as urban analytics, enabling the encoding of shapes, locations, and spatial relationships (topological and distance-based) of geo-entities like points, polylines, and polygons. Existing methods either target a single geo-entity type or, like Poly2Vec, decompose entities into simpler components to enable Fourier transformation, introducing high computational cost. Moreover, since the transformed space lacks geometric alignment, these methods rely on uniform, non-adaptive sampling, which blurs fine-grained features like edges and boundaries. To address these limitations, we introduce Geo2Vec, a novel method inspired by signed distance fields (SDF) that operates directly in the original space. Geo2Vec adaptively samples points and encodes their signed distances (positive outside, negative inside), capturing geometry without decomposition. A neural network trained to approximate the SDF produces compact, geometry-aware, and unified representations for all geo-entity types. Additionally, we propose a rotation-invariant positional encoding to model high-frequency spatial variations and construct a structured and robust embedding space for downstream GeoAI models. Empirical results show that Geo2Vec consistently outperforms existing methods in representing shape and location, capturing topological and distance relationships, and achieving greater efficiency in real-world GeoAI applications. Code and Data can be found at:

https://github.com/chuchen2017/GeoNeuralRepresentation.

Introduction

Representation learning for geospatial entities, such as points, lines, and polygons, has become crucial for deep neural network models aiming to effectively address various downstream geospatial tasks. The ability to learn robust and unified embeddings for these entities facilitates generalization across diverse GeoAI applications, including land use classification (Li et al. 2023), population prediction (Boo et al. 2022), urban flow inference (Balsebre et al. 2024), and urban morphology analysis (Wu et al. 2025).

Several Spatial Representation Learning (SRL) approaches have been developed specifically for individual entity types like lines and polygons. For example, polyline-based methods often employ sequence models like RNN or Transformer (Li et al. 2024), but these approaches primarily capture vertex connectivity and largely overlook crucial geometric and topological details associated with line segments. Similarly, polygon-specific methods typically use graph neural networks (GNNs) to represent vertices and edges as graph components (Yu et al. 2024a), yet these methods inadequately preserve the polygons' spatial extent (interior and exterior) and often struggle with complex geometries, particularly polygons with holes.

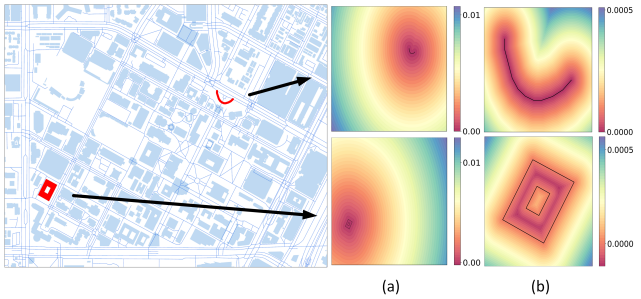


Figure 1: Signed Distance Fields for two types of geo-entities at spatial scales: (a) coarse scale, (b) fine scale.

To enable unified embeddings across all geo-entity types, recent methods like Poly2Vec (Siampou et al. 2025) decompose complex entities into simpler components suitable for Fourier transformation. This decomposition, combined with the computational overhead of performing the Fourier transformation itself, results in high processing cost. Moreover, since the transformed Fourier space lacks direct correspondence with the original geometry and topology, these methods are limited to uniform, non-adaptive sampling, which fails to preserve fine-grained geometric features like edges and boundaries.

Consequently, there is a need for a unified SRL approach that captures geometry and location across all geospatial entity types while performing well on standard evaluation tasks like capturing topological and distance relationships (Ji et al. 2025), to ensure effectiveness in real-world GeoAI tasks.

Towards this end, we propose Geo2Vec, a neural representation approach that explicitly learns a Signed Distance Field (SDF) of each geospatial entity. Specifically, the SDF is defined as the shortest distance from any point in space to the boundary of the entity, with negative values indicating points inside the entity and positive values outside. Ex-

amples of SDFs for polygon- and polyline-type geo-entities are shown in Figure 1. In Figure 1(a), the coarse-scale SDF clearly captures the spatial locations of the two geo-entities, with low-value regions highlighted in red. In contrast, the fine-scale SDFs in Figure 1(b) capture the detailed shapes of the entities as continuous fields. Notably, Geo2Vec's use of SDFs enables a **unified** representation across all geo-entity types: polygons with spatial extent (including those with holes) yield negative SDF values within their interiors, while points and lines, lacking interior regions, do not. This continuous and differentiable representation overcomes all the limitations of discrete vertex-edge models and entity-specific decompositions.

Moreover, Geo2Vec leverages coordinates in the original (non-transformed) space, allowing strategic sampling near entity boundaries or regions requiring higher precision. This adaptive sampling significantly enhances representation quality. Notably, our experiments show that our adaptive sampling method significantly outperform Fourier space sampling, achieving comparable accuracy with less than 35% of samples and thus delivering superior efficiency.

Our empirical evaluations show that Geo2Vec significantly outperforms SOTA methods on standard evaluation tasks for shape and location representation, achieving improvement up to 61.95% and 54.3%, respectively. Finally, we introduce a rotation-invariant positional encoding that produces a more structured and robust embedding space, where geo-entities with similar shapes are positioned closer together regardless of their orientation. This property is useful for unsupervised models and supervised models with weaker learning signals, and our experiments specifically demonstrate its effectiveness in improving Geo2Vec performance in unsupervised downstream tasks.

Related Works

Spatial Representation Learning (SRL) aims at directly learning the neural representation of various types of spatial data in their native format without the need for feature engineering and data conversion stage (Mai et al. 2024). Most prior work has focused on learning representations for different types of spatial data in isolation. For example, point encoding (Mai et al. 2023b), trajectory representation learning (Jiang et al. 2023), road network representation (Zhao et al. 2025), and polygon representation (Huang et al. 2024). Current methods primarily rely on feeding the discrete data structures directly into neural models to learn representations (Ma et al. 2024; Yu et al. 2024a). Although, the discrete vertex/edge representation is suitable for data storage and visualization clarity, it is not effective for representing spatial extent or their topological characteristics. This mismatch between representation format and geo-entity leads to limitations in expressiveness.

The current state-of-the-art method, Poly2Vec (Siampou et al. 2025), encodes points, polylines, and polygons using Fourier transforms. However, existing research has not revealed the relationship between real-world coordinates and the Fourier feature space (Spectral domain). As a result, Fourier-based methods typically employ non-adaptive, heuristic sampling strategies. Although some approaches incorporate geometric frequency selection and improve feature expressiveness (Mai et al. 2023a), they still fall short of identifying the most discriminative frequencies. This limitation fundamentally restricts the representational power of Fourier-based encoding, particularly when sampling frequencies are low. Moreover, applying Fourier transforms to complex objects like polygons is nontrivial, which is why they must first be decomposed into simpler shapes like triangles, further adding to the already high computational cost of the transformation.

Employing neural networks to learn continuous fields is a widely studied topic in 3D computer vision. Prior work has explored learning 3D shape representations through signed distance function (Park et al. 2019; Yu et al. 2024b) and occupancy fields (Mescheder et al. 2019). These field learning methods have shown strong effectiveness in modeling complex scenarios, as shown by NeRF (Mildenhall et al. 2021) and 3DGS (Kerbl et al. 2023). However, most of this research focuses on accurately reconstructing specific shapes or scenes, rather than leveraging field-based representations for broader downstream geospatial tasks. In contrast, our work aims to learn a **generalizable embedding space** from SDFs, explicitly designed to efficiently capture geospatial semantics and (topological and distance) relationships.

Preliminary

Definition 1: **Spatial Position** refers to the location of a geo-entity expressed in geographic coordinates or a global reference system, representing its precise placement in physical or world space.

Definition 2: **Spatial Extent** refers to the coverage area of a geo-entity, representing its shape and spatial footprint.

Definition 3: A **Geo-entity** E is an object characterized by its spatial position and, optionally, its spatial extent. It is generally recorded in a sequence of coordinates $P_E = \{\mathbf{x}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 2}$, where $\mathbf{x}_i = (x_i, y_i)$ denotes a vertex, and N denotes the number of vertices. Common examples include points, polylines, polygons, multi-polygons, and other related spatial data types.

Definition 4 (Signed Distance Function). Given a geo-entity E and a query point in space $x \in \mathbb{R}^2$, the signed distance function $\mathbf{SDF}(\mathbf{x}, E) = s$ returns the shortest distance $s \in \mathbb{R}$ from the query point $\mathbf{x}$ to the boundary of E. For any geo-entity with spatial extent, the value s is positive if $\mathbf{x}$ lies outside the spatial extent of E, and negative if $\mathbf{x}$ lies inside.

Definition 5 (Signed Distance Field). The Signed Distance Field of a geo-entity E is a scalar field defined over a continuous spatial domain $\Omega_E \subseteq \mathbb{R}^2$, in which each point $\mathbf{x} \in \Omega_E$ is assigned a scalar value representing its signed distance to E. This field provides a continuous representation of the geo-entity, capturing both location and shape information.

Definition 6 (Representation Learning of Geo-entity). Given a dataset of geo-entities $G = \{E_i\}_{i=1}^{N}$, the goal of representation learning is to learn a mapping function $\mathcal{E}_\theta : E \rightarrow \mathbf{z}_E \in \mathbb{R}^d$, where $\mathbf{z}_E \in \mathbb{R}^d$ is a d-dimensional embedding. The learned representations should preserve the data utility of the original formats, allowing effective support for a range of spatial reasoning tasks. Moreover, by unifying different

types of geo-entities into a common representation embedding, the representation becomes broadly applicable across diverse downstream models.

Nueral Representation of Geo-entity

In this section, we present Geo2Vec, which aims to learn the representation of a geo-entity by explicitly modeling its SDF. We employ a neural network $\mathcal{G}_\theta$ to approximate the signed distance function $\mathbf{SDF}(\mathbf{x}, E)$ and learn the corresponding SDF Ω_E. To achieve this, for each geo-entity E, we sample a set of training points $X_E := \{(\mathbf{x}, s) \mid s = \mathbf{SDF}(\mathbf{x}, E)\}$. Then train the neural network $\mathcal{G}_\theta$ to learn the underlying SDF based on sample points.

For polygon shape learning, we scale each polygon individually to a canonical space $[-1, 1] \times [-1, 1]$ and then learn its scale-invariant shape embedding $\mathbf{z}_E^{shp}$. For location representation learning, we normalize the entire dataset G to a canonical space and then learn the location representation of each geo-entity $\mathbf{z}_E^{loc}$. The final representation is formed by concatenating the location and shape vectors: $\mathbf{z}_E = [\mathbf{z}_E^{loc}, \mathbf{z}_E^{shp}]$. For point entities, we use a uniform vector as their shape representation $\mathbf{z}_E^{shp}$. The learning pipelines for shape and location representation are identical, the major differences lie in the sampling strategy and the positional encoding method.

An Adaptive Sampling Strategy

One key advantage of representing a geo-entity using its SDF is that it allows us to directly sample in the coordinate space. To better leverage this property, we propose an adaptive sampling strategy that adjusts sampling parameters based on the learning objective and characteristics of the dataset. We first introduce our sampling methods, then describe how the associated parameters are tuned accordingly.

Firstly, we sample N_{Vertex} points near each vertex V of a geo-entity E following a 2D normal distribution:

$$\mathbf{x}_i \sim \mathcal{N}(P_E, \sigma^2 I)$$

where σ is the standard deviation controlling the sampling radius. Each sampled point is paired with its signed distance value to construct the training set X_E^{Vertex}:

$$X_E^{\text{Vertex}} = \{\mathbf{x}_i, \mathbf{SDF}(\mathbf{x}_i, E)\}_{i=1}^{N_{\text{Vertex}}}$$

To enhance boundary coverage, we introduce stochastic perpendicular sampling, which perturbs sampled points along each edge by applying a small normal-direction offset drawn from a symmetric distribution. Formally, for any two continuous points $\mathbf{x}_i, \mathbf{x}_{i+1} \in P_E$, P_E denotes the sequence of coordinates of entity E, we sample $\mathbf{x}'$ according to the following formulate:

$$\mathbf{x}' = (1-f)\mathbf{x}_i + f\mathbf{x}_{i+1} + s \cdot d \cdot \frac{1}{||p2 - p1||} \cdot \begin{pmatrix} -(y_{i+1} - y_i) \\ x_{i+1} - x_i \end{pmatrix}$$

where $f \sim \mathcal{U}(0, 1)$ controls the position along the edge, $d \sim \mathcal{N}(0, \sigma^2 I)$ specifies the magnitude of the perpendicular offset, and $s \sim \mathcal{U}\{-1, +1\}$ randomly selects the side of the edge. Stochastic perpendicular sampling improves the model's ability to capture the edge position in the SDF. Similarly, we construct the training dataset by sampling N_{Edge} points for each $Edge$:

$$X_E^{\text{Edge}} = \{\mathbf{x}_i, \mathbf{SDF}(\mathbf{x}_i, E)\}_{i=1}^{N_{\text{Edge}}}$$

Lastly, we uniformly sample points from the coordinate space to capture the global structure of the geo-entity and to fill in regions that may have been overlooked by the previous two sampling stages. Specifically, we sample N_{axis} points along each axis, resulting in a total of $N_{\text{Space}} = {N_{\text{axis}}}^2$ uniformly distributed points across the space, which constitute the dataset X_E^{space}.

$$X_E^{\text{Space}} = \{\mathbf{x}_i, \mathbf{SDF}(\mathbf{x}_i, E)\}_{i=1}^{N_{\text{Space}}}$$

Finally, we combine all sampled points to form our training dataset:

$$X_E = \{X_E^{\text{Vertex}}, X_E^{\text{Edge}}, X_E^{\text{Space}}\}_{\text{Vertex} \in E, \text{Edge} \in E}$$

During the sampling process, we leave several parameters flexible, allowing Geo2Vec to adaptively sample based on the data distribution of the target dataset.

When learning the location representation, which aims to capture the spatial relationships among geo-entities, it is important to provide the model with information about its local neighborhood. Therefore, we set the sampling parameters according to the distances between geo-entities. It is worth noting that, although it would be beneficial to sample a variable number of points for different geo-entities, we fix the following parameters as global constants within each dataset to ensure computational efficiency. Specifically, we randomly sample a subset of geo-entities E, compute the distances to their k nearest neighbors, and define the location sampling parameter σ_{loc} as the standard deviation of the resulting distance distribution.

For learning the shape representation, we follow a similar strategy, but base it on edge distances. A subset of geo-entities E is randomly selected, and for each of their edges, we compute the distances to their top k nearest edges. The standard deviation of these distances is then used to define the shape sampling parameter σ_{shp}.

After determining the sampling deviation σ, we introduce a resolution parameter ϵ to decide the number of points to sample per unit. This parameter controls how finely we capture local spatial variation. The number of samples is given by: $N_{\text{Vertex}} = \pi\sigma^2\epsilon^2$, $N_{\text{Edge}} = 2\sigma l_{\text{Edge}}\epsilon^2$, where l_{Edge} denotes the length of the $Edge$. For computational simplicity, we approximate them as: $N_{\text{Vertex}} = \epsilon \cdot \sigma$, $N_{\text{Edge}} = \epsilon \cdot l_{\text{Edge}}$, which retains the core dependency on resolution, neighborhood spread, and edge length.

Positional Encoding

SDF shows various spatial patterns in different scales, and successfully modeling these patterns is crucial for its representation. We employ a Positional Encoding (PE) that maps the spatial coordinate $\mathbf{x} \in \mathbb{R}$ to a higher dimensional space $\mathbb{R}^{2L}$, providing spatial features that encode local and global

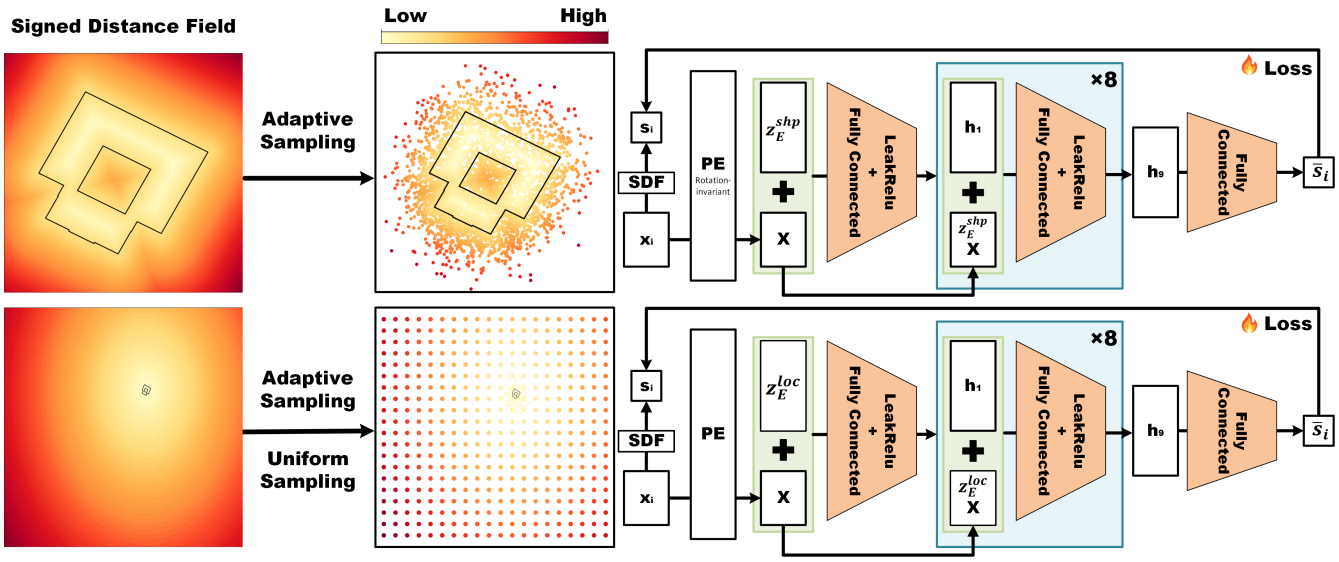


Figure 2: An illustration of the Geo2Vec learning framework.

signed distance variation. The positional encoding is formulated as follows:

$$PE(\mathbf{x}) = (\sin(2^{L_{\min}}\pi\mathbf{x}), \cos(2^{L_{\min}}\pi\mathbf{x}), \ldots, \sin(2^{L_{\max}}\pi\mathbf{x}), \cos(2^{L_{\max}}\pi\mathbf{x}))$$

where $L_{\min}$ and $L_{\max}$ define the lower and upper bounds of frequency levels, and we uniformly sample L frequencies in this bound. Unlike the positional encoding used in Transformer and NERF, we do not predefine these bounds. Instead, we set $L_{\min}$ and $L_{\max}$ based on the distribution of geo-entities and the specific learning objective.

Positional encoding is essential for both shape and location learning, but for opposite reasons. When learning location representations, the model aims to capture the coarse-scale trends of SDF, which, as shown in Figure 1(a), decreases uniformly in all directions and is largely independent of the specific shape of the entity. Thus, positional encoding helps encode such global variation and is expected to generalize across geo-entities. In this case, high-frequency components will introduce repeated features that hinder learning of smooth global patterns, so we avoid using these repeated frequencies when leaning location representation.

In contrast, shape representation learning focuses on capturing fine-grained, local variations in SDFs that are unique to each geo-entity, as shown in Figure 1 (b). Modeling such fine spatial variations requires encoding the input coordinates with high-frequency signals, which enables the model to represent sharp transitions. These patterns are typically difficult for neural networks to learn from smooth coordinate inputs alone. Therefore, we have the following settings for positional encoding:

$$\Delta_x = \max_{E\in G}(E.x) - \min_{E\in G}(E.x), \Delta_y = \max_{E\in G}(E.y) - \min_{E\in G}(E.y)$$

$$\Delta_{\min} = \min(\Delta_x, \Delta_y), \Delta_{\max} = \max(\Delta_x, \Delta_y)$$

$$L^{\text{loc}}_{\max} \le \log_2\left(\frac{2}{\Delta_{\min}}\right), \quad L^{\text{shp}}_{\max} \ge \log_2\left(\frac{2}{\Delta_{\min}}\right)$$

$$L^{\text{loc}}_{\min}, L^{\text{shp}}_{\min} \le 1 - \log_2(\Delta_{\max})$$

Following the rules, $L_{\max}$ and $L_{\min}$ can be determined according to the dataset. We leave the number of sampling frequencies L as a hyperparameter.

When learning shape representations, it is important that the model learns similar embeddings for geo-entities with the same shape but different orientations. To achieve that, we propose a rotation-invariant positional encoding method, which can be formulated as:

$$PE_R(\mathbf{x}) = PE(\mathbf{x}'), \quad \text{where} \quad \mathbf{x}' = \begin{bmatrix} x \\ y \\ r \end{bmatrix}, r = \sqrt{x^2 + y^2}$$

The method transforms each point's Cartesian coordinates $\mathbf{x}$ into polar coordinates and augments the original input with the radial distance r. This augmentation introduces rotation-invariant features into the positional encoding, encouraging the model to capture shape geometry rather than absolute orientation. As a result, the learned embeddings become more structured and robust—a property that is particularly valuable for unsupervised downstream GeoAI models.

Geo2Vec Model

Given a set of sampled points, we propose the Geo2Vec model $\mathcal{G}_\theta$ to learn the SDF of a geo-entity based on sampled points X_E:

$$\mathcal{G}_\theta(X_E, \mathbf{z}_E) \approx \mathbf{SDF}(X_E, E), \quad \forall \mathbf{x} \in X_E \subset \Omega_E$$

Following (Park et al. 2019), we formulate this problem from a probabilistic perspective. We define the posterior distribution over the latent code $\mathbf{z}_E$ given the sampled points X_E as:

$$p_\theta(\mathbf{z}_E \mid X_E) = p(\mathbf{z}_E) \prod_{(\mathbf{x}_i, s_i)\in X_E} p_\theta(s_i \mid \mathbf{z}_E; \mathbf{x}_i)$$

Table 1: Model accuracy on Shape Classification (**Shape**) and MAE on Predicting the Number of Edges (**Edge**). All accuracy values are scaled by $\times 10^{-2}$. In all tables, the values after ± indicate the standard-deviation, and **Best** results are highlighted.

	Building		MNIST		Singapore	NYC
	Shape↑	Edge↓	Shape↑	Edge↓	Edge↓	Edge↓
PolyGNN	87.84±0.005	–	7.77±0.013	–	–	–
NUFTSPEC	90.46±0.730	3.04±0.125	96.90±0.116	16.76±0.588	3.66±0.023	1.45±0.020
Poly2Vec	76.59±1.403	3.34±0.127	92.52±0.265	29.29±0.807	3.68±0.109	1.21±0.045
Geo2Vec	**97.34±0.310**	**2.22±0.050**	**97.58±0.097**	**9.45±0.124**	**1.40±0.011**	**0.72±0.002**

Table 2: Model performance on Length of Line prediction, evaluated by MAE. All values are scaled by $\times 10^{-4}$

	Singapore	NYC
	Line Length↓	Line Length↓
T2Vec	10.38±0.45	13.20±0.42
T-JEPA	10.25±0.54	12.65±0.36
Poly2Vec	13.55±0.79	21.11±0.48
Geo2Vec	**5.75±0.26**	**7.07±0.16**

where $p(\mathbf{z}_E)$ denotes the prior distribution over latent codes, which we assume to follow a multivariate Gaussian $\mathcal{N}(\mathbf{0}, \sigma_z^2\mathbf{I})$, σ_z controls the density of the latent distribution. And the conditional likelihood $p_\theta(s_i \mid \mathbf{z}_E; \mathbf{x}_i)$, can be expressed as:

$$p_\theta(s_i \mid \mathbf{z}_E; \mathbf{x}_i) \propto \exp\left(-\mathcal{L}(\bar{s}_i, s_i)\right)$$

where $\bar{s}_i = \mathcal{G}_\theta(\mathbf{z}_E, \mathbf{x}_i)$ is the predicted signed distance at coordinate $\mathbf{x}_i$, and $\mathcal{L}$ is SDF the loss function.

Therefore, maximizing the posterior probability $p_\theta(\mathbf{z}_E \mid X_E)$ is equivalent to minimizing the summed loss between the predicted and observed signed distances, along with a regularization term on the latent code. This formulation justifies that approximating the signed distance field using $\mathcal{G}_\theta$ directly induces learning of the optimal latent representation $\mathbf{z}_E$ for each geo-entity. Therefore, the resulting loss function for training Geo2Vec over the dataset G can be expressed as:

$$\mathcal{L}_{\text{Geo2Vec}} = \sum_{E \in G} \left(\sum_{(\mathbf{x}_i, s_i) \in X_E} \mathcal{L}(\mathcal{G}_\theta(\mathbf{z}_E, \mathbf{x}_i), s_i) + \frac{\gamma}{\sigma_z^2} \|\mathbf{z}_E\|_2^2 \right)$$

where γ is a hyperparameter that controls how strongly the latent codes are encouraged to follow the prior distribution. We set $\gamma = 0$ when learning location representations, as the spatial variation across geo-entities is sufficiently large. In this case, enforcing a tightly clustered latent space can negatively impact learning by suppressing the natural diversity of location information.

During the training process, to encourage the latent representations to reside in a shared and structured space, we jointly optimize the posterior over a large batch that includes as many geo-entities as possible. This joint training helps the model learn consistent and meaningful representations across different entities. The detailed representation learning algorithm is described in Algorithm 1.

The architecture of the Geo2Vec network is shown in Figure 2. Instead of using ReLU, we employ LeakyReLU as the activation function, as learning the negative interior structure is also crucial, and LeakyReLU preserves a non-zero gradient in the negative domain. The input point $\mathbf{x}_i$ is projected by PE to $\mathbf{X}$, which is then concatenated with the latent representation $\mathbf{z}_E$. This combined vector is then concatenated with the hidden states of the neural network at each layer, serving as a conditioning input for prediction.

Algorithm 1: Geo2Vec Training Algorithm

Input: $G = \{E\}$
Input: Sample Density ϵ, Uniform N_{axis}, batch size b
Output: $\{\mathbf{z}_E\}_{E \in G}$

Initialize $\mathcal{G}_\theta$, $\{z_E\}_{E \in G} \sim \mathcal{N}(\mathbf{0}, \sigma_z^2\mathbf{I})$
Initialize N_{Edge}, N_{Vertex}, σ, $X_G = \{\}$
for each $E \in G$ **do**
  $X_E \sim \text{Sample}(E, N_{\text{Edge}}, N_{\text{Vertex}}, N_{\text{axis}}, \sigma)$
  $X_G \leftarrow X_G \cup \{(\mathbf{x}_i, s_i, E) \mid (\mathbf{x}_i, s_i) \in X_E\}$
end for
shuffle $X_G = \{(x_i, s_i, E_i)\}$
for each mini-batch $\{(x_i, s_i, E_i)\}_{i=1}^b \subset X_G$ **do**
  $\mathcal{L} = \mathcal{L}_{\text{Geo2Vec}}(\{(x_i, s_i, E_i)\}_{i=1}^b)$
  Update $\{\mathbf{z}_{E_i}\}_{E_i \in b}$ and $\mathcal{G}_\theta$ using $\mathcal{L}$
end for
return $\{\mathbf{z}_E\}_{E \in G}$

Experimental Evaluation

We evaluated SRL methods based on their effectiveness in capturing shape and location. To further assess the quality of the learned representations, we tested them within a downstream GeoAI model. Details on the **experimental setup**, **hyperparameter sensitivity analysis**, **visualization results** and **performance discussions** are provided in the appendix.

Datasets

We used four datasets in our experiments: two with shape labels to evaluate the model's performance on shape representation, and two real-world datasets to assess generalization in practical scenarios.

MNIST(Lecun et al. 1998): The original rasterized images are converted into polygon representations, and all digit shapes are randomly placed within a unit space. It contains 60,000 polygons, labeled according to its digit class.

Building(Yan et al. 2021): This dataset contains 5,000 building footprints, each manually labeled based on its geometric shape, where includes 10 common categories, such as E-shape, T-shape.

Table 3: Overall model performance on distance estimation, evaluated by MAE. All values are scaled by $\times 10^{-3}$.

	Building	MNIST	Singapore			NYC		
	Pg-Pg↓	Pg-Pg↓	Pt-Pg↓	Pl-Pg↓	Pg-Pg↓	Pt-Pg↓	Pl-Pg↓	Pg-Pg↓
TILE	217.1±1.6	223.8±1.0	99.9±1.7	115.5±1.5	114.3±1.4	127.6±0.7	154.3±2.1	167.4±2.0
THEORY	7.3±4.3	34.2±1.0	24.3±0.9	25.0±0.4	25.0±1.1	26.2±1.1	26.6±0.5	27.4±0.7
Poly2Vec	13.1±1.0	21.0±0.4	15.9±0.6	19.9±1.7	22.0±0.6	28.7±1.4	28.5±0.4	52.7±0.8
Geo2Vec	**6.4±0.9**	**13.0±0.8**	**5.4±0.5**	**5.0±0.1**	**5.5±0.4**	**10.2±0.1**	**13.0±0.9**	**12.9±0.6**

Table 4: Model accuracy on Topological Relationship Classification. All values are scaled by $\times 10^{-2}$.

	Singapore					NYC				
	Pt-Pl↑	Pt-Pg↑	Pl-Pl↑	Pl-Pg↑	Pg-Pg↑	Pt-Pl↑	Pt-Pg↑	Pl-Pl↑	Pl-Pg↑	Pg-Pg↑
NUFTSPEC	–	–	–	–	60.2±0.9	–	–	–	–	58.5±0.8
T2VEC	–	–	72.8±2.3	–	–	–	–	80.7±12.1	–	–
T-JEPA	–	–	75.4±1.8	–	–	–	–	79.8±8.6	–	–
DIRECT	82.3±1.3	84.3±0.5	73.3±0.7	36.8±1.0	35.7±1.8	84.6±1.1	90.9±1.8	74.5±0.8	49.5±0.9	44.6±2.3
TILE	79.0±2.1	70.0±1.0	50.5±0.5	45.9±1.3	41.1±1.3	65.9±1.3	78.3±0.7	50.2±0.9	49.4±3.8	40.5±0.5
WRAP	88.6±0.3	88.0±0.8	71.6±1.1	47.6±1.0	47.6±1.0	88.6±0.6	88.0±1.7	73.3±0.9	55.0±1.1	38.1±0.7
GRID	84.6±0.4	84.4±0.4	69.7±3.1	45.8±0.4	45.8±0.4	82.2±3.9	89.1±0.4	73.9±0.9	51.6±0.8	38.1±3.1
THEORY	89.2±0.3	90.0±0.5	71.9±0.8	45.0±1.0	45.0±1.0	89.7±0.8	90.9±0.8	73.4±0.8	59.1±0.6	45.5±4.1
Poly2Vec	95.5±0.7	94.9±0.2	81.2±1.0	50.9±0.8	70.2±0.6	95.3±0.3	98.0±0.2	83.0±0.4	64.1±6.2	68.4±0.8
Geo2Vec	**98.5±0.3**	**96.1±0.2**	**96.4±0.5**	**61.2±0.4**	**75.6±0.4**	**98.7±0.4**	**99.1±0.3**	**98.9±0.3**	**67.5±0.8**	**70.0±0.4**

Singapore(Li et al. 2023): This real-world dataset from OpenStreetMap includes 4,347 POIs, 45,634 roads and 109,877 buildings from the region of Singapore.
NYC(Li et al. 2023): Also sourced from OpenStreetMap, this dataset covers New York City and includes 14,943 POIs, 139,512 roads, and 1,153,008 buildings.

Baselines

Four types of baselines are included:
Point encoders: DIRECT (Chu et al. 2019), directly utilizing coordinates; TILE (Berg et al. 2014), partitions the whole area into tiles, and represents with tile embeddings; WARP (Mac Aodha, Cole, and Perona 2019), uses a wrapping mechanism to encode points; GRID (Mai et al. 2023a), multi-scale positional encoding based on Transformer's encoding; THEORY (Mai et al. 2020), encoding with unit vectors separated by 120°.
Polyline encoders: T2VEC (Li et al. 2018), GRU-based autoencoder to learn trajectory representations; T-JEPA (Li et al. 2024), contrastive learning-based trajectory representation learning method.
Polygon encoders: NUFTSPEC (Mai et al. 2023a), encodes polygons through Fourier transform; PolyGNN (Yu et al. 2024a), polygon encoder that encodes polygons and multi-polygons with GNN.
Unified encoder: Poly2Vec (Siampou et al. 2025), decomposes points, polylines, and polygons, and encodes them by geometrically sampling from the Fourier spectral space.

Effectiveness of Shape Representation

We evaluate the effectiveness of our polygon shape representation through two tasks: **shape classification** (Shape) and **predicting the number of edges** (Edge), reporting accuracy and Mean Absolute Error (MAE), respectively. As depicted in Table 1, Geo2Vec significantly outperforms all baselines. PolyGNN, as a GNN-based method, shows poor performance when modeling complex polygons from the MNIST dataset. Moreover, the method relies on contrastive learning and is not able to learn general-purpose polygon representations, limiting its applicability to regression tasks.

To evaluate the model's performance on line entities, we use the learned representations to infer the **length of lines** in two real-world datasets. Results in Table 2 show the superior performance of Geo2Vec (almost 2× improvement over the best baseline). RNN-based approaches like T2Vec and T-JEPA primarily model relationships between individual vertices, overlooking line segments and thereby limiting their ability to represent line entities effectively.

Additionally, we observe that the embeddings learned by Geo2Vec consistently exhibit the lowest standard deviation across nearly all tasks, which holds throughout almost all our experiments. This indicates that the learned embedding space is well-structured and robust.

Effectiveness of Location Representation

To evaluate the effectiveness of location representations generated by different methods, we employ two basic spatial reasoning tasks: **distance estimation** (Table 3) and **topological relationship classification** (Table 4). We report MAE and accuracy for evaluation. To assess the uniformity of the learned representations, we measure their performance when inferring across different types of geo-entities. For brevity, we denote Point as **Pt**, Polyline as **Pl**, and Polygon as **Pg** in the following two tables.

Geo2Vec consistently outperforms all baselines across various distance estimation scenarios. In particular, for complex distance pairs such as polygon-to-polygon and polygon-to-polyline, the performance improvement over the

Table 5: Comparison of spatial representation learning methods on Land Use Classification and Population Prediction tasks.

	Land Use Classification					
	Singapore			NYC		
	L1↓	KL↓	Cosine↑	L1↓	KL↓	Cosine↑
RegionDCL	0.498±0.038	0.294±0.047	0.879±0.021	0.418±0.012	0.229±0.013	0.912±0.006
RegionDCL w/ Poly2Vec	0.484±0.021	**0.278±0.025**	0.881±0.012	0.397±0.010	0.212±0.011	0.923±0.007
RegionDCL w/o Rotation	0.493±0.054	0.309±0.068	0.872±0.028	0.408±0.014	0.226±0.021	0.913±0.008
RegionDCL w/ Geo2Vec	**0.475±0.053**	0.287±0.058	**0.884±0.025**	**0.390±0.013**	**0.208±0.017**	**0.928±0.007**
	Population Prediction					
	Singapore			NYC		
	MAE↓	RMSE↓	R^2 ↑	MAE↓	RMSE↓	R^2 ↑
RegionDCL	5807.54±522.74	7942.74±779.44	0.427±0.108	5020.20±216.63	6960.51±282.35	0.575±0.039
RegionDCL w/ Poly2Vec	4957.58±506.02	6874.47±851.73	0.561±0.117	4602.75±179.66	6393.38±279.70	0.621±0.037
RegionDCL w/ Geo2Vec	**4658.51±483.02**	**6515.26±795.91**	**0.585±0.156**	**4486.49±163.65**	**6189.85+280.05**	**0.625±0.055**

SOTA methods is at least 54.3%.

For topological relationship classification, Pt-Pl, Pt-Pg, and Pl-Pl are binary tasks, while Pl-Pg and Pg-Pg are multiclass. Details can be found in the Appendix. Geo2Vec outperforms both specialized encoders and the unified encoder. The most significant improvement is observed in Pl–Pl relationship inference, where Geo2Vec achieves at least an 18.7% increase in accuracy.

Effectiveness in GeoAI Model

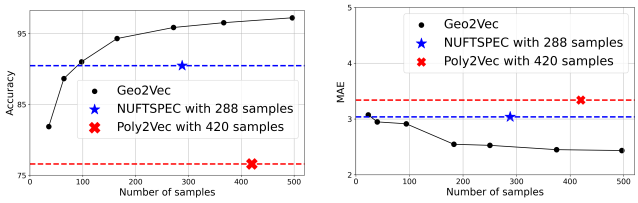
We further evaluate representation effectiveness within existing GeoAI models. This experiment shows the practical effectiveness and real-world potential of Geo2Vec.

Following the experimental setup of previous work (Siampou et al. 2025), we adopt **RegionDCL** (Li et al. 2023) as our GeoAI pipeline model. RegionDCL is designed to learn region-level representations based on the spatial distribution and shape of buildings. The effectiveness of the learned representations is evaluated through two downstream tasks: **Land Use Classification** and **Population Prediction**. In the original setting, each building is rasterized into an image, and its representation is extracted using a Convolutional Neural Network. Since rasterization discards location information, the model incorporates a distance-biased Transformer to reintroduce spatial relationships. In our experiment, we modify this pipeline by replacing it with a standard Transformer network. Instead of using CNN-extracted features, we directly input features obtained from Geo2Vec and Poly2Vec.

In Table 5, **RegionDCL w/o Rotation** refers to the Geo2Vec model without rotation-invariant positional encoding, while **RegionDCL w/ Geo2Vec** represents the full Geo2Vec model. The representations generated by Geo2Vec enable RegionDCL to produce the highest-quality region embeddings. We attribute this improvement to the learning-friendly shape information, and global location information captured by Geo2Vec, which is absent in the raster representation of buildings.[1] The ablation experiment shows that the rotation-invariant property preserved by our positional encoding is beneficial for unsupervised downstream GeoAI model like RegionDCL.

[1]See Appendix for an explanation of the limited improvement, due to RegionDCL's performance ceiling with this input type.

Efficiency Analysis



(a) Classification-Accuracy ↑ (b) Edge-MAE ↓

Figure 3: Comparison between number of sampled points and models' performance on the **Building** dataset.

Previous experiments have showed that Geo2Vec achieves superior embedding quality compared to existing methods under the same embedding dimensionality. We now compare their performance in terms of the number of sample points required. As shown in Figure 3, spectral methods such as NUFTSPEC and Poly2Vec rely on sampling in the Fourier domain, using 288 and 420 points, respectively. However, benefiting from direct access to coordinate space and adaptive sampling, Geo2Vec requires significantly fewer sample points to achieve the same performance. Highlighting the effectiveness of learning geo-entity representations directly from coordinate space rather than relying on less interpretable spectral features.

Conclusion

In this paper, we proposed a unified spatial representation learning method, which is generalizable to all types of geo-entities, including multipolygons and polygons with holes. The learned spatial representation shows superior performance on tasks such as shape classification, distance estimation, and topological relationship classification. Through experiments with an existing GeoAI model, we further show its practicality in real-world scenarios.

To the best of our knowledge, this is the first study to learn geo-entity representations directly from coordinate space,

without relying on decomposition or Fourier transform techniques. Our research reveals the possibility of using neural networks to directly learn both the location and shape representations of geo-entities, and serves as a promising step toward the development of future representation methods for geo-entities.

References

Balsebre, P.; Huang, W.; Cong, G.; and Li, Y. 2024. City Foundation Models for Learning General Purpose Representations from OpenStreetMap. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, 87–97. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704369.

Berg, T.; Liu, J.; Woo Lee, S.; Alexander, M. L.; Jacobs, D. W.; and Belhumeur, P. N. 2014. Birdsnap: Large-scale Fine-grained Visual Categorization of Birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Boo, G.; Darin, E.; Leasure, D. R.; Dooley, C. A.; Chamberlain, H. R.; Lázár, A. N.; Tschirhart, K.; Sinai, C.; Hoff, N. A.; Fuller, T.; Musene, K.; Batumbo, A.; Rimoin, A. W.; and Tatem, A. J. 2022. High-resolution population estimation using household survey data and building footprints. *Nature Communications*, 13(1).

Chu, G.; Potetz, B.; Wang, W.; Howard, A.; Song, Y.; Brucher, F.; Leung, T.; and Adam, H. 2019. Geo-Aware Networks for Fine-Grained Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

Huang, Z.; Khoshelham, K.; Tomko, M.; and xx. 2024. Contrastive Graph Autoencoder for Shape-based Polygon Retrieval from Large Geometry Datasets. *Transactions on Machine Learning Research*.

Ji, Y.; Gao, S.; Nie, Y.; Majić, I.; and Janowicz, K. 2025. Foundation models for geospatial reasoning: assessing the capabilities of large language models in understanding geometries and topological spatial relations. *International Journal of Geographical Information Science*, 0(0): 1–38.

Jiang, J.; Pan, D.; Ren, H.; Jiang, X.; Li, C.; and Wang, J. 2023. Self-supervised Trajectory Representation Learning with Temporal Regularities and Travel Semantics. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 843–855.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, L.; Xue, H.; Song, Y.; and Salim, F. 2024. T-JEPA: A Joint-Embedding Predictive Architecture for Trajectory Similarity Computation. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '24, 569–572. New York, NY, USA: Association for Computing Machinery. ISBN 9798400711077.

Li, X.; Zhao, K.; Cong, G.; Jensen, C. S.; and Wei, W. 2018. Deep Representation Learning for Trajectory Similarity Computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 617–628.

Li, Y.; Huang, W.; Cong, G.; Wang, H.; and Wang, Z. 2023. Urban Region Representation Learning with OpenStreetMap Building Footprints. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, 1363–1373. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701030.

Ma, Z.; Tu, Z.; Chen, X.; Zhang, Y.; Xia, D.; Zhou, G.; Chen, Y.; Zheng, Y.; and Gong, J. 2024. More Than Routing: Joint GPS and Route Modeling for Refine Trajectory Representation Learning. In *Proceedings of the ACM Web Conference 2024*, WWW '24, 3064–3075. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701719.

Mac Aodha, O.; Cole, E.; and Perona, P. 2019. Presence-Only Geographical Priors for Fine-Grained Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Mai, G.; Janowicz, K.; Yan, B.; Zhu, R.; Cai, L.; and Lao, N. 2020. Multi-Scale Representation Learning for Spatial Feature Distributions using Grid Cells. In *The Eighth International Conference on Learning Representations*. openreview.

Mai, G.; Jiang, C.; Sun, W.; Zhu, R.; Xuan, Y.; Cai, L.; Janowicz, K.; Ermon, S.; and Lao, N. 2023a. Towards general-purpose representation learning of polygonal geometries. *GeoInformatica*, 27(2): 289–340.

Mai, G.; Lao, N.; He, Y.; Song, J.; and Ermon, S. 2023b. CSP: Self-Supervised Contrastive Spatial Pre-Training for Geospatial-Visual Representations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 23498–23515. PMLR.

Mai, G.; Yao, X.; Xie, Y.; Rao, J.; Li, H.; Zhu, Q.; Li, Z.; and Lao, N. 2024. SRL: Towards a General-Purpose Framework for Spatial Representation Learning. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '24, 465–468. New York, NY, USA: Association for Computing Machinery. ISBN 9798400711077.

Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1): 99–106.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Siampou, M. D.; Li, J.; Krumm, J.; Shahabi, C.; and Lu, H. 2025. Poly2Vec: Polymorphic Fourier-Based Encoding of Geospatial Objects for GeoAI Applications. In *Forty-second International Conference on Machine Learning*.

Wu, C.; Wang, J.; Wang, M.; Biljecki, F.; and Kraak, M.-J. 2025. Formalising the urban pattern language: A morphological paradigm towards understanding the multi-scalar spatial structure of cities. *Cities*, 161: 105854.

Yan, X.; Ai, T.; Yang, M.; and Tong, X. 2021. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science*, 35(3): 490–512.

Yu, D.; Hu, Y.; Li, Y.; and Zhao, L. 2024a. PolygonGNN: Representation Learning for Polygonal Geometries with Heterogeneous Visibility Graph. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, 4012–4022. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704901.

Yu, M.; Lu, T.; Xu, L.; Jiang, L.; Xiangli, Y.; and Dai, B. 2024b. GSDF: 3DGS Meets SDF for Improved Neural Rendering and Reconstruction. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 129507–129530. Curran Associates, Inc.

Zhao, J.; Chen, C.; Zhu, Y.; Deng, M.; and Liang, Y. 2025. UniTR: A Unified Framework for Joint Representation Learning of Trajectories and Road Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12): 13348–13356.

Appendix

Experiment Setting

For polygon shape learning, we scale each polygon individually to a canonical space $[-1, 1] \times [-1, 1]$ and then learn its shape embedding. For location representation learning, we normalize the entire dataset to a canonical space before applying the learning algorithm. All distance- and length-related results reported in the paper are computed within this canonical coordinate system.

For all the shape representation testing tasks, we split each dataset into training, validation, and test sets using a 70:15:15 ratio. A Multi-Layer Perceptron (MLP) is trained on the 70% training dataset, and then testing and evaluating on the rest two datasets. And for all the experiments, we randomly split the dataset and train the MLP for 5 times and report average performances and standard deviation.

For distance estimation and topological relationship classification, we follow the experiment settings of previous research, details can be found at (Siampou et al. 2025).

To ensure fair comparison across methods, we fix the output dimensionality of all representation learning approaches: 256 dimensions for shape and 256 dimensions for location. Aside from this adjustment, all other parameters follow the original settings specified in the respective papers.

For Geo2Vec, Sample Density ϵ is set as 100 for shape learning and 50000 for location learning, Uniform Sample Density N_{axis} is set as 10 for shape learning and 30 for location learning, Batch Size b is set as 1024×20. We set the frequency range as $L_{\min} = \log_2\left(\frac{1}{\pi}\right), L_{\max} = \log_2\left(\frac{6}{\pi}\right)$ for location representation, and $L_{\min} = 0, L_{\max} = 8$ for shape representation. For shape representation learning, we set $L = 8$, for location representation leaning, we set $L = 16$, meaning we uniformly sample 8 and 16 frequency bands on a logarithmic scale between $L_{\min}$ and $L_{\max}$.

The experiment is conducted on a cluster node equipped with an 16-core CPU, 64GB of memory, and one 12GB NVIDIA GeForce RTX 3060 GPU.

Topological Relationship Classification

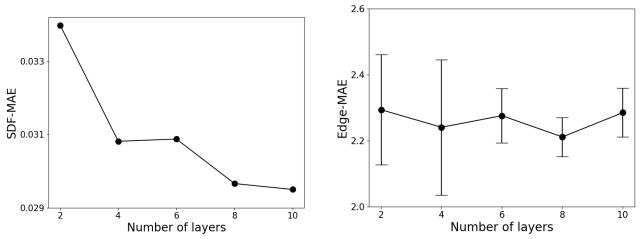
We evaluated the embedding's capability in topological relationship classification, where relationships between geo-entity pairs are defined by the DE-9IM model (Clementini, Di Felice, and van Oosterom 1993). For clarity, we denote Point as **Pt**, Polyline as **Pl**, and Polygon as **Pg** in the following two tables. Details are shown in Table 1.

Table 1: Topological relationships of geo-entity pairs.

Geo-entity Pair	Topological Relationships (a relationship b)
Pt-Pl	disjoint, intersects
Pt-Pg	disjoint, contains
Pl-Pl	disjoint, intersects
Pl-Pg	disjoint, touches, intersects, within
Pg-Pg	disjiont, touches, intersects, contains, within, equals

Parameter Sensitivity

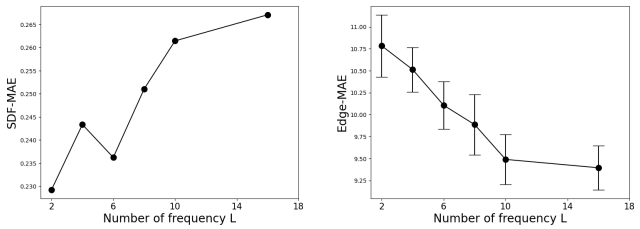
Firstly, we analyze the effect of varying the number of layers. On the **Building** dataset, we evaluate model performance using SDF test MAE and number of edge prediction MAE. Lower test SDF-MAE means model learns SDFs better. As shown in Figure 1, increasing the number of layers improves the model's ability to learn the SDF, resulting in lower SDF prediction error. This, in turn, enhances the quality of the learned shape features, leading to more accurate number of edge predictions and reduced variance. To balance computational efficiency and performance, we chose an 8-layer architecture as the default.



(a) Test SDF-MAE ↓ (b) Edge-MAE ↓

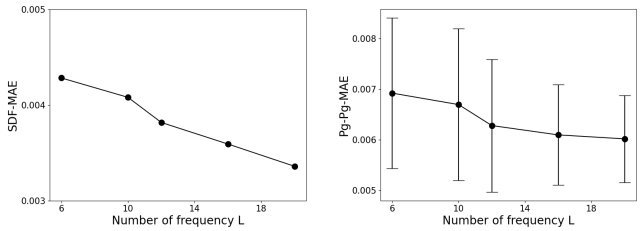
Figure 1: Effect of number of layers in Geo2Vec to model's performance on the **Building** dataset.

Besides the number of layers, we also evaluated the model's performance under different settings of L, the number of sampling frequencies used in the Positional Encoding function. To investigate how positional encoding facilitates learning fine-grained characteristics of geo-entities, we conducted experiments on the **MNIST** dataset, which contains more complex polygonal shapes than the other datasets. The results are shown in Figure 2. As the number of frequencies increases, the period of the positional encoding function decreases, producing higher-frequency features that better capture fine-scale variations in the SDF. However, these strong high frequency signals also reduce the model's ability to learn more generalizable patterns. This explains why increasing the number of frequencies improves performance on edge count prediction, but degrades the quality of SDF simulation. To balance the capture of fine-grained features and the overall quality of the learned SDF, we set $L = 8$ as the default configuration in our experiments.



(a) Test SDF-MAE ↓ (b) Edge-MAE ↓

Figure 2: Effect of number of number of frequencies L in Positional Encoding of Geo2Vec on the **MNIST** dataset.



(a) Test SDF-MAE ↓ (b) Distance estimation MAE ↓

Figure 3: Effect of number of number of frequencies L in Positional Encoding of Geo2Vec on the location learning capability on the **Building** dataset.

We analyze the effect of positional encoding on location representation learning. As shown in Figure 3, experiments on the **Building** dataset demonstrate that increasing the number of frequencies L in the positional encoding improves the model's ability to capture the overall variation patterns of the SDF. This is further supported by improved performance on the distance estimation task, indicating that better learning of SDFs enhances the model's ability to represent location information and thus benefits downstream tasks. We set $L = 16$ in our experiment.

Visualization of SDFs

In Figure 4, we visualize several representative samples of the learned SDFs and their corresponding ground truth for both digits and buildings. As shown in the figure, Geo2Vec effectively captures fine-grained details of complex real-world polygons from **Singapore** dataset, such as building footprints, and also models smooth shape variations, particularly for curved structures like the digits in the **MNIST** dataset.

Performance Discussion

In our experiment, Geo2Vec shows superior performance over SOTA on all spatial reasoning tasks, while the improvement on the RegionDCL is marginal. This is likely because RegionDCL is already near its performance ceiling. RegionDCL is an unsupervised model and is not learning end-to-end, which means, it cannot fully exploit the richer features Geo2Vec provides. Additionally, relying solely on building data limits performance in tasks like land use classification and population prediction.

References

Clementini, E.; Di Felice, P.; and van Oosterom, P. 1993. A small set of formal topological relationships suitable for end-user interaction. In Abel, D.; and Chin Ooi, B., eds., *Advances in Spatial Databases*, 277–295. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-47765-5.

Siampou, M. D.; Li, J.; Krumm, J.; Shahabi, C.; and Lu, H. 2025. Poly2Vec: Polymorphic Fourier-Based Encoding of Geospatial Objects for GeoAI Applications. In *Forty-second International Conference on Machine Learning*.

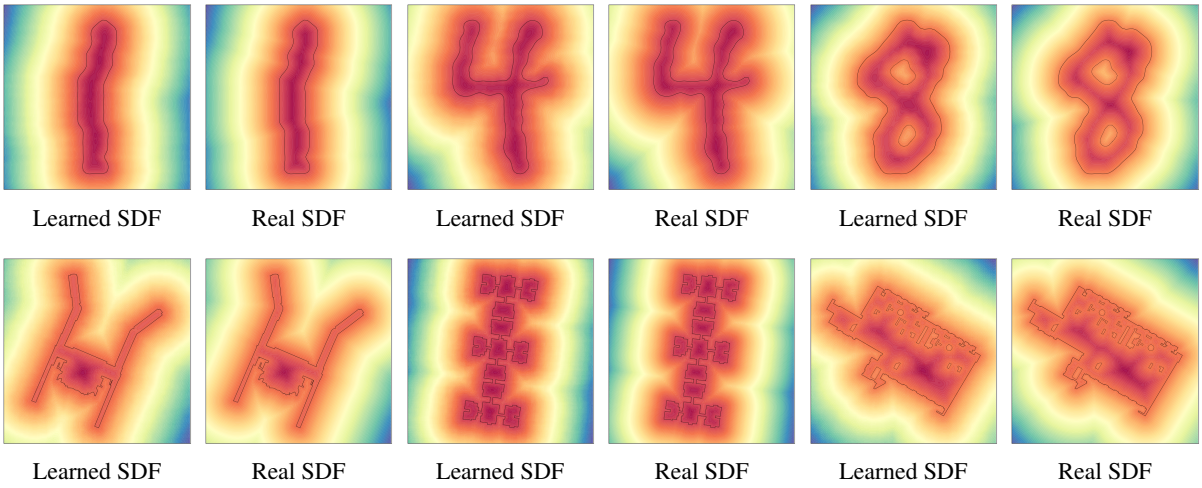


Figure 4: Visualization of the learned SDFs by Geo2Vec and the ground truth. The top row shows digits from the **MNIST** dataset, while the bottom row displays real-world buildings from the **Singapore** dataset.