

Article

Learning Deformation-Induced Shape Representations for Building Footprints via Self-Supervision

Shuqi Cao  and Guohua Ji * 

School of Architecture and Urban Planning, Nanjing University, Nanjing 210093, China;
shuqicao@smail.nju.edu.cn

* Correspondence: jgh@nju.edu.cn

Abstract

Effective shape representation of building footprints is essential for many geospatial analysis and applications, such as building retrieval, cartographic generalization, and urban morphology analysis. Constrained by scarce and coarse annotations of building shapes, self-supervised learning (SSL) is regarded as a promising paradigm. However, existing SSL approaches primarily infer invariance from augmented views of the same instance without explicitly defined similarity, limiting their ability to capture multi-layered relationships—from geometric regularity to structural layout—across building shapes. Providing reliable and interpretable similarity supervision remains challenging. In this work, we propose Deformation-Induced Self-Supervised Learning (DI-SSL), a framework that explicitly defines similarity through Geometric-Aware Deformations (GAD). GAD constrains deformations along global topology, geometric variation and structural layout, jointly characterizing structural comparability between shapes and ensuring geometrically valid, structurally coherent variants under controlled form deviations. Consequently, similarity is explicitly defined through these constrained deformation processes, yielding an embedding space where structurally similar shapes are consistently organized. Experiments demonstrate that DI-SSL achieves strong performance in retrieval and few-shot generalization. The learned embedding space is well-structured, exhibiting global separability of structural prototypes and local continuity consistent with morphologically meaningful similarities. Ablation studies further highlight the critical role of deformation-induced supervision in shaping this space.

Keywords: building footprint analysis; urban morphology; shape representation learning; self-supervised learning; shape similarity measurement



Academic Editor: Wolfgang Kainz

Received: 14 June 2026

Revised: 27 August 2026

Accepted: 10 September 2026

Published: 14 September 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the

International Society for

Photogrammetry and Remote Sensing.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

The shapes of the building footprint are fundamental elements in urban morphology analysis, spatial data mining, and large-scale built environment studies [1–3]. With the increasing availability of urban spatial data, the effective representation of building shapes has become an important problem in geospatial analysis and computational urban studies [4–7]. Recent advances in deep learning have enabled data-driven approaches to shape representation learning, while self-supervised learning (SSL) [8–11] has emerged as a promising paradigm for learning representations without manual labels.

However, multi-layered similarity among building shapes challenges the assumptions underlying most existing SSL frameworks. Beyond instance-level variations caused by resolution or data sources, building shapes exhibit structured relationships ranging from

shared prototypes (e.g., I-shaped, L-shaped, or C-shaped) [12,13] to fine-grained similarities between distinct instances. Capturing these complex relationships is essential for addressing various spatial analysis tasks. In contrast, existing SSL frameworks rely on instance-level invariance, treating augmented views of the same sample as equivalent while implicitly considering all other samples as equally dissimilar. This binary formulation is insufficient for capturing the composite similarity structures inherent in building shapes. This raises a central question: how can representation learning explicitly model structural similarity beyond instance equivalence?

To address this limitation, we propose Deformation-Induced Self-Supervised Learning (DI-SSL), a framework that explicitly constructs similarity relationships. Instead of relying on instance identity or predefined categories, DI-SSL defines similarity as a structured and controllable relation. Specifically, it employs Geometry-Aware Deformations (GAD) as a mechanism to generate structure-preserving shape variants, simulating structural similarity under explicit geometric and topological constraints. These constraints ensure that the constructed relations reflect meaningful morphological variations rather than arbitrary perturbations. Within this framework, similarity is explicitly defined through constrained deformation processes, inducing structured supervision for contrastive learning. As a result, DI-SSL learns embedding spaces where structural similarities are coherently organized while dissimilar shapes remain separable, enabling the model to capture both local structural variations and global relationships among building shapes.

Experiments demonstrate that DI-SSL achieves superior performance in both fine-grained retrieval and few-shot generalization, indicating coherence across both instance-level and cross-instance variations. Structural analyses reveal that the embedding space is not only separable but also locally continuous under structure-preserving variations observed in real-world data. Rather than approximating predefined geometric metrics (e.g., turning function distance), the learned representation reorganizes similarity relationships, where neighborhood structures align with morphologically meaningful variations instead of boundary-level resemblance. Ablation studies further show that this behavior arises from enforcing consistency under geometrically valid deformations, rather than from instance-level invariance.

These results indicate that DI-SSL learns a representation by explicitly constructing shape similarity, inducing a structured and transferable embedding space by organizing building shapes according to geometry-consistent relationships, thereby providing a consistent basis for retrieval, clustering, and large-scale morphological analysis.

2. Related Work

Early studies on architectural and urban form analysis rely on hand-crafted geometric descriptors and predefined similarity metrics. Building footprints are encoded using shape indices [14–16], Fourier descriptors [17], or other compact representations, with similarity quantified through measures such as turning function distances [18], skeleton-based similarity [19], or graph edit distances [20]. While effective in controlled settings, these approaches prescribe similarity a priori, limiting their ability to represent the structured and continuous relationships observed in real-world data.

Recent work has explored neural encoders for polygonal geometry, including rasterization-based models, graph neural networks [21], sequence-based architectures [22], spectral representations [23,24], and signed distance field-based models [25]. These methods focus on encoding geometry, while similarity is implicitly determined by the learning objective. In many cases, formulations based on classification or proxy tasks assume discrete categories, providing limited control over the similarity relationships encoded in the embedding space and failing to explicitly capture continuous geometric variation.

Self-supervised learning enables representation learning without manual annotations. Reconstruction-based approaches [13,26] enforce fidelity to input geometry, while contrastive approaches [27] enforce consistency across transformed views. However, the transformations used to generate such views implicitly define which samples are treated as equivalent, thereby inducing similarity without explicitly modeling its structure. For building footprints, generic augmentations may not reflect morphologically meaningful relationships.

Transformations have been used to model shape variation and learn representations [28,29]. Prior work models deformation itself, whereas DI-SSL uses constrained deformations to explicitly define similarity. For building footprints, this requires geometric and topological constraints to ensure that induced relationships remain valid and structurally meaningful.

Taken together, existing approaches either prescribe similarity through predefined metrics or rely on objectives that implicitly define similarity with limited control. Few methods explicitly construct similarity relationships as part of representation learning. This motivates a framework that defines similarity in a structured and controllable manner, enabling the modeling of fine-grained and continuous relationships between building shapes beyond instance identity or predefined distances.

3. Method

The complex similarity relationships among building footprints require shape representation learning to be guided by well-defined supervisory signals. In this work, the GAD framework (Section 3.2) provides a mechanism to define similarity explicitly via a constrained deformation process, enabling the DI-SSL framework (Section 3.3) to learn a representation space by aligning with that similarity.

3.1. Problem Formulation and Framework Overview

Let $x \in \mathcal{S}$ denote a building footprint represented as a planar polygon. Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, the goal is to learn an embedding function $f_\theta : \mathcal{S} \rightarrow \mathbb{R}^d$ that preserves meaningful structural relationships between shapes. We define a set of admissible deformations $\mathcal{T}_{\text{adm}}$ on $\mathcal{S}$ under explicit geometric and topological constraints. Each $T \in \mathcal{T}_{\text{adm}}$ produces a valid instance $x' = T(x)$, and the set $\mathcal{T}_{\text{adm}}$ specifies which shape variations are considered structurally similar. The embedding function f_θ is trained such that instances related through $\mathcal{T}_{\text{adm}}$ are mapped to nearby representations, while unrelated shapes remain separable. In this way, similarity is explicitly constructed through constrained deformations, rather than implicitly defined by instance identity or predefined metrics.

In practice, this is implemented within a contrastive learning framework (Figure 1), where deformed instances of the same shape form positive pairs and others serve as negatives. As a result, the learned embedding space is structured such that shapes exhibiting similar admissible deformation patterns are consistently grouped into coherent clusters, providing an explicit organization of structural similarity.

3.2. Geometry-Aware Deformations Framework

The admissible deformations set $\mathcal{T}_{\text{adm}}$ is operationalized as a GAD framework that restricts the deformation space to a set of controlled GAD, whose effects are well-defined and interpretable.

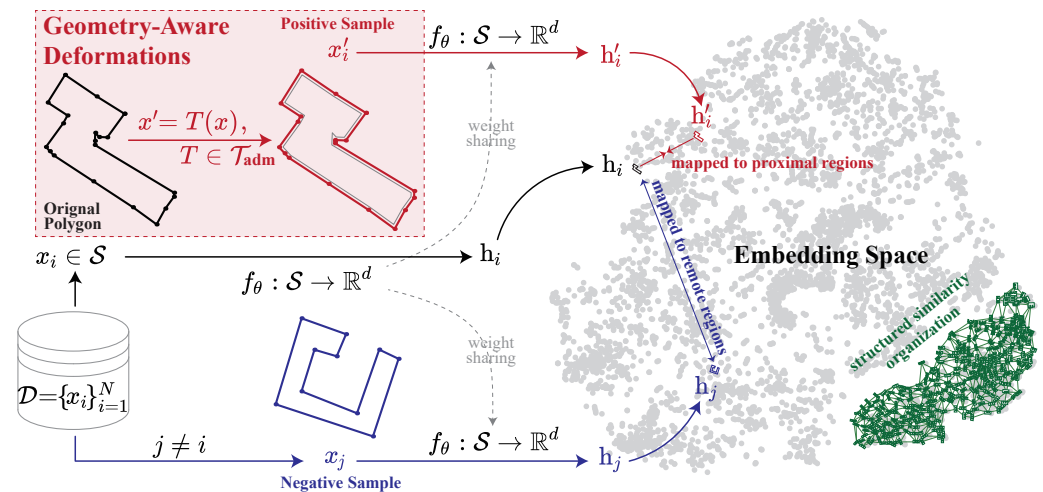


Figure 1. Overview of the proposed framework. Building footprints are deformed under a constrained deformation process to construct similarity relationships. The encoder f_θ maps shapes into an embedding space where these relationships are enforced via contrastive learning, resulting in a structured similarity organization. In the embedding-space panel, gray dots schematically denote the overall embedding distribution, while the green cluster highlights a representative region organized by structured similarity; the red and blue arrows indicate mappings of positive and negative samples, respectively.

3.2.1. Geometry-Aware Deformations

In this work, we primarily instantiate GAD using straight-skeleton-based polygon offsetting (SS-Offsetting), defined as

$$x' = \text{Offset}_{\text{SS}}(x, \delta), \quad |\delta| \leq \alpha \cdot r_{\text{in}}(x), \quad \alpha \in (0, 1) \quad (1)$$

where SS denotes the straight skeleton of polygon x , and δ is the signed offset distance. Positive values of δ produce outward expansion, while negative values yield inward contraction. The offset magnitude is normalized by the inradius $r_{\text{in}}(x)$, corresponding to the maximum admissible interior offset under straight-skeleton propagation, ensuring scale-adaptive and shape-consistent deformation across polygons with varying sizes and geometric complexity.

As shown in Figure 2, the straight skeleton defines an intrinsic geometric structure that governs the evolution of edges and vertices under uniform offsets. Within the admissible range, SS-Offsetting produces shape variants that maintain geometric and topological consistency, including valid boundary configurations (no self-intersection), consistent orientation, and largely preserved combinatorial structure. These properties ensure that the resulting deformations remain within a structurally coherent regime of the shape space. SS-Offsetting can, in principle, be extended to polygons with interior holes, where both exterior and interior boundaries participate in the offsetting process. The presence of multiple boundary components increases the geometric complexity of the straight-skeleton computation and may introduce greater numerical sensitivity in degenerate configurations, such as extremely acute angles or narrow gaps. In the current implementation, numerical instabilities associated with extremely acute angles are mitigated through a preprocessing step that regularizes such configurations by local corner flattening. A systematic evaluation of multiply connected footprints is beyond the scope of this study and is left for future work.

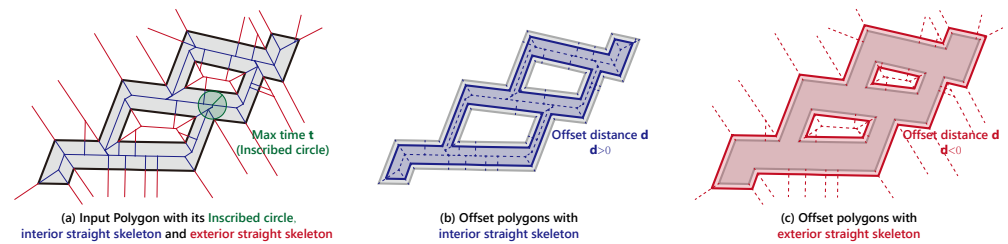


Figure 2. Straight-Skeleton-Based Polygon Offsetting. (a) An input polygon (gray) with its inscribed circle and interior (blue)/exterior (red) straight skeletons. (b) Positive offsetting $d > 0$ generated using the interior straight skeleton, producing an inward-offset polygon. (c) Negative offsetting $d < 0$ generated using the exterior straight skeleton, producing an outward-offset polygon.

Among several possible GADs, we adopt SS-Offsetting as the main deformation because it is intrinsically coupled to polygonal structure and can induce controlled, structure-preserving variation. More importantly, in DI-SSL, the choice of GAD is not merely an augmentation strategy: it explicitly defines the structural similarity prior through the positive pairs used for contrastive learning. A suitable GAD should therefore produce variations that are morphologically interpretable and preserve the structural organization that the representation is expected to regard as invariant. SS-Offsetting satisfies these requirements by providing an edge-driven deformation mechanism with an explicit geometric interpretation and a bounded, scale-adaptive deformation range. Moreover, it aligns with the architectural notion of uniform wall offsetting, providing a morphologically interpretable mechanism for generating layout variations while maintaining structural organization. Offset polygons generated in this manner are therefore treated as geometry-aware variants of the original footprint and serve as the primary source of positive pairs in the DI-SSL framework.

3.2.2. Geometrically Invariant Transformations

Beyond GAD, we employ Geometrically Invariant Transformations (GIT) to handle the inherent variability in vector representations. Specifically, GITs encompass rigid transformations, reflections, uniform scaling, and re-parameterizations of the contour (e.g., resampling and vertex reindexing). As depicted in Figure 3, these operations modify explicit coordinate values while preserving the intrinsic geometry.

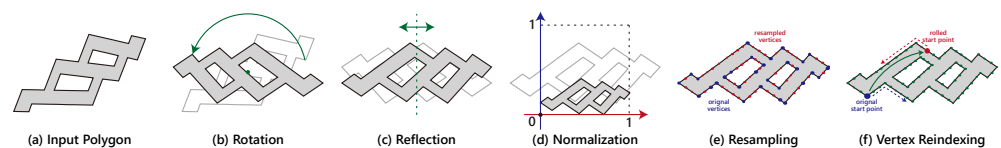


Figure 3. Geometrically Invariant Transformations (GIT). (a) An input polygon. (b) Rigid rotation relative to the origin of the coordinate system (the green point). (c) Reflection transformation with respect to a symmetry axis (the green dashed line). (d) Uniform scaling and normalization that map polygons into a canonical coordinate range. (e) Contour resampling, where blue and red vertices denote the original and resampled vertices, respectively. (f) Vertex reindexing, where the blue and red markers indicate the original and rolled start points, respectively. The operations in (e,f) are re-parameterizations that alter the vector representation without changing the underlying geometry.

The fundamental distinction between GIT and GAD lies in their role in the shape space $\mathcal{S}$: GITs define equivalence transformations that preserve geometric identity, whereas GADs induce controlled deformations that traverse the shape space beyond equivalence instances. GITs primarily enforce invariance at the instance level, encouraging the encoder to learn representations that are stable under geometric equivalence transformations such as rotation and scaling. However, such organization remains confined to enforcing invariance

within individual shape instances. In contrast, GADs serve as the primary driver of the framework, as they introduce structured variations that go beyond invariance, enabling the model to capture geometric regularities and structural layout similarities between shapes.

3.2.3. Validity Constraints for Admissible Deformations

To ensure effective similarity supervision, we employ validity constraints to guarantee geometric feasibility and control form deviations. This work focuses on validity constraints associated with SS-Offsetting.

Topology Preservation: The deformation must preserve the topological structure of the input footprint. In particular, the number of connected components, the number of interior holes, and their containment relations must remain invariant, i.e., $\text{Top}(x') = \text{Top}(x)$. This constraint is enforced by explicit topology checks after offsetting, ensuring that no spurious self-intersections or invalid polygon configurations are introduced.

Bounded Geometric Deviation: To prevent degenerate or excessively distorted shapes, the deformation magnitude is bounded. Specifically, the offset parameter δ is constrained by a shape-dependent upper bound $\alpha \cdot r_{\text{in}}(x)$ (Section 3.2.1), which limits the extent of geometric variation induced by the deformation.

Shape Similarity Constraint: To further ensure that admissible deformations remain within a geometrically coherent regime, we impose an additional shape similarity constraint based on the Turning Function (TF) representation. Specifically, the deviation between the original shape x and its transformed counterpart x' is required to satisfy $d_{\text{TF}}(x, x') \leq \tau_{\text{TF}}$, where $d_{\text{TF}}(\cdot, \cdot)$ denotes the TF distance and τ_{TF} is a predefined threshold. This constraint acts as a validity filter that excludes extreme or structurally implausible deformations that may still satisfy topological and parametric constraints but deviate significantly in global shape characteristics. Importantly, the TF is not used as an optimization objective or supervision signal, but solely as a criterion for restricting the admissible deformation space.

Structural Consistency: Structural consistency is implicitly guaranteed by the SS-Offsetting operation, which preserves combinatorial properties such as edge adjacency and vertex ordering. As a result, admissible deformations maintain the overall structural organization of the footprint without requiring additional explicit constraints.

Together, topology preservation enforces global structural correctness, bounded deformation controls geometric variation, and structural consistency maintains combinatorial coherence. These constraints jointly define admissibility at complementary levels, which is the admissible deformation space underlying the proposed framework.

3.2.4. Transformation Pipeline

We integrate GAD and GIT into a unified transformation pipeline, designed to explicitly control the formation of positive pairs in DI-SSL. This pipeline enforces a structured separation between similarity and invariance, enabling the learned embedding space to reflect well-defined modes of geometric variation.

Given an input polygon, GAD is applied via SS-Offsetting by randomly sampling a signed offset distance δ within the predefined range. The resulting polygon is evaluated using the validity constraints in Section 3.2.3. This procedure is repeated for a limited number of attempts; if no admissible deformation is obtained, the original polygon is retained.

Following GAD, GIT is applied to remove nuisance variations and standardize representation. Specifically, rigid transformations and reflections are first applied to eliminate orientation dependence. The polygon is then uniformly scaled to a unit bounding box to normalize global size. Subsequently, it is resampled into a fixed-length vertex sequence to ensure consistent input dimensionality. Finally, vertex reindexing is performed to eliminate biases induced by vertex ordering.

While many commonly used polygon augmentations (e.g., anisotropic scaling, elastic deformations or path perturbation) lie on a spectrum between purely GIT and GAD, we deliberately restrict the transformation set to a single, controlled deformation. This design isolates a coherent mode of structural variation, allowing the organization of the embedding space to be directly attributed to structural similarity under well-defined geometric constraints, rather than to a mixture of heterogeneous transformation effects.

This pipeline induces a clear functional decomposition: GAD defines the positive relation between instances by introducing admissible geometric variation, while GIT enforces invariance within each instance by removing irrelevant degrees of freedom. Their combination shapes an embedding space in which structural similarity is encoded through controlled deformation, rather than through heuristic augmentation diversity.

3.3. Deformation-Induced Self-Supervised Learning

Building upon the deformation generation mechanism introduced in Section 3.2, we now specify how these deformations are translated into a learning signal. Rather than treating transformations as mere data augmentation, DI-SSL formulates them as structured relational supervision that explicitly encodes shape similarity under geometry-preserving deformations.

3.3.1. DI-SSL Paradigm

This section formalizes how deformation-induced geometric relations explicitly define similarity and how this definition is translated into an optimization objective. Instead of relying on predefined similarity metrics or semantic labels, the framework uses admissible GAD to construct a set of relations, specifying which variations of a polygon should be regarded as similar. Contrastive learning is then employed as a mechanism to enforce these predefined relations during representation learning, rather than to infer similarity implicitly.

Deformation-Induced Positive Construction: For each polygon x_i , a positive counterpart $x'_i = T(x_i)$ is generated via an admissible GAD $T \in \mathcal{T}_{\text{adm}}$, as introduced in Section 3.2. These deformations preserve structural validity while introducing controlled geometric variation. As such, the pair (x_i, x'_i) explicitly defines a notion of similarity grounded in deformation, where admissible deformations determine which geometric variations should be treated as similar instances of the same underlying structure.

Negative Sample Construction: Negative samples are formed by pairing x_i with other polygons x_j randomly drawn from the dataset ($j \neq i$). No explicit geometric assumptions are imposed on negative pairs. Instead, dissimilarity is defined implicitly as the complement of the deformation-induced similarity, allowing the model to separate shapes that are not connected through admissible deformations without introducing additional bias.

Contrastive Supervision Signal: Given the above construction, contrastive learning serves to enforce the explicitly defined similarity relations. Embeddings of x_i and x'_i are encouraged to be close, while embeddings of unrelated polygons are separated. In this formulation, the contrastive objective does not determine similarity but operationalizes the deformation-defined similarity into a learnable objective, ensuring that the resulting embedding space faithfully reflects the predefined geometric relations.

3.3.2. Polygon Representation Encoder

For DI-SSL, the encoder f_θ maps each polygon $x_i \in \mathcal{D}$ to a fixed-dimensional latent representation $h_i \in \mathbb{R}^d$ to capture its geometric configuration. The encoder is treated as a modular component rather than a central contribution of this work. It does not encode building semantics, perform classification, or reconstruct polygons. Instead, its role is to provide a generic representation space upon which deformation-induced relations

can be imposed. In principle, this formulation is compatible with a range of architectures capable of processing polygonal geometry (e.g., sequence-based, spectral-based, or graph-based encoders).

In this work, we instantiate f_θ using a sequence-based encoder, ResNet1D, primarily due to its building maturity and stable empirical behavior, rather than any task-specific design. This choice serves as a neutral and reliable backbone for evaluating the DI-SSL framework. Importantly, the independence from a specific encoder arises from the separation between representation and relational supervision: the deformation strategy defines the structural similarity to be learned, while the encoder determines how this structure is embedded. Empirical results across alternative encoder choices are provided in Appendix A.1 as supporting evidence.

Consequently, the applicability of DI-SSL does not rely on a particular encoder architecture, although its effectiveness remains dependent on the encoder's capacity to preserve geometric structure.

3.3.3. MLP Projection Head

To facilitate contrastive optimization, we introduce a projection head g_ϕ , implemented as a multi-layer perceptron (MLP), which maps the encoder output $h = f_\theta(x)$ to a latent vector $z = g_\phi(h)$. The contrastive loss is applied in this projection space rather than directly on h .

This design decouples the optimization space from the representation space. In particular, contrastive objectives tend to impose geometric constraints (e.g., uniformity on the hypersphere) that may distort the intrinsic structure induced by deformation-based relations. By introducing a projection head, such objective-specific effects are absorbed in z , allowing the encoder f_θ to focus on learning deformation-induced representations.

3.3.4. Learning Objective

The learning objective instantiates the deformation-induced relational constraints defined in Section 3.3 into a contrastive optimization problem.

Each footprint polygon x_i is first encoded as $h_i = f_\theta(x_i)$ and then projected to $z_i = g_\phi(h_i)$. Its deformed counterpart $x'_i = T_k(x_i)$ is mapped to $z'_i = g_\phi(f_\theta(x'_i))$. These deformation-induced pairs (z_i, z'_i) serve as positive samples, while projected representations of other instances in the dataset act as negatives. To enforce this structure, we adopt the normalized temperature-scaled cross-entropy (NT-Xent) loss [9]:

$$\mathcal{L}_i = -\log \frac{\exp(\text{sim}(z_i, z'_i)/\tau)}{\exp(\text{sim}(z_i, z'_i)/\tau) + \sum_j \exp(\text{sim}(z_i, z_j)/\tau)}, \quad (2)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is a temperature parameter controlling the concentration level of the distribution. Although NT-Xent is used here, the DI-SSL framework is not tied to a specific contrastive formulation.

Optimizing under NT-Xent loss results in an embedding space (i.e., the encoder output h) that is aligned with geometry-preserving deformation relations: shapes connected through admissible deformations are mapped to nearby regions, while structurally unrelated instances are separated. In contrast to conventional contrastive learning setups where invariance is implicitly induced through generic augmentations, DI-SSL explicitly specifies this similarity via GAD, with contrastive optimization serving only to encode them in a fully unsupervised manner.

4. Experiments

4.1. Training Data and Implementation

4.1.1. Training Dataset: Real-World Building Footprints

We construct a large-scale training dataset of building footprint polygons from OpenStreetMap (OSM), covering diverse geographic regions worldwide. To mitigate the influence of degenerate or overly trivial geometric cases that may bias contrastive learning, we apply a set of lightweight filtering criteria to exclude extreme instances. Specifically, we remove (1) overly simplified polygons with very few vertices; (2) highly regular, grid-aligned shapes with limited structural variation; (3) near-circular footprints; (4) extremely small or large buildings; and (5) objects associated with special semantic categories (e.g., transportation infrastructure).

These criteria are designed to reduce the influence of degenerate, overly trivial, or extreme instances and to maintain a diverse and informative training distribution, rather than to enforce any task-specific assumptions. Importantly, these filtering criteria are dataset-specific preprocessing choices and are not required by the DI-SSL framework itself.

We restrict our study to simple polygons without holes to focus on geometric variations while avoiding additional topological complexity. Shape simplification is applied with a tolerance of 0.15 m to remove fine-scale geometric noise. After preprocessing, the dataset contains 12,298,902 building footprints. No manual semantic labels are used during training.

4.1.2. Implementation Details

We employ a 18-layer ResNet1D as the main polygon encoder f_θ to map each input polygon to a latent representation $h \in \mathbb{R}^{512}$. This representation is further projected to $z \in \mathbb{R}^{32}$ via a two-layer MLP projection head g_ϕ for contrastive optimization. Unless otherwise specified, all evaluations and downstream analyses are conducted using the encoder output $h = f_\theta(x)$.

The model is trained using the complete preprocessed training corpus within the DI-SSL framework using the NT-Xent loss. Optimization is performed using AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$) with a base learning rate of 0.001 and weight decay of 0.001. The model is trained for 10 epochs under a linear learning rate schedule with warmup: the learning rate increases linearly from 0 to the peak value during the first epoch, followed by a linear decay to 0 over the remaining epochs. Training was performed on a single NVIDIA RTX A6000 GPU. The observed epoch time varied across runs and training stages (Appendix Table A8), with a typical duration of approximately 8–9 min.

For the transformation pipeline, $\alpha = 0.5$ and $\tau_{TF} = 0.1\pi$ are adopted as the default deformation parameters based on the corresponding sensitivity analyses. The contrastive-loss temperature is set to $\tau = 0.01$ based on the temperature sensitivity analysis. These values are subsequently fixed throughout the main experiments. SS-Offsetting operation, TF-based validity checking and resampling are performed offline during preprocessing and therefore do not contribute to the reported per-epoch training time. The computational efficiency of this preprocessing stage across different α values is further analyzed in Appendix A.5. For the input representation, each vertex is encoded using a multi-scale local geometric descriptor. Specifically, distances are computed to neighboring vertices at offsets defined by powers of two and their complements with respect to the sequence length.

4.2. Empirical Evaluation of Learned Shape Representations

4.2.1. Evaluation Protocol

We evaluate the quality of the learned shape embeddings through shape retrieval and few-shot classification.

Shape Retrieval performance is evaluated using Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), and normalized Discounted Cumulative Gain (nDCG). Retrieval scores are computed across the entire IJGIS-CGAE dataset and averaged over all query instances. Shape Retrieval metrics characterize the local geometric ordering of the embedding space, indicating whether shapes undergoing similar structural variations remain proximal.

Few-shot Classification performance is evaluated using an N -way K -shot episodic protocol. In each episode, K samples are randomly drawn from each of 10 classes to form the support set, and the remaining samples serve as queries. Each query is assigned the label of its nearest neighbor in the support set based on Euclidean distance in the embedding space. This procedure is repeated over 1000 episodes, and the final accuracy is reported as the average classification accuracy across all queries and episodes. We report results for $K \in \{1, 3, 5\}$, enabling evaluation under varying levels of supervision. Few-shot classification metrics reflect the emergent global structure, evaluating whether the embedding space supports the formation of separable and generalizable class structures from sparse labels.

Both tasks are conducted on the IJGIS-CGAE dataset [13], which consists of 10 building footprint classes (ILOHFZYETC), each containing 500 manually curated polygonal examples. While limited in scale and diversity, its curated and low-noise annotation space makes it suitable for assessing whether learned representations exhibit alignment with established building shape semantics. This dataset serves solely as an external evaluation probe and a standardized reference for comparative evaluation, without participating in training or affecting representation learning.

Unless otherwise specified, all the learning-based models are trained in accordance with the training protocol and deformation settings described in Section 4.1.2, but using input representations adapted to the encoder architecture and employing Euclidean distance in the embedding space for evaluation.

4.2.2. Performance Comparison

We compare representative methods from three paradigms in Table 1. All learning-based models are averages over 10 runs with different random seeds. Standard deviations and 95% confidence intervals are given in Table A9 in the Appendix.

Table 1. Quantitative comparison of representative methods from hand-crafted descriptors, autoencoder-based models, and contrastive learning under a unified evaluation protocol. Bold entries indicate the best result for each evaluation metric.

Method	Paradigm	NN	Retrieval			Few-Shot		
			FT	ST	nDCG	1-Shot	3-Shot	5-Shot
FD	Hand-crafted descriptor	0.994	0.732	0.880	0.957	0.771	0.882	0.913
TF	Hand-crafted descriptor	0.997	0.787	0.928	0.972	0.827	0.918	0.943
GCAE *	Autoencoder-based	0.895	0.293	0.412	0.800	0.291	0.422	0.488
NUFT-based AE	Autoencoder-based	0.904	0.220	0.298	0.764	0.210	0.308	0.369
PSRT *	Contrastive learning	0.995	0.770	0.902	0.968	0.811	0.917	0.943
DI-SSL (ours)	Contrastive learning	0.997	0.804	0.946	0.977	0.851	0.941	0.961

* Input Polygons are resampled to 64 vertices to maintain consistency with the original work's configuration. DI-SSL results with 64 vertices are additionally reported in Appendix Table A9, which shows that changing the input resolution from 64 to 128 vertices does not consistently improve performance and does not alter the main conclusions.

Among hand-crafted descriptors, Fourier Descriptor (FD) provides a strong baseline with efficient fixed-length representations, while the Turning Function (TF) consistently outperforms FD across all metrics, particularly in retrieval quality (e.g., FT: 0.787 vs. 0.732,

ST: 0.928 vs. 0.880). This indicates that explicitly designed geometric signatures remain highly effective for capturing fine-grained shape similarity. However, TF relies on pairwise alignment for distance computation and does not yield a parametric embedding space, limiting its applicability to downstream tasks such as clustering or continuous structure analysis. In contrast, FD supports efficient vectorized comparison but exhibits weaker performance, especially in few-shot settings, suggesting limited capacity to encode higher-order structural variation.

Autoencoder-based methods, including Graph Convolutional Autoencoder (GCAE) [13] and Non-Uniform Fourier Transforms based autoencoder (NUFT-based AE) [23], perform substantially worse across all metrics. This suggests that reconstruction objectives at the vertex or spectral level do not necessarily induce a semantically meaningful embedding geometry aligned with similarity, resulting in poor neighborhood structure and weak transferability to downstream tasks.

Among contrastive learning approaches, Pre-trained Shape Representation Transformer (PSRT) [27] achieves strong and balanced performance across both retrieval and few-shot evaluation, confirming the effectiveness of contrastive objectives for learning discriminative representations. Its retrieval metrics approach saturation (e.g., NN = 0.995, nDCG = 0.968), and it maintains high accuracy under limited supervision. However, DI-SSL consistently outperforms PSRT across all metrics (e.g., FT: 0.804 vs. 0.770, ST: 0.946 vs. 0.902, 1-shot: 0.863 vs. 0.811), with a paired bootstrap analysis over 10,000 random retrieval queries confirming the significance ($p < 0.001$ for FT, ST, and nDCG), indicating a more coherent local neighborhood structure. In contrast, PSRT relies on discretized polygon encoding and anisotropic normalization, which implicitly impose a non-uniform similarity prior and may weaken alignment with fine-grained geometric variation.

Overall, the results show that different paradigms induce fundamentally different notions of shape similarity. Hand-crafted methods capture accurate pairwise distances but lack a global embedding structure; autoencoder-based approaches optimize reconstruction rather than structural similarity and therefore may fail to form meaningful similarity neighborhoods; and contrastive methods achieve strong discriminability but remain sensitive to the representation and augmentation strategies used to define positive pairs. In contrast, DI-SSL explicitly defines similarity through geometry-aware, structure-preserving deformations, providing a direct correspondence between the variations treated as similar during training and the morphological variations of interest. This deformation-induced similarity prior yields an embedding that supports both reliable retrieval and robust few-shot generalization. This suggests that the central challenge lies not in separability but in how similarity is defined and structured.

4.3. Structural Analysis of Shape Space

While the results in Section 4.2.2 demonstrate that DI-SSL is consistently competitive across both few-shot classification and retrieval metrics, these performance gains provide only indirect evidence of the underlying representation quality. We proceed to analyze the structural properties of the learned representations in order to more explicitly examine how different methods organize the shape space.

4.3.1. Separability of Benchmark Shape Classes

We begin with a label-based separability analysis as a sanity check to verify whether the learned representation preserves basic class separability on curated benchmarks.

We constructed k-nearest neighbor (kNN) graphs ($k = 50$) from distance matrices computed by four representative methods (FD, TF, PSRT, and DI-SSL) and analyzed the edge confusion matrix (Figure 4a) to quantify inter-class connectivity. Across all four

representations, the inter-class edge ratio remains consistently low, indicating that the IJGIS-GCAE dataset itself exhibits a high degree of intrinsic separability, independent of the specific representation used. Despite this overall consistency, some differences emerge in the structure of inter-class connections among the different representative methods. DI-SSL achieves the lowest inter-class edge ratio with the highest geodesic distance cross-/intra ratio (2.47), indicating a stronger separation between inter-class and intra-class neighborhoods in the induced shape space. Notably, its inter-class edges are concentrated on a smaller number of class pairs compared to other representations, with the vast majority occurring between T-shaped and Y-shaped classes, indicating that inter-class confusion is highly structured rather than uniformly distributed and primarily reflects proximity between geometrically similar classes.

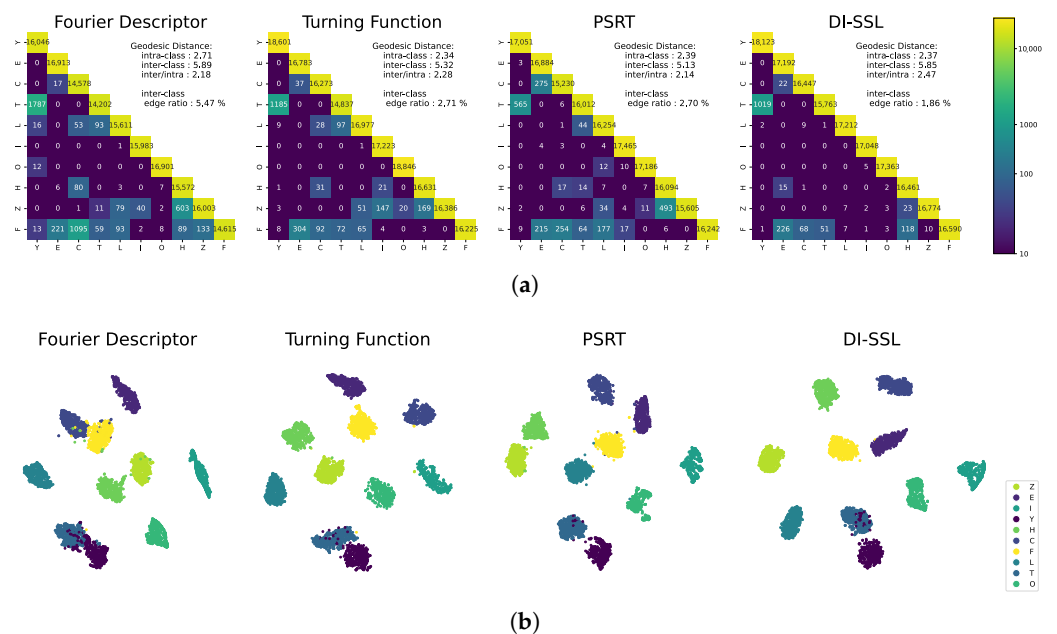


Figure 4. Label-based separability analysis on the IJGIS-GCAE dataset. (a) Edge confusion matrix of kNN-graph (k = 50). (b) t-SNE visualization.

This observation is corroborated by t-SNE visualizations (Figure 4b), where all methods produce compact and well-separated clusters with minimal overlap. However, there is less confusion between different clusters in DI-SSL, with the only consistent ambiguity arising between T-shaped and Y-shaped forms. Rather than indicating a deficiency, this localized overlap reflects genuine structural similarity between classes with closely related branching configurations. The results collectively suggest that DI-SSL not only preserves the global separability of the dataset but also yields a more structured and semantically coherent organization of inter-class relationships.

These results confirm that the benchmark classes are highly separable in the learned embedding space. However, as this dataset is curated with clean class definitions and limited intra-class variation, such separability does not necessarily reflect the intrinsic, deformation-driven organization of real-world shape spaces.

4.3.2. Continuity in Real-World Shape Space

We then move to a continuity analysis to probe the intrinsic geometric organization of the embedding space, examining whether smooth structure-preserving transitions emerge between shapes.

Specifically, we project the original 512-dimensional embeddings into a 32-dimensional space using a Principal Component Analysis (PCA) model fitted on the full training dataset

(real-world data sourced from OSM as described in Section 4.1.1). Linear interpolation is then performed between selected pairs of shapes in the PCA-projected space. Each interpolated point is mapped to its nearest neighbor in the dataset, forming a discrete sequence of observed shapes along the interpolation path.

While this procedure does not recover true continuous trajectories in the original high-dimensional space, the resulting sequences provide insight into the local organization of the embedding. If the embedding space were fragmented or lacked coherent structure, such interpolations would typically produce discontinuous or implausible transitions after projection. Thus, this procedure can serve as an informative test of geometric organization with minimal structural assumptions.

Figure 5 shows five representative examples across distinct structural pairs (IO, YT, CL, ZH, EF), which were manually selected as representative examples to cover different types of morphological similarity observed across the ten shape categories. In these cases, the retrieved sequences exhibit gradual and consistent geometric variations, without abrupt structural discontinuities. The transitions are not limited to trivial blending effects, but instead reflect non-trivial, structure-preserving deformations between the endpoints, with intermediate shapes remaining plausible and geometrically coherent. This deformation-consistent organization provides a structural basis for the empirical performance observed in Section 4.2.2. This behavior is particularly evident for structurally similar class pairs. For example, along the YT trajectory discussed in Section 4.3.1, the transition between the two structures occurs progressively, without a clearly defined point at which one class definitively changes into the other.

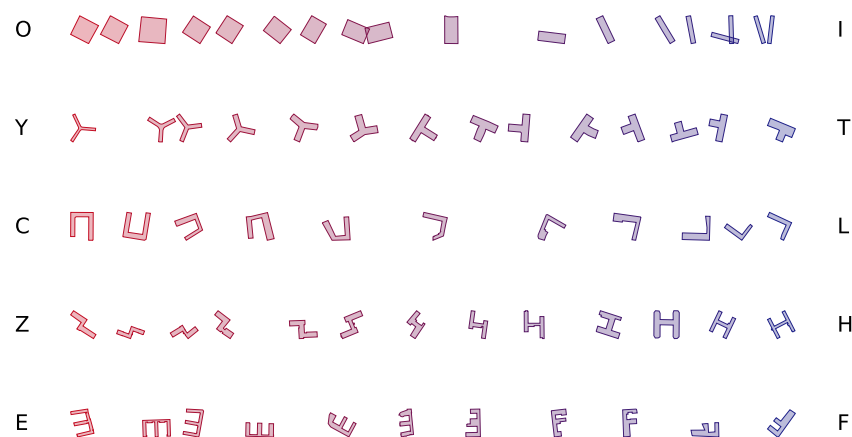


Figure 5. Continuous deformation paths revealed along PCA directions. For each of five inter-class pairs (IO, YT, CL, ZH, EF), we linearly interpolate between their positions in the PCA-projected space and retrieve the nearest neighbors in the original embedding for each interpolated point (duplicates removed). The resulting sequences exhibit consistent, structure-preserving geometric transitions between the endpoint shapes. The spacing between consecutive shapes is approximately proportional to their Euclidean distance in the PCA space.

These observations indicate that the embedding is organized such that its dominant variation directions correspond to coherent, structure-preserving deformations. Rather than merely separating instances, the representation reflects deformation-induced similarity relations, providing empirical evidence that the learned space captures meaningful structural variation in real-world building shapes.

4.4. Relationship to Conventional Shape Metrics

We further investigate the relationship between the learned embedding space and a conventional geometric similarity measure by comparing Euclidean distances in the embedding space (d_{emb}) with the TF distance (d_{TF}).

To quantify the overall relationship between the two metrics, we randomly sampled 5000 shapes from the training dataset and computed all pairwise distances, resulting in 12,497,500 shape pairs. Figure 6 shows the scatter distribution along with a linear regression fit $d_{emb} = \alpha d_{TF} + \beta$. The fitted slope is $\alpha = 224.47$ (95% CI: [224.26, 224.69]) and the intercept is $\beta = 155.37$ (95% CI: [155.18, 155.57]). The slope is statistically different from zero ($t = 2054.39$, $p < 0.001$), whereas the coefficient of determination is $R^2 = 0.2525$.

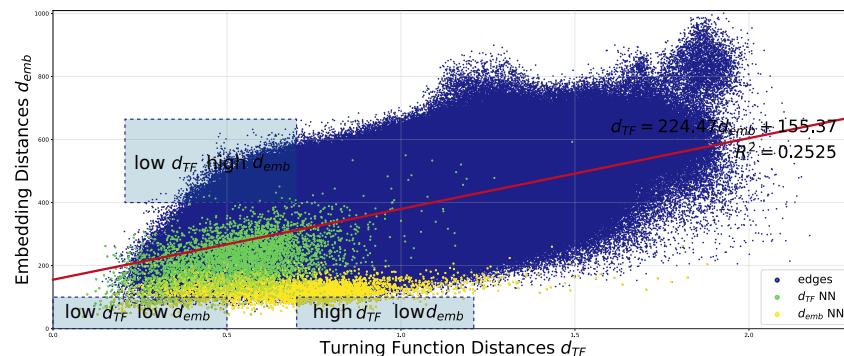


Figure 6. This scatter plot compares pairwise Euclidean distances in the learned embedding space (d_{emb}) with TF distances (d_{TF}) for 5000 randomly sampled shapes from the training dataset, with a linear regression fit. Boxes mark the three regions analyzed in subsequent sections. Green and yellow scatters show NN pair distributions induced by d_{emb} and d_{TF} , respectively.

The statistically significant linear association indicates that the learned embedding distance and the TF distance are related, but the relatively low R^2 indicates that this relationship explains only a limited proportion of the observed variance. Thus, the learned embedding does not simply reproduce the TF metric through a linear or global transformation. More importantly, the scatter distribution exhibits clear structural patterns with systematic deviations from the regression line. Specifically, we observe regions where shapes with large d_{TF} remain close in the embedding space, as well as regions where shapes with small d_{TF} are mapped far apart. This indicates that the discrepancy between the two distance measures is not adequately explained by random variation around a global linear relationship, but reflects a structured reorganization of similarity relationships in the learned embedding space.

To better characterize this structured relationship, we further perform distance-based and neighborhood-based comparisons between the two metrics.

In the low d_{TF} , low d_{emb} region (Figure 7a), shape pairs are nearly identical in both geometric detail and structural organization, indicating agreement between the two metrics. In contrast, in the low d_{TF} high d_{emb} region (Figure 7b), we observe pairs such as rectangle–semicircle and triangle–arch. Although these pairs appear geometrically similar under the TF metric, they are regarded as different prototypes in the architectural context. This structural disparity between shape pairs becomes more pronounced as d_{TF} increases, even when $d_{TF} < 0.7$.



Figure 7. Shape pairs in distinct regions, with $d_{TF} | d_{emb}$ values denoted below each pair. (a) Low d_{TF} low d_{emb} region: shape pairs satisfying $d_{TF} < 0.5, d_{emb} < 100$. (b) Low d_{TF} high d_{emb} region: shape pairs satisfying $d_{TF} < 0.7, d_{emb} > 400$. (c) High d_{TF} low d_{emb} region: shape pairs satisfying $d_{TF} > 0.7, d_{emb} < 100$.

Conversely, in the high d_{TF} , low d_{emb} region (Figure 7c), we observe shape pairs that remain close in the embedding despite large contour differences. These pairs typically share

a similar medial-axis structure (consistent with the GAD definition), while differing in boundary-level details. This indicates that the learned embedding is relatively insensitive to contour variations that preserve the underlying structural skeleton. In contrast, a small number of cases correspond to shapes with weak or ambiguous skeletal structure, where such consistency is less clearly defined.

In addition, Figure 6 also reveals systematic differences in the nearest-neighbor (NN) pair distributions induced by the two metrics. Specifically, NN pairs derived from d_{emb} form a more localized and compact distribution in the joint space, whereas NN pairs derived from d_{TF} are more dispersed. This contrast indicates that the two metrics induce substantially different neighborhood structures. Detailed instance-level comparisons are provided in Appendix A.4 as supporting evidence.

Overall, the results indicate that DI-SSL does not approximate geometric distance, but instead reorganizes shape similarity through deformation-induced structural relations. Although the TF distance is used as a validity constraint when constructing GAD samples, it is not the target distance or optimization objective of the representation learning process. Rather, the straight-skeleton offsetting process determines how morphological variations are generated and, consequently, which structural variations the model is exposed to during self-supervised learning. Therefore, some agreement between d_{emb} and d_{TF} is expected, but the observed discrepancies in both global distance distributions and local neighborhoods indicate that the learned representation is not simply reproducing the TF metric. This distinction reflects the methodological coupling between the TF-based deformation constraint and the learned representation.

4.5. Learning Similarities Through Deformation

To investigate how geometric deformations influence the learned representations, we analyze the impact of different deformation strategies used to construct positive pairs. In particular, we compare GIT (Section 3.2.2) with GAD (Section 3.2.1), and further examine the role of TF-based validity constraints in controlling deformation strength. The results are summarized in Table 2.

Table 2. The effect of different deformation strategies and different threshold of TF-based validity constraints. Bold entries indicate the best result for each evaluation metric.

TF Threshold	Strategies	NN	Retrieval			Few-Shot		
			FT	ST	nDCG	1-Shot	3-Shot	5-Shot
0.0π	GIT	0.992	0.763	0.901	0.965	0.802	0.900	0.926
0.05π	GAD	0.996	0.793	0.924	0.972	0.831	0.924	0.947
0.1π (default)	GAD	0.997	0.804	0.946	0.977	0.851	0.941	0.961
0.2π	GAD	0.998	0.795	0.928	0.976	0.843	0.949	0.970
1.0π (w/o constraint)	GAD	0.996	0.709	0.850	0.956	0.750	0.894	0.932

We first analyze the effect of different deformation strategies. GIT (0.0π) represents a highly constrained setting, where transformations only modify explicit coordinate values while preserving the intrinsic geometry. While this preserves strict structural consistency, it restricts the diversity of positive pairs, resulting in suboptimal representation quality (e.g., 1-shot: 0.802, 5-shot: 0.926). In contrast, GAD enables controlled geometric perturbations that introduce meaningful intra-class variability. As observed at the default setting (0.1π), this leads to substantial improvements across both retrieval (FT: 0.804, ST: 0.946) and few-shot metrics (1-shot: 0.851, 5-shot: 0.961). This indicates that moderate, structure-preserving deformations are essential for learning consistent yet discriminative embeddings.

To further examine the role of validity constraints, we vary the TF threshold controlling the maximum allowable deformation. As the threshold increases from 0.0π to 0.1π , perfor-

mance improves consistently, suggesting that increasing deformation diversity enhances representation robustness, as long as structural validity is preserved. Further increasing the threshold to 0.2π maintains comparable few-shot performance, but slightly degrades retrieval consistency (FT: 0.795, ST: 0.928). This reflects a trade-off between representation diversity for generalization and the preservation of fine-grained local neighborhood structure. When the constraint is removed, performance degrades significantly across all metrics (e.g., FT: 0.709, ST: 0.850, 1-shot: 0.750), indicating that excessive deformation introduces structural inconsistency and corrupts the learning signal.

Figure 8 provides a visual interpretation of the effects observed in the quantitative results, illustrating how different deformation strategies influence the structural validity of generated shapes. Figure 8a shows the original footprint. GIT (Figure 8b) introduces only limited geometric variation, resulting in positive pairs that are nearly identical to the original footprint. While such transformations preserve strict structural equivalence, they fail to expose meaningful intra-class variability. In contrast, unconstrained GAD (Figure 8c) produces highly diverse shapes, but often violates structural plausibility, leading to unrealistic or semantically inconsistent geometries. These distortions weaken the correspondence between positive pairs and introduce noise into the contrastive objective. When validity constraints are applied (Figure 8d), the generated shapes achieve a balance between diversity and structural consistency. The deformations remain within a plausible geometric regime, preserving the overall shape identity while introducing non-trivial variations. As a result, positive pairs become both structurally consistent and geometrically informative, providing stable supervision for contrastive learning.

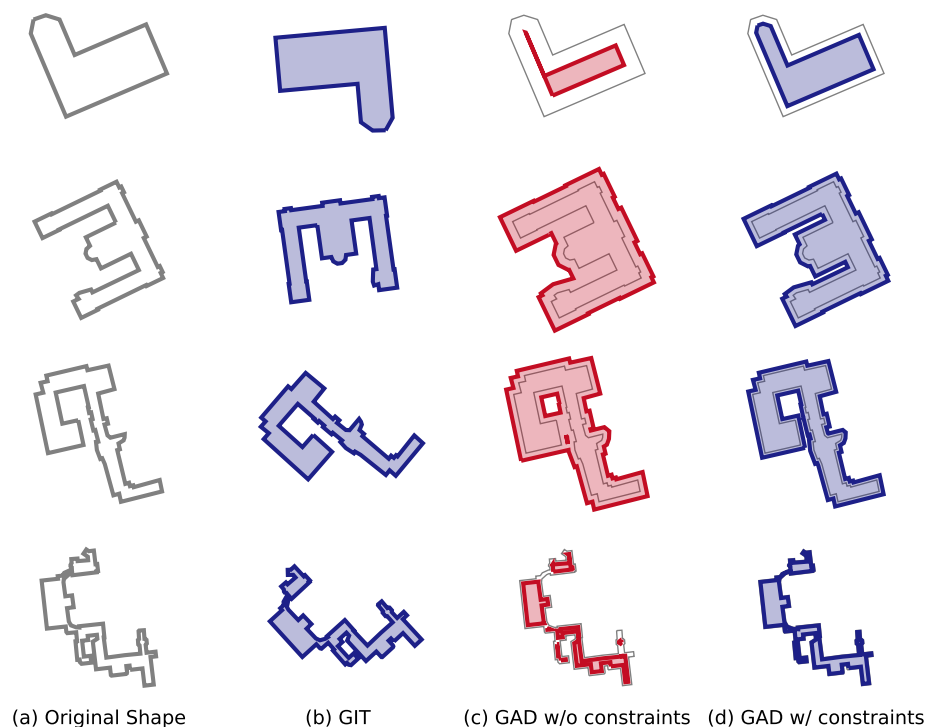


Figure 8. Illustration of different deformation strategies. (a) Original footprint. (b) GITs introduce only rigid or low-level geometric changes. (c) GADs without validity constraints can produce unrealistic or semantically implausible shapes. (d) GADs with validity constraints generate diverse yet structurally plausible variants, ensuring positive pairs provide reliable supervision for contrastive learning.

Overall, results show that the advantage of GAD lies in aligning deformation with geometric structure to construct meaningful similarity relations. Validity constraints are essential to prevent structural collapse, an optimal deformation regime exists around 0.1π ,

and the deformation design directly determines the induced similarity structure of the embedding space.

5. Discussion

Beyond empirical performance, DI-SSL also offers a representation-centric view of morphological analysis by organizing building shapes through deformation-induced similarity rather than predefined metrics. This enables scalable clustering, retrieval, and large-scale analysis within a unified embedding space. By deriving supervision from GAD, the approach embeds morphological priors as explicit structural relations, moving beyond heuristic augmentation toward principled similarity construction. While the curated benchmark primarily evaluates the intrinsic quality of the learned morphological representation, it does not by itself establish its practical utility or generalizability in real-world settings. To provide an initial assessment in an unseen urban context, we therefore examine a 3.5×2.7 km urban region near Shanghai South Station containing 3347 building footprints (Figure 9).



Figure 9. (a) Low-dimensional visualization of the embeddings for complex building shapes, together with representative instances from each cluster, reveals that recurring geometric patterns are consistently grouped in the embedding space. (b) When cluster labels are mapped back to geographic space for all building footprints, buildings assigned to the same cluster exhibit localized spatial aggregation in several areas.

The embeddings are first projected to a low-dimensional space using UMAP, where a dominant cluster corresponding to simple geometric forms (e.g., rectangles and circular shapes) can be readily identified. After excluding these highly regular structures, the remaining embeddings—representing more complex shapes—are further analyzed through an additional dimensionality reduction and clustering step (k-means, $k = 26$). As shown in Figure 9a, the resulting clusters exhibit coherent geometric patterns, with recurring shape configurations consistently grouped together. To provide spatial context, the cluster assignments are mapped back to geographic space (Figure 9b). Despite no spatial information being used during training, buildings assigned to the same cluster tend to exhibit localized spatial aggregation, in some cases forming visually identifiable groups. To quantitatively assess this spatial organization, we further computed cluster-wise binary

Moran's I statistics using an 8-nearest-neighbor spatial weight matrix, with significance assessed by permutation tests and corrected for multiple comparisons using the false discovery rate. Among the 26 clusters, 22 (84.6%) exhibited significant positive spatial autocorrelation ($q < 0.05$), with Moran's I values ranging from approximately -0.005 to 0.268 (Table A12 in Appendix A.8). This observation suggests that the learned representation captures certain structural regularities that are not only geometrically meaningful but also spatially organized in real-world settings. Nevertheless, this case study provides a preliminary assessment of real-world spatial applicability rather than a task-specific downstream evaluation, and future work could further validate the utility of DI-SSL embeddings in applications such as urban functional zone classification or building retrieval using weak, noisy, or expert-annotated ground truth. The observed patterns may also depend on the specific urban context and clustering configuration.

Several limitations of the proposed framework should be acknowledged: (1) The learned representation depends on deformation design, which defines admissible structural variation; biased designs may yield incomplete or distorted similarity structures. (2) The framework does not aim to capture a universal notion of shape similarity, but rather a controlled, deformation-induced one; performance may degrade for highly irregular or weakly structured shapes (Section 4.4, Appendix A.4). (3) While the framework is designed to be encoder-agnostic, representation quality remains constrained by the model's capacity to capture global geometry. These limitations suggest future directions, including more adaptive deformation design and extensions to complex representations (e.g., polygons with holes, multi-part footprints, 3D structures, and building groups), enabling richer yet structurally grounded similarity modeling.

6. Conclusions

This work introduces DI-SSL, a framework for learning building shape representations by explicitly constructing similarity relationships. The results show that the learned embedding space is not only discriminative but also structurally organized, where similarity is defined through geometry-consistent deformations rather than implicit instance equivalence or predefined metrics. The observed structure arises from enforcing consistency under constrained deformations, and are supported by strong performance in retrieval and few-shot tasks. More broadly, this work highlights the importance of explicitly defining similarity in representation learning for shape analysis. By constructing controlled relationships between shapes, the proposed framework provides a data-driven approach to organizing building shape space, where similarities reflect morphologically meaningful variations instead of discrete categories or handcrafted metrics. This structured representation supports expressive analysis in large-scale geospatial data. Several limitations remain. The approach depends on a predefined deformation space, which may not capture all relevant forms of geometric variation, and its extension to more complex shape representations requires further investigation. Future work may explore adaptive mechanisms for constructing similarity and broader applications in geospatial analysis.

Author Contributions: Conceptualization, Shuqi Cao and Guohua Ji; methodology, Shuqi Cao; software, Shuqi Cao; formal analysis, Shuqi Cao; investigation, Shuqi Cao; data curation, Shuqi Cao; writing—original draft preparation, Shuqi Cao; writing—review and editing, Guohua Ji; visualization, Shuqi Cao; supervision, Guohua Ji; funding acquisition, Guohua Ji. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (No. 52178017) and the Jiangsu Province Special Fund for Science and Technology Innovation on Carbon Peaking and Carbon Neutrality (Grant No. BT2025028).

Data Availability Statement: The data and codes that support the findings of this study are available in Figshare at <https://doi.org/10.6084/m9.figshare.32211402> (accessed on 1 September 2026)

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

Appendix A.1. Encoder-Agnostic Representation Learning

To evaluate the generalization of DI-SSL across different representation forms, we conduct experiments using a diverse set of polygon encoders spanning multiple types. The goal of this analysis is not to identify the optimal encoder but to examine whether deformation-guided contrastive learning remains effective under heterogeneous representations. Specifically, we consider four distinct representation types: (1) raster-based encoding (Resnet2D), (2) sequence-based encoders (ResNet1D, LineStringNet), (3) spectral-based representations (NUFT+MLP, Poly2Vec), and (4) token-based representations (VPT+DNN). Unless otherwise specified, all experiments follow the training protocol and deformation settings described in Section 4.1.2, varying only the encoder architecture, for which the deformation input is converted into a suitable representation accordingly. The results are summarized in Table A1.

Table A1. Generalization analysis across different polygon encoders spanning multiple types.

Encoder	Type	NN	Retrieval			Few-Shot		
			FT	ST	nDCG	1-Shot	3-Shot	5-Shot
Resnet2D ^a	raster-based	0.992	0.630	0.779	0.929	0.659	0.798	0.846
Resnet1D ^b (default)	sequence-based	0.997	0.804	0.946	0.977	0.851	0.941	0.961
LineStringNet ^c	sequence-based	0.979	0.734	0.894	0.955	0.769	0.853	0.881
NUFT+MLP ^d	spectral-based	0.988	0.719	0.870	0.954	0.755	0.870	0.903
Poly2Vec ^e	spectral-based	0.961	0.617	0.807	0.925	0.643	0.766	0.807
VPT+DNN ^f	token-based	0.980	0.584	0.726	0.914	0.610	0.762	0.816

Architecture details: ^a ResNet2D (14.4M parameters) uses ResNet18 and takes rasterized inputs of $128 \times 128 \times 1$; ^b ResNet1D (4.4M parameters) uses an 18-layer 1D ResNet with DeltaNet features and takes 128×2 sequences; ^c LineStringNet (190K parameters) processes 128×2 point sequences; ^d NUFT+MLP (1.1M parameters) processes 128×2 vertex sequences through spectral transformation; ^e Poly2Vec (815K parameters) takes variable-length triangulated polygons as input, which are generated via constrained Delaunay triangulation with ‘pYq’ switches to preserve boundaries and enforce quality, consistent with the original configuration. ^f VPT+DNN (29.3M parameters) processes 64×1 tokenized inputs.

A consistent observation across all encoders is that DI-SSL produces stable and non-degenerate representations, as evidenced by uniformly high NN scores and consistent performance across both retrieval and few-shot metrics. All representation types achieve strong nearest-neighbor retrieval performance and maintain reasonable effectiveness across multiple evaluation metrics, indicating that the framework can be applied to heterogeneous feature spaces without requiring encoder-specific design. This supports the claim that DI-SSL operates by imposing deformation-induced relational structure onto an existing representation space, rather than relying on a particular encoding mechanism.

Despite this cross-type consistency, notable performance differences emerge, which can be attributed to the degree of geometric fidelity preserved by each encoder, i.e., how directly spatial relationships in the original polygon are retained in the intermediate representation. Continuous sequence-based encoders achieve the most balanced performance, as they directly model ordered vertex coordinates and thus preserve both local geometric continuity and global structural relationships. Raster- and spectral-based encoders show

comparatively lower performance, suggesting that rasterization and frequency-domain transformations introduce abstraction that weakens explicit spatial correspondence.

While DI-SSL is designed to be encoder-agnostic, this property implicitly assumes that the encoder preserves the geometric structure of the input space. For geometry-faithful encoders (e.g., continuous coordinate-based models), DI-SSL consistently achieves strong performance. In contrast, for token-based encoders such as VPT, which rely on discretization and anisotropic normalization, this assumption is violated. As a result, the learned representation no longer reflects intrinsic shape similarity, leading to significant performance degradation under isotropic scaling. This observation highlights that DI-SSL does not impose invariance by itself, but rather leverages the consistency of the underlying representation space.

These results indicate that DI-SSL is largely encoder-agnostic across geometry-preserving representations, but not representation-neutral. While the framework can consistently organize embeddings across diverse representation types, its effectiveness is ultimately bounded by the encoder’s ability to preserve geometric structure. Performance variations should therefore be interpreted as a consequence of representational alignment with polygonal geometry, rather than differences in the learning objective itself.

Appendix A.2. Model Hyperparameter Ablation Study

Table A2. Effect analysis across depth.

Encoder	Layer	NN	Retrieval			Few-Shot		
			FT	ST	nDCG	1-Shot	3-Shot	5-Shot
Resnet1D	18 (default)	0.997	0.804	0.946	0.977	0.851	0.941	0.961
Resnet1D	34	0.997	0.808	0.936	0.977	0.852	0.942	0.962
Resnet1D	50	0.996	0.780	0.912	0.971	0.822	0.932	0.957

Table A2 shows that the depth of ResNet1D does not significantly affect performance metrics; even the 50-layer ResNet1D shows slight degradation. This suggests that over-parameterization may hinder the preservation of fine-grained geometric structure, possibly due to excessive smoothing or optimization difficulty in the contrastive setting.

Table A3. Effect analysis across the temperature parameter.

Encoder	Temperature	NN	Retrieval			Few-Shot		
			FT	ST	nDCG	1-Shot	3-Shot	5-Shot
Resnet1D	0.10	0.998	0.822	0.940	0.979	0.859	0.952	0.972
Resnet1D	0.05	0.998	0.824	0.952	0.981	0.868	0.951	0.969
Resnet1D	0.02	0.996	0.805	0.949	0.978	0.853	0.941	0.960
Resnet1D	0.01 (default)	0.997	0.804	0.946	0.977	0.851	0.941	0.961
Resnet1D	0.005	0.996	0.798	0.942	0.976	0.844	0.937	0.956

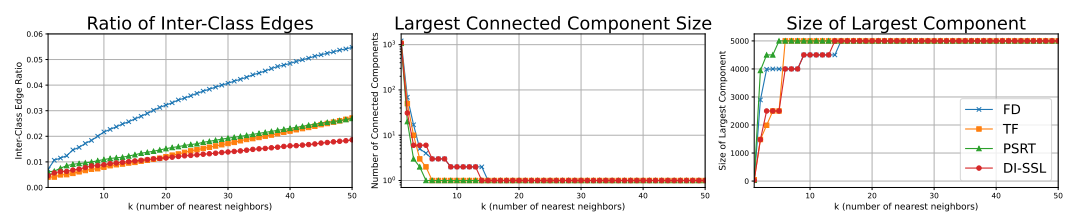
Table A3 shows that the temperature parameter τ of the contrastive loss has no significant impact on performance metrics. Although $\tau = 0.05$ yields the best results, our default setting of $\tau = 0.01$ is also reasonable.

Table A4. Effect analysis across embedding dimensions.

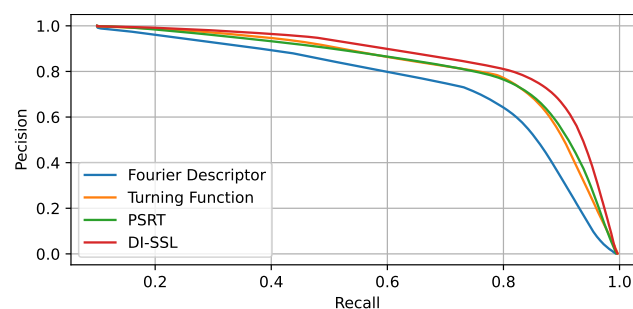
Encoder	Dimensions	NN	Retrieval			Few-Shot		
			FT	ST	nDCG	1-Shot	3-Shot	5-Shot
Resnet1D	16	0.996	0.695	0.807	0.946	0.721	0.875	0.920
Resnet1D	32	0.996	0.781	0.910	0.971	0.820	0.926	0.952
Resnet1D	128	0.997	0.798	0.932	0.975	0.841	0.934	0.957
Resnet1D	512 (default)	0.997	0.804	0.946	0.977	0.851	0.941	0.961

Table A4 shows that performance improves as a function of embedding dimensionality. Increasing the dimension from 32 to 512 yields continued yet modest gains, whereas decreasing it from 32 to 16 leads to a pronounced degradation. This suggests that a minimum dimensionality of 32 is necessary for sufficient representational capacity.

Appendix A.3. Further Evidence Supporting Well-Structured Embedding Space

**Figure A1.** KNN graph statistics.

As shown in Figure A1, the inter-class edge ratio remains consistently low across all values of k , increasing only from 0.46% at $k = 1$ to 1.86% at $k = 50$. Meanwhile, the graph rapidly becomes connected as k increases, with the largest connected component covering the entire dataset at $k = 14$. This indicates that although the global structure is well connected, local neighborhoods are highly class-consistent, suggesting strong separation between classes in the embedding space.

**Figure A2.** Precision–recall curves of different methods.

As shown in Figure A2, DI-SSL consistently achieves the highest precision across the full recall range, indicating superior retrieval quality and robustness. Compared to traditional descriptors, the Turning Function and PSRT exhibit competitive performance at moderate recall levels but degrade more rapidly as recall approaches 1.0, while the Fourier Descriptor shows the weakest overall performance with a steady decline. In contrast, DI-SSL maintains a clear advantage, particularly in the high-recall regime, suggesting a more effective preservation of fine-grained shape similarity in the learned embedding space.

Appendix A.4. Neighborhood Structure Under Different Metrics

We compare the nearest-neighbor (NN) relationships induced by the two metrics, examining how each metric organizes local neighborhoods in the shape space.

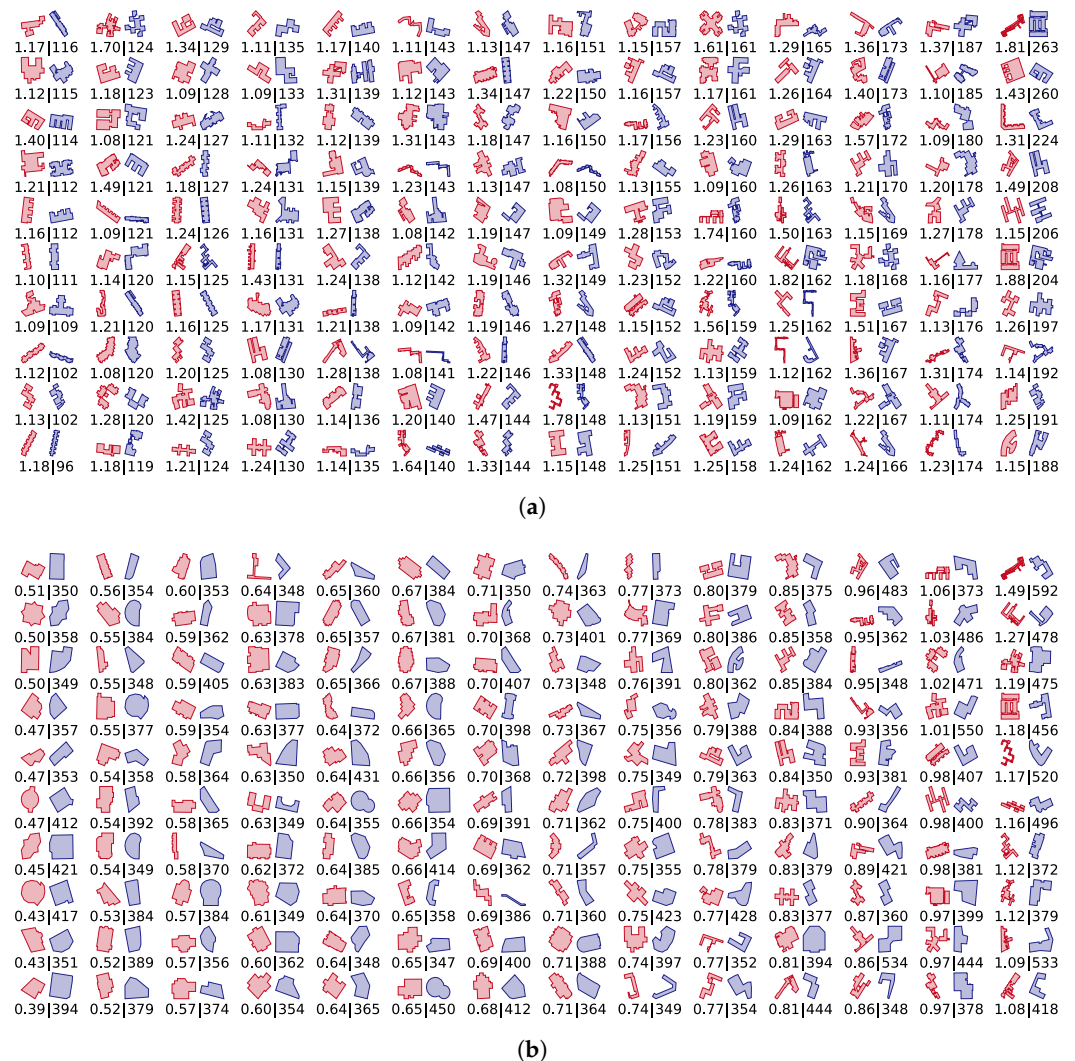


Figure A3. Retrieval targets (red) and their Nearest Neighbor (blue) in the randomly sampled sub-datasets, with $d_{TF} | d_{emb}$ values donating below each pair: (a) 140 shape pairs with the largest d_{emb} among all NN pairs retrieval by d_{emb} ; (b) 140 shape pairs with the largest d_{TF} among all NN pairs retrieval by d_{TF} .

We first analyze nearest neighbors defined by the embedding distance d_{emb} . To probe the behavior under extreme conditions, we focus on the 140 shape pairs with the largest embedding distances among all NN pairs (Figure A3a). These pairs predominantly involve shapes with high geometric complexity and rich local details. Despite their large TF distances (typically $d_{TF} > 1.0$), the retrieved neighbors generally preserve coarse structural similarity to the query shapes. This observation is consistent with the high- d_{TF} , low- d_{emb} region identified in Section 4.4, indicating that the embedding remains sensitive to structural organization even when geometric discrepancies are substantial.

However, this structural consistency becomes less pronounced under extreme geometric complexity. In such cases, the embedding appears less compressive with respect to global pairwise distances, and the notion of structural similarity becomes increasingly ambiguous. Two factors may contribute to this behavior. First, highly complex shapes often lack clearly defined global organization, making structural correspondence inherently less stable. Second, data sparsity may limit the availability of sufficiently similar counterparts within the sampled dataset, further reducing the observable consistency in nearest-neighbor relationships. As a result, while the embedding preserves structural similarity in many

cases, its discriminative power degrades in regions characterized by extreme geometric variation and limited sample support.

We then examine nearest neighbors defined by the TF distance d_{TF} , again focusing on the 140 pairs with the largest TF distances among NN relationships (Figure A3b). In contrast to the embedding-based neighborhoods, these pairs exhibit substantially greater variability in structural organization. Although they are identified as nearest neighbors under d_{TF} , the corresponding shapes often differ markedly in their global structure, reflecting the fact that TF primarily captures boundary-level geometric similarity and is less sensitive to higher-level structural configuration. Consequently, TF-induced neighborhoods are less aligned with perceptual or structural similarity when shapes undergo complex or non-uniform deformations.

Taken together, these observations highlight a fundamental difference in how the two metrics organize local neighborhoods. Embedding-based nearest neighbors tend to preserve structural relationships, even in the presence of significant geometric variation, whereas TF-based neighborhoods are more strongly influenced by boundary-level similarity and may group structurally dissimilar shapes. These neighborhood-level behaviors are consistent with the cross-metric distance regions identified earlier, and further support the view that the learned embedding induces a structurally reorganized notion of shape similarity.

Appendix A.5. Computational Efficiency and Sensitivity of Straight Skeleton Deformation

To further assess the computational implications of the deformation-range parameter α , we evaluated the preprocessing pipeline under four representative values, $\alpha \in \{0.1, 0.25, 0.5, 1.0\}$, while fixing the TF-based validity threshold at $\tau_{TF} = 0.1\pi$. For each setting, the complete input set of 12,298,902 polygons was processed using the same implementation and computational environment. Following the procedure described in Section 3.2.4, we performed 400 sampling attempts for each polygon and recorded the resulting admissible δ pairs. These sampling attempts were used to identify admissible deformation parameters rather than to generate 400 training pairs; after sampling, one admissible δ pair was ultimately selected for each input polygon, yielding 12,298,902 polygon pairs in total. The selected δ pairs were then used to perform SS-Offsetting and resampling to 128 points, producing the 128-point polygon sequence pairs used for training. Parallel preprocessing was performed using `joblib.Parallel` with 60 workers. We separately recorded the wall-clock time for the δ -sampling stage and the subsequent SS-Offsetting and resampling stage. We also recorded the valid pair rate and the proportions of generated pairs satisfying $\tau_{TF} > 0.05\pi$ and $\tau_{TF} > 0.09\pi$ and evaluated the resulting representations under the same downstream evaluation protocol.

Table A5. Straight Skeleton Deformation Preprocessing time statistic across deformation-range parameter α , while fixing the TF-based validity threshold at $\tau_{TF} = 0.1\pi$.

Deformation-range parameter (α)	Sampling Stage Runtime (s)	Subsequent Stage Runtime (s)	Valid Pair Rate	$\tau_{TF} > 0.05\pi$ Rate	$\tau_{TF} > 0.09\pi$ Rate
0.1	2387.5	1093.2	99.9%	90.5%	39.8%
0.25	2524.8	1067.0	99.9%	93.1%	46.4%
0.5 (default)	2585.0	1266.8	99.9%	93.4%	47.7%
1.0	2753.0	1064.7	99.9%	93.5%	47.8%

As shown in Table A5, the preprocessing time increased from 2387.5 s at $\alpha = 0.1$ to 2753.0 s at $\alpha = 1.0$. Thus, increasing the deformation range from 0.1 to 1.0 resulted in an approximately 15.3% increase in preprocessing time. The results in Table A6 show that the learned representations are robust to the tested deformation ranges: NN, FT, ST, and

nDCG vary within 0.995–0.997, 0.798–0.807, 0.943–0.951, and 0.976–0.977, respectively. The default $\alpha = 0.5$ therefore represents a stable intermediate operating point rather than a highly sensitive parameter choice.

Table A6. Effect analysis across deformation-range parameter α .

Deformation-Range Parameter (α)	Retrieval			Few-Shot			
	NN	FT	ST	nDCG	1-Shot	3-Shot	5-Shot
0.1	0.996	0.807	0.951	0.977	0.854	0.940	0.959
0.25	0.996	0.798	0.943	0.976	0.847	0.939	0.960
0.5 (default)	0.997	0.804	0.946	0.977	0.851	0.941	0.961
1.0	0.995	0.800	0.943	0.976	0.848	0.940	0.960

Appendix A.6. Training Configuration and Runtime Records

This section provides detailed records of the training configuration and runtime behavior used to train DI-SSL. Table A7 summarizes the complete training configuration, including the batch size, data-loading settings, numerical precision, optimizer, and hardware environment. Table A8 reports the measured epoch-level runtimes across independent runs.

Table A7. Actual training configuration recorded by W&B.

Configuration	Value
Training pairs	12,298,902
Effective pairs/epoch	12,298,240
Input	128-point polygon sequences pair
Batch size	1024
drop last	TRUE
Batches/epoch	12,010
Epochs	10
Optimization steps/epoch	120,100
Backbone	ResNet1D-18
Parameters	4.39 M
Precision	BF16 mixed precision
GPU	1 × RTX A6000
Data-loader workers	60

Table A8. The observed epoch time (s) across 10 independent runs.

Seed	Epoch 0	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9
40	494.4 s	474.9 s	483.9 s	484.3 s	493.4 s	490.9 s	483.1 s	502.8 s	479.5 s	482.7 s
41	663.9 s	619.8 s	599.0 s	615.5 s	628.2 s	643.4 s	620.8 s	647.0 s	651.0 s	620.4 s
42	535.7 s	524.7 s	484.1 s	501.6 s	492.0 s	498.5 s	508.1 s	513.1 s	500.1 s	502.4 s
43	462.2 s	456.9 s	453.4 s	451.6 s	453.9 s	454.7 s	453.9 s	457.8 s	450.5 s	451.9 s
44	450.6 s	440.0 s	440.6 s	440.3 s	437.8 s	442.8 s	447.5 s	442.5 s	441.2 s	446.9 s
45	452.1 s	441.1 s	441.1 s	426.3 s	439.6 s	437.0 s	423.6 s	429.4 s	430.1 s	441.4 s
46	490.1 s	450.2 s	446.4 s	545.6 s	477.0 s	841.0 s	1148.4 s	647.7 s	556.3 s	536.4 s
47	647.1 s	825.6 s	479.2 s	483.3 s	580.6 s	560.7 s	480.2 s	483.9 s	567.6 s	739.0 s
48	495.6 s	475.0 s	473.7 s	601.0 s	578.0 s	483.6 s	466.1 s	473.7 s	577.6 s	480.5 s
49	490.0 s	587.1 s	484.8 s	476.3 s	472.7 s	486.6 s	577.0 s	480.7 s	479.5 s	576.5 s

Figure A4 further visualizes the training loss, GPU utilization, and allocated GPU memory throughout training. Several runs exhibit irregular GPU memory-allocation patterns under the WDDM driver mode used on the experimental system, which may contribute to the observed runtime variability. These records document the actual computational configuration and runtime behavior of the reported experiments. The expensive

SS-offsetting, TF-based validity checking, and vertex resampling operations are performed offline during dataset preprocessing and are not repeated during model training.

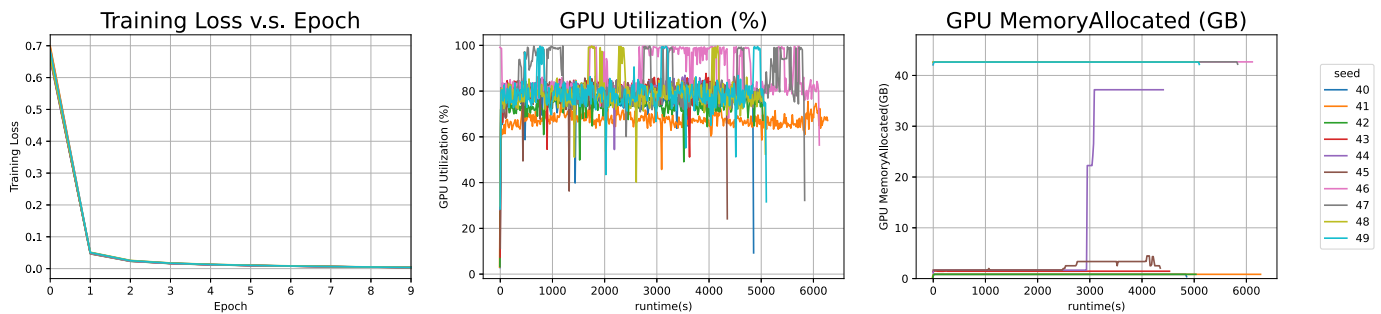


Figure A4. Runtime monitoring during training. Different colors denote different random seeds.

Appendix A.7. Statistical Details of Performance Comparison in Section 4.2.2

This subsection provides detailed statistical results to complement the condensed performance summary in Table 1 of Section 4.2.2.

Table A9 reports the full *mean* $\pm$ *std* deviation and 95% confidence intervals (CI) for every method across all benchmark datasets. The CI is computed using Student's t-distribution with 9 degrees of freedom ($\alpha = 0.05$). These results confirm that the performance gaps reported in the main text are statistically robust.

Table A9. Comprehensive evaluation results of all learning-based methods on benchmark datasets.

Encoder	Resampled N	NN	FT	ST	nDCG	1-Shot	3-Shot	5-Shot
GGCAE	64	0.8952 ± 0.0161 ([0.8913, 0.8990])	0.2927 ± 0.0261 ([0.2865, 0.2990])	0.4118 ± 0.0290 ([0.4049, 0.4188])	0.8002 ± 0.0095 ([0.7979, 0.8024])	0.2907 ± 0.0278 ([0.2841, 0.2973])	0.4215 ± 0.0313 ([0.4140, 0.4289])	0.4878 ± 0.0314 ([0.4803, 0.4952])
NUFT-based AE	128	0.9044 ± 0.0033 ([0.9036, 0.9052])	0.2204 ± 0.0030 ([0.2197, 0.2211])	0.2978 ± 0.0018 ([0.2974, 0.2982])	0.7641 ± 0.0010 ([0.7638, 0.7643])	0.2102 ± 0.0029 ([0.2095, 0.2109])	0.3084 ± 0.0038 ([0.3075, 0.3093])	0.3688 ± 0.0040 ([0.3679, 0.3698])
PSRT	64	0.9948 ± 0.0008 ([0.9946, 0.9950])	0.7700 ± 0.0094 ([0.7678, 0.7723])	0.9021 ± 0.0105 ([0.8996, 0.9046])	0.9675 ± 0.0025 ([0.9669, 0.9680])	0.8113 ± 0.0108 ([0.8087, 0.8139])	0.9172 ± 0.0056 ([0.9159, 0.9186])	0.9432 ± 0.0041 ([0.9423, 0.9442])
DI-SSL (default)	128	0.9966 ± 0.0002 ([0.9964, 0.9967])	0.8040 ± 0.0020 ([0.8026, 0.8054])	0.9464 ± 0.0019 ([0.9451, 0.9478])	0.9771 ± 0.0004 ([0.9769, 0.9774])	0.8510 ± 0.0019 ([0.8496, 0.8524])	0.9409 ± 0.0010 ([0.9402, 0.9416])	0.9608 ± 0.0008 ([0.9603, 0.9614])
DI-SSL	64	0.9971 ± 0.0004 ([0.9970, 0.9972])	0.8058 ± 0.0059 ([0.8044, 0.8072])	0.9385 ± 0.0044 ([0.9374, 0.9395])	0.9770 ± 0.0012 ([0.9768, 0.9773])	0.8509 ± 0.0061 ([0.8495, 0.8524])	0.9420 ± 0.0034 ([0.9412, 0.9428])	0.9626 ± 0.0024 ([0.9620, 0.9631])

All values are reported as *mean* $\pm$ *std* across 10 independent runs with different random seeds, with 95% confidence intervals (CI) shown in brackets. For few-shot classification, each run is evaluated over 1000 randomly sampled episodes.

Beyond run-level aggregation, Table A10 provides a finer-grained analysis of DI-SSL's few-shot evaluation. Specifically, for each of the 10 random seeds, we report episode-level statistics across the 1,000 evaluation episodes used in the standard *N*-way *K*-shot protocol. Table A11 reports the same episode-level statistics for the strongest baseline (PSRT). Both methods exhibit high stability with small standard deviations. Critically, DI-SSL consistently outperforms PSRT across all shot settings, demonstrating that its superiority is not occasional but systematic and robust to episode sampling variability.

Table A10. Episode-level statistics of DI-SSL’s 1000-episode few-shot evaluation.

Seed	1-Shot	3-Shot	5-Shot
40	0.8538 ± 0.0450 ([0.8510, 0.8566])	0.9421 ± 0.0191 ([0.9409, 0.9433])	0.9611 ± 0.0125 ([0.9604, 0.9619])
41	0.8483 ± 0.0454 ([0.8455, 0.8511])	0.9403 ± 0.0197 ([0.9390, 0.9415])	0.9608 ± 0.0126 ([0.9600, 0.9615])
42	0.8562 ± 0.0442 ([0.8535, 0.8590])	0.9434 ± 0.0189 ([0.9423, 0.9446])	0.9622 ± 0.0127 ([0.9614, 0.9630])
43	0.8395 ± 0.0456 ([0.8367, 0.8423])	0.9344 ± 0.0200 ([0.9332, 0.9357])	0.9562 ± 0.0132 ([0.9554, 0.9570])
44	0.8572 ± 0.0441 ([0.8545, 0.8600])	0.9436 ± 0.0188 ([0.9424, 0.9447])	0.9627 ± 0.0124 ([0.9620, 0.9635])
45	0.8579 ± 0.0452 ([0.8551, 0.8607])	0.9459 ± 0.0188 ([0.9447, 0.9470])	0.9649 ± 0.0120 ([0.9641, 0.9656])
46	0.8450 ± 0.0457 ([0.8422, 0.8478])	0.9397 ± 0.0196 ([0.9385, 0.9409])	0.9608 ± 0.0127 ([0.9600, 0.9616])
47	0.8468 ± 0.0452 ([0.8440, 0.8496])	0.9416 ± 0.0198 ([0.9404, 0.9428])	0.9625 ± 0.0128 ([0.9617, 0.9633])
48	0.8497 ± 0.0447 ([0.8469, 0.8525])	0.9394 ± 0.0195 ([0.9382, 0.9406])	0.9599 ± 0.0129 ([0.9591, 0.9607])
49	0.8555 ± 0.0449 ([0.8528, 0.8583])	0.9387 ± 0.0188 ([0.9376, 0.9399])	0.9573 ± 0.0125 ([0.9565, 0.9580])

For each seed, the mean, standard deviation, and 95% confidence interval are computed across 1000 randomly sampled episodes.

Table A11. Episode-level statistics of PSRT’s 1000-episode few-shot evaluation.

Seed	1-Shot	3-Shot	5-Shot
40	0.8163 ± 0.0485 ([0.8133, 0.8193])	0.9185 ± 0.0195 ([0.9173, 0.9197])	0.9437 ± 0.0125 ([0.9429, 0.9444])
41	0.8155 ± 0.0463 ([0.8127, 0.8184])	0.9201 ± 0.0195 ([0.9189, 0.9213])	0.9461 ± 0.0122 ([0.9453, 0.9468])
42	0.8187 ± 0.0484 ([0.8157, 0.8217])	0.9218 ± 0.0183 ([0.9206, 0.9229])	0.9469 ± 0.0119 ([0.9461, 0.9476])
43	0.8149 ± 0.0493 ([0.8119, 0.8180])	0.9211 ± 0.0187 ([0.9199, 0.9223])	0.9466 ± 0.0116 ([0.9459, 0.9473])
44	0.8044 ± 0.0509 ([0.8012, 0.8076])	0.9109 ± 0.0209 ([0.9096, 0.9122])	0.9376 ± 0.0131 ([0.9368, 0.9384])
45	0.7866 ± 0.0517 ([0.7834, 0.7898])	0.9057 ± 0.0221 ([0.9043, 0.9071])	0.9355 ± 0.0141 ([0.9346, 0.9363])
46	0.8239 ± 0.0468 ([0.8210, 0.8268])	0.9227 ± 0.0182 ([0.9215, 0.9238])	0.9468 ± 0.0121 ([0.9461, 0.9476])
47	0.8224 ± 0.0485 ([0.8194, 0.8254])	0.9240 ± 0.0189 ([0.9228, 0.9251])	0.9477 ± 0.0119 ([0.9469, 0.9484])
48	0.8006 ± 0.0500 ([0.7975, 0.8037])	0.9131 ± 0.0211 ([0.9118, 0.9145])	0.9407 ± 0.0135 ([0.9399, 0.9415])
49	0.8097 ± 0.0468 ([0.8068, 0.8126])	0.9143 ± 0.0193 ([0.9131, 0.9155])	0.9409 ± 0.0127 ([0.9401, 0.9417])

For each seed, the mean, standard deviation, and 95% confidence interval are computed across 1,000 randomly sampled episodes.

Appendix A.8. Quantitative Spatial-Clustering Statistic to Support Section 5’s Case Study

Table A12. Moran’s I analysis on cluster labels.

Cluster	N	Moran’s	z	q
A	58	0.010877	1.359467	0.111913
B	16	−0.004728	−0.592077	0.391083
C	46	0.025434	3.041389	0.005200
D	47	−0.000612	−0.052767	0.486000
E	43	0.266969	32.683994	0.001368
F	37	0.180201	22.500317	0.001368
G	30	0.213485	26.999278	0.001368
H	63	0.267929	33.284257	0.001368
I	36	0.080160	10.020329	0.001368
J	39	0.205108	25.471461	0.001368
K	32	0.062356	7.572149	0.001368
L	35	0.151619	18.122665	0.001368
M	53	0.044808	5.738544	0.001368
N	36	0.060497	7.915120	0.001368
O	65	0.150074	18.482109	0.001368
P	45	0.098832	11.312732	0.001368
Q	42	0.157325	19.776822	0.001368
R	112	0.132275	16.262358	0.001368

Table A12. Cont.

Cluster	N	Moran's	z	q
S	62	0.044920	5.359068	0.001368
T	26	0.031498	3.796968	0.008667
U	43	0.019189	2.365414	0.023636
V	59	0.110771	13.609504	0.001368
W	24	0.176950	22.980905	0.001368
X	51	0.216981	25.999775	0.001368
Y	34	-0.001589	-0.168139	0.466960
Z	47	0.048481	6.015069	0.001368

References

1. Cao, Q.; Luan, Q.; Liu, Y.; Wang, R. The effects of 2d and 3d building morphology on urban environments: A multi-scale analysis in the beijing metropolitan region. *Build. Environ.* **2021**, *192*, 107635. [\[CrossRef\]](#)
2. Dibble, J.; Prelorndjos, A.; Romice, O.; Zanella, M.; Strano, E.; Pagel, M.; Porta, S. On the origin of spaces: Morphometric foundations of urban form evolution. *Environ. Plan. B Urban Anal. City Sci.* **2019**, *46*, 707–730. [\[CrossRef\]](#)
3. Durst, N.J.; Sullivan, E.; Jochem, W.C. The spatial and social correlates of neighborhood morphology: Evidence from building footprints in five u.s. metropolitan areas. *PLoS ONE* **2024**, *19*, e0299713. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Cao, J.; Chen, J.; Wang, X.; Huang, W.; Chen, D.; Zhao, T.; Tu, W.; Li, Q. Urbanmml: Urban region representations via multi-modal and multi-graph self-supervised contrastive learning. *Isprs J. Photogramm. Remote Sens.* **2025**, *232*, 75–93. [\[CrossRef\]](#)
5. Fleischmann, M.; Feliciotti, A.; Kerr, W. Evolution of urban patterns: Urban morphology as an open reproducible data science. *Geogr. Anal.* **2022**, *54*, 536–558. [\[CrossRef\]](#)
6. Xu, J.; Zhang, H.; Liu, C.; Guo, J. Recognition of building shape in maps using deep graph filter neural network. *Geocarto Int.* **2023**, *38*, 2272662. [\[CrossRef\]](#)
7. Yan, X.F.; Yang, M. A comparative study of various deep learning approaches to shape encoding of planar geospatial objects. *Int. J. Geo-Inf.* **2022**, *11*, 527. [\[CrossRef\]](#)
8. Chen, D.; Feng, Y.; Li, X.; Qu, M.; Luo, P.; Meng, L. Interpreting core forms of urban morphology linked to urban functions with explainable graph neural network. *Comput. Environ. Urban Syst.* **2025**, *118*, 102267. [\[CrossRef\]](#)
9. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. *Int. Conf. Mach. Learn.* **2020**, *119*, 1597–1607.
10. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
11. Wang, J.; Huang, W.; Biljecki, F. Learning visual features from figure-ground maps for urban morphology discovery. *Comput. Environ. Urban Syst.* **2024**, *109*, 102076. [\[CrossRef\]](#)
12. Li, Z.; Chen, H.; Lin, B.; Zhu, Y. Fast bidirectional building performance optimization at the early design stage. *Build. Simul.* **2018**, *11*, 647–661. [\[CrossRef\]](#)
13. Yan, X.; Ai, T.; Yang, M.; Tong, X. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 490–512. [\[CrossRef\]](#)
14. Basaraner, M.; Cetinkaya, S. Performance of shape indices and classification schemes for characterising perceptual shape complexity of building footprints in gis. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1952–1977. [\[CrossRef\]](#)
15. Biljecki, F.; Chow, Y.S. Global building morphology indicators. *Comput. Environ. Urban Syst.* **2022**, *95*, 101809. [\[CrossRef\]](#)
16. Bourdic, L.; Salat, S.; Nowacki, C. Assessing cities: A new system of cross-scale spatial indicators. *Build. Res. Inf.* **2012**, *40*, 592–605. [\[CrossRef\]](#)
17. Ai, T.; Cheng, X.; Liu, P.; Yang, M. A shape analysis and template matching of building features by the fourier transform method. *Comput. Environ. Urban Syst.* **2013**, *41*, 219–233. [\[CrossRef\]](#)
18. Rodrigues, E.; Sousa-Rodrigues, D.; de Sampayo, M.T.; Gaspar, A.R.; Gomes, Á.; Antunes, C.H. Clustering of architectural floor plans: A comparison of shape representations. *Autom. Constr.* **2017**, *80*, 48–65. [\[CrossRef\]](#)
19. Destler, N.; Singh, M.; Feldman, J. Skeleton-based shape similarity. *Psychol. Rev.* **2023**, *130*, 1653–1671. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Lu, X.C.; Li, H.; Xu, Y.; Liu, J.; Chen, Z. Measuring the similarity between shapes of buildings using graph edit distance. *Int. J. Digit. Earth* **2024**, *17*, 2310749. [\[CrossRef\]](#)
21. Yu, D.; Hu, Y.; Li, Y.; Zhao, L. Polygongnn: Representation learning for polygonal geometries with heterogeneous visibility graph. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024.

22. Li, P.B.; Yan, H.W.; Lu, X.M. A siamese neural network for learning the similarity metrics of linear features. *Int. J. Geogr. Inf. Sci.* **2023**, *37*, 684–711. [[CrossRef](#)]
23. Mai, G.; Jiang, C.; Sun, W.; Zhu, R.; Xuan, Y.; Cai, L.; Janowicz, K.; Ermon, S.; Lao, N. Towards general-purpose representation learning of polygonal geometries. *GeoInformatica* **2022**, *27*, 289–340. [[CrossRef](#)]
24. Siampou, M.D.; Li, J.; Krumm, J.; Shahabi, C.; Lu, H. Poly2vec: Polymorphic fourier-based encoding of geospatial objects for geoai applications. *Proc. Mach. Learn. Res.* **2024**, *267*, 55511.
25. Chu, C.; Shahabi, C. Geo2vec: Shape- and distance-aware neural representation of geospatial entities. *Proc. AAAI Conf. Artif. Intell.* **2025**, *40*, 18985–18993. [[CrossRef](#)]
26. Li, L.; Xue, H.; Song, Y.; Salim, F. 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*, Atlanta, GA, USA, 29 October–1 November 2024.
27. Cui, L.; Qian, H.; Xu, J.; Li, C.; Niu, X. Contrastive learning for one-shot building shape recognition using vector polygon transformers. *Geocarto Int.* **2025**, *40*, 2471087. [[CrossRef](#)]
28. Jiang, C.M.; Huang, J.; Tagliasacchi, A.; Guibas, L. Shapeflow: Learnable deformations among 3d shapes. *arXiv* **2021**, arXiv:2006.07982.
29. Sorkine, O.; Alexa, M. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*; SGP '07, Goslar, DEU; Eurographics Association: Eindhoven, The Netherlands, 2007; pp. 109–116.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.