Training-free Spatially Grounded Geometric Shape Encoding

Yuhang He
Microsoft Research
yuhanghe@microsoft.com
Code: https://github.com/yuhanghe01/XShapeEnc
Project: https://yuhanghe01.github.io/XShapeEnc-Proj/

Abstract

In this work, we introduce a novel training-free encoding framework, dubbed **XShapeEnc**, that encodes an arbitrary spatially grounded 2D geometric shape into a compact representation exhibiting five favorable properties, including invertibility, adaptivity, generality and controllability. Specifically, a 2D spatially grounded geometric shape is decomposed into its normalized geometry within the unit disk and its pose vector, where the pose is further transformed into a harmonic pose field that also lies within the unit disk. A set of orthogonal Zernike basis is constructed to encode shape geometry and pose either independently or jointly, with controllable relative emphasis on shape geometry or shape pose. We demonstrate the theoretical validity, efficiency, discriminability, and wide applicability of **XShapeEnc** via extensive analysis and experiments across a wide range of shape-aware tasks and our self-curated *XShapeCorpus* dataset. We envision **XShapeEnc** as a foundational framework for encoding tasks beyond one-dimensional discrete data and data-driven, learning-based paradigms, paving the way for unified and trained-free spatial encoding framework design.

1 Introduction

Discrete positional encoding has been thoroughly investigated since the emergence of Transformer network architectures [11, 20, 37, 42, 44, 49]. It explicitly embeds discrete positions indexed by coordinates into a high-dimensional latent space, by which the follow-up deep neural network can be successfully grounded on the input discrete position. Serving as a standard practice, discrete positional encoding has been widely applied in position-sensitive tasks such as natural language processing [49], image patch tokenization in computer vision [11], microphone position encoding in acoustics [18] and point's coordinate encoding in 3D point clouds [42]. Apart from point-wise position, 2D spatial geometric shapes often emerge as an integral object of interest that either deep neural networks or other methods need to reason about. Although there has been thorough discussion of discrete position encoding, research on unified 2D spatial geometric shape encoding has lagged far behind. *How to encode an arbitrary 2D spatial geometric shape into a compact representation that is both structure-preserving, spatially sensitive and neural-friendly awaits thorough investigation.*

We argue that this under-exploration stems from two factors. First, 2D shapes are often treated as ordinary images: with the success of CNNs, ViTs, generative models [11, 17, 54], and classical image features [31, 51, 53], shapes are commonly rasterized and processed as pixels, which reduces the incentive to study shape encoding itself. Second, existing methods frequently entangle shape encoding with downstream tasks [15, 19, 43, 55], either by focusing on restricted shape families such as polygons or sectors, or by learning task-specific representations that transfer poorly. This issue is especially evident in recognition-oriented settings [4, 6, 12], where spatial position and fine structural variation are often intentionally discarded. As spatial intelligence gains importance [9], applications

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Desirable properties comparison between potential 2D geometric shape encodings.

Method	arbitrary shape?	high frequency?	training-free?	task-agnostic?	spatial-context ?
AngularSweep [15, 19]	✗	✗	✓	✗	✓
Poly2Vec [43]	✗	✗	✗	✗	✓
Space2Vec [29]	✗	✗	✗	✗	✓
DeepSDF [35]	✓	✓	✗	✓	✗
2DPE [49]	✗	✗	✗	✓	✓
ShapeEmbed [40]	✓	✓	✗	✓	✗
ShapeDist [33]	✓	✗	✓	✓	✗
XShapeEnc (Ours)	✓	✓	✓	✓	✓

such as target region control and region-guided image generation [13, 15, 19, 22, 54] increasingly require unified, spatially grounded shape representations. As shown in Table 1, existing methods differ substantially in what they encode, which hinders systematic comparison and slows progress in geometric shape modeling.

Motivated by the aforementioned discussion, we seek a unified 2D spatially grounded geometric shape encoding framework that exhibits five main desirable properties:

XShapeEnc: Five Main Desirable Properties

1. **General-Purpose and Training-Free**: encoding is task-agnostic and involves no training.
2. **Invertibility and Interpretability**: encoding is invertible in principle and interpretable.
3. **Controllability**: flexibly allows controllable balance between shape geometry and pose.
4. **Generality and Efficiency**: encode arbitrary 2D shapes and is highly efficient.
5. **Adaptivity**: the encoding strategy can be flexibly tuned to fit various practical needs.

In this work, we introduce *XShapeEnc*, a framework that intrinsically preserves the aforementioned five desirable properties. Drawing inspiration from classical functional approximation theory and recent advances in frequency-rich positional encodings [44, 49], *XShapeEnc* is built upon orthogonal Zernike basis [50] defined over the unit disk, enabling the separate yet consistent encoding of both shape geometry and shape pose. Specifically, an arbitrary spatially grounded 2D geometric shape is decomposed into its normalized geometry within the unit disk and its spatial pose that specifies its position. By projecting the normalized shape geometry onto the orthogonal Zernike basis and optionally applying frequency propagation, we obtain a geometry encoding that is fully interpretable, efficient, and general-purpose. Meanwhile, the shape pose is represented as a harmonic pose field within the same unit disk so as to be encodable by Zernike basis. A controllable shape geometry and shape pose joint encoding strategy is introduced to enable adjustable emphasis on geometry or pose. The entire encoding process is task-agnostic, training-free and invertible in principle.

We demonstrate how *XShapeEnc* fulfills the five desirable properties outlined above through: 1. rigorous mathematical derivation of the whole encoding process; 2. in-depth analysis on encoding discriminability regarding both shape geometry and pose encoding; 3. extensive experiments across various downstream tasks, including shape retrieval [26, 27], inter-shape topological relation classification [28, 43] and spatial target region control [19].

2 Related Work

Geometric Shape Related Task. Geometric shape-related task can be divided into two main categories depending on if they require shape's spatial context, where spatial context indicates a shape's spatial position, orientation and scaling. Most existing works on geometric shape focus on shape recognition [4, 6, 12, 25, 40] in which the spatial context has been ignored. They tend to learn inter-class discriminative and intra-class consistent shape encoding. On the contrary, another line of work explicitly accounts for the shape's spatial context. Should the spatial position, orientation, or scale of a geometric shape changes, its encoded shape representation changes accordingly. Such spatially grounded geometric shape encoding has been preliminarily explored in recent years within the broader context of spatial intelligence [9, 15, 19, 30, 43, 48, 54–56]. However, these approaches are either tailored to specific tasks [15, 19, 22, 54] or workable for special geometric shapes [15, 19, 22, 43, 55, 56], intrinsically constraining the generalization and applicability of their shape encoding

in handling arbitrary spatially grounded geometric shapes. Such diverse shape encoding requirements set by various tasks, as well as the intimate entanglement of 2D geometric shape encoding and specific tasks, results in the lack of thorough and independent investigation on geometric shape encoding. Our proposed *XShapeEnc* is capable of encoding arbitrary spatially grounded geometric shapes within a unified framework, it can be further modified to cater for different encoding requirements.

Training-Required Geometric Shape Encoding. Most existing work dealing with 2D geometric shapes, whether they are spatially grounded or not, rely on deep neural network to learn shape representation. Depending on if they entangle the learning process with the downstream task, they can be classified into two main categories. The methods in the first category learn shape representation in latent space for each shape separately [7, 14, 32, 35, 46] or in self-supervised manner [40]. Involving no task centered training, they usually use shape itself as supervision signal. For example, the signed distance field (SDF) family methods (*e.g.*, DeepSDF [35], OccuNet [32]) associate each individual shape with a learnable latent representation, which is further fed to a shared decoder network. The latent representation is optimized by predicting the shape decision boundary. Generative modeling methods [7, 14, 46] learn the shape representation by generating the geometric shape or shape surface. In recent years, autoencoder has been leveraged to learn shape representation in self-supervised manner [3, 40]. The methods [4, 6, 15, 19] in the second category tightly couple the geometric shape encoding with the specific downstream task, in which the shape is implicitly learned and tailored for a specific task. These training-based encoding methods often require massive training datasets and are computationally expensive, their adopted neural networks need to be carefully designed to fit for various training setting. On the contrary, our proposed *XShapeEnc* is totally training-free and general-purpose. It provides shape encoding that is fully interpretable, invertible and frequency-rich, potentially serving as expressive representation for followup neural network learning.

Training-Free Geometric Shape Encoding. Training-free geometric shape encoding has rich history in both image processing and statistic modeling domain. The prominent shape relevant feature such as curvature, perimeter and contour are widely used to represent a geometric shape [5, 31, 33, 53]. For example, we can use points sampled along a geometric shape's contour to represent a geometric shape. Distance matrix [3, 40] or elliptical Fourier transform (EFT [34, 36] can further applied on top of the contour points to extract more advanced shape representation. In addition to contour points, statistical modeling methods [33] sample points across the whole shape area, then extract statistical feature from sampled points like pairwise distance. In parallel, decomposition methods are often used to approximate a 2D geometric shape by a set of basis. By projecting a 2D geometric shape onto each element of the basis, we can take the projection coefficients as the shape encoding. Typical decomposition includes elliptical Fourier transform [34, 36], Zernike basis [23, 50], Legendre and Chebyshev polynomials. In this work, we depend on Zernike basis for both shape geometry and shape pose encoding. As Zernike basis are mutually orthogonal and directly operate on shape area, *XShapeEnc* naturally obtains compact shape representation and unifies shape geometry and shape pose encoding within the same framework.

Another line of works integrate multiple training-free strategies or combine training-free and training-required encoding strategy together [3, 15, 19, 40, 43, 55] to encode a geometric shape. For example, based on the contour points, SoundTRC [19] applies positional encoding [49] to each point to get the shape encoding, ShapeEmbed [40] combines contour points extracted distance matrix and variational autoencoder (VAE) to learn shape encoding, Poly2Vec [43] combines Fourier transform and multiple layer perceptron (MLP). Although promising results, these methods lay strong assumptions on the type of geometric shapes (*e.g.*, convex [15, 19], connected without holes [3, 15, 19, 40], polygons [43]). Such strong assumption and the sophisticated encoding strategy internally inhibit from extending these encoding strategy to arbitrary shape and general-purpose geometric shape encoding. *XShapeEnc* overcomes all these obstacles, it unifies shape geometry and shape pose encoding within the same framework without requiring any training process.

3 XShapeEnc: General 2D Geometric Shape Encoding

3.1 Problem Definition

We seek a general geometric shape encoding framework $\mathcal{F}$ that is capable of encoding an arbitrary spatially grounded 2D geometric shape [1] $S \subset \mathbb{R}^2$ into high-dimensional compact representation

[1] In this work, the "shape" denotes a planar region with well-defined interior that admits a finite-resolution binary mask representation within a bounded domain, allowing digital rasterization and computational processing.

$Z \in \mathbb{R}^L$: $Z = \boldsymbol{\mathcal{F}}(S)$. Since the 2D shape is spatially grounded, we decompose the shape S into two orthogonal components: the normalized within-unit-disk shape geometry $S_\mathcal{G}$ that describes the shape's geometric structure (*e.g.*, contour, edge, etc.), and the shape pose $S_\mathcal{P}$ that describes the shape's spatial position (*e.g.*, scaling, translation, etc.), $S = [S_\mathcal{G}, S_\mathcal{P}]$. Specifically, the shape geometry $S_\mathcal{G}$ is represented by a 2D binary mask in the polar coordinate system $f_\mathcal{G}(r, \theta)$ ($S_\mathcal{G} \coloneqq f_\mathcal{G}(r, \theta)$), while the shape pose consists of 2×2 linear transformation matrix A and a 2D translation vector b, $S_\mathcal{P} = [A, b]$. By flattening the transformation A and translation b into a one-dimensional vector, we get the shape pose vector $\mathbf{p} \in \mathbb{R}^K$ ($S_\mathcal{P} \coloneqq \mathbf{p}$). Taking the shape geometry $S_\mathcal{G}$ and shape pose $S_\mathcal{P}$ as input, *XShapeEnc* is capable of encoding shape geometry and shape pose within a unified framework either independently or jointly (see Fig. 9 for pipeline visualization). Let $\boldsymbol{\mathcal{F}} = \{\boldsymbol{\mathcal{F}}_\mathcal{G}, \boldsymbol{\mathcal{F}}_\mathcal{P}, \boldsymbol{\mathcal{F}}_{\mathcal{GP}}\}$,

$$Z = \{Z_\mathcal{G}, Z_\mathcal{P}, Z_{\mathcal{GP}}\};\ Z_\mathcal{G} = \boldsymbol{\mathcal{F}}_\mathcal{G}(S_\mathcal{G}),\ Z_\mathcal{P} = \boldsymbol{\mathcal{F}}_\mathcal{P}(S_\mathcal{P}), Z_{\mathcal{GP}} = \boldsymbol{\mathcal{F}}_{\mathcal{GP}}(S_\mathcal{G}, S_\mathcal{P}); \tag{1}$$

where $\boldsymbol{\mathcal{F}}_\mathcal{G}$, $\boldsymbol{\mathcal{F}}_\mathcal{P}$ and $\boldsymbol{\mathcal{F}}_{\mathcal{GP}}$ are the encoding frameworks for shape geometry encoding, shape pose encoding and joint shape geometry and pose encoding, respectively.

3.2 Zernike Basis Introduction

To encode shape geometry and shape pose in Eqn. (1) in a principled and compact representation, we adopt the Zernike basis as the encoding basis. Frits introduced by Frist Zernike in the 1930s [50] for optical wavefront analysis, Zernike basis forms a set of complete and orthogonal basis over the unit disk and is defined in a polar coordinate system (r, θ).

In the unit disk defined in polar coordinate system $\mathbb{D} = \{(r, \theta) \mid 0 \leq r \leq 1; 0 \leq \theta \leq 2\pi\}$, given a radial order n and angular frequency band m in $\mathbb{D}$, the complex Zernike basis $V_n^m(r, \theta)$ is the multiplication of radial polynomial $R_n^{|m|}(r)$ and angular frequency $e^{im\theta}$. While the radial polynomial encodes the radial variation, the angular frequency captures the angular oscillation,

$$V_n^m(r, \theta) = R_n^{|m|}(r) \cdot e^{im\theta} \tag{2}$$

where the radial polynomial $R_n^{|m|}(r)$ is defined as,

$$R_n^{|m|}(r) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+|m|}{2} - s\right)!\left(\frac{n-|m|}{2} - s\right)!} r^{n-2s} \tag{3}$$

The Zernike basis $V_n^m(r, \theta)$ is in the complex domain $V_n^m(r, \theta) \in \mathbb{C}$, while $R_n^{|m|}(r)$ is in the real domain $R_n^{|m|}(r) \in \mathbb{R}$. By constraining $n - |m|$ to be even and $|m| < n$, the constructed Zernike bases are mutually orthogonal over the unit disk with respect to the area measure $r d_r d_\theta$,

$$\int_0^1 \int_0^{2\pi} V_n^m(r, \theta) \left(V_{n'}^{m'}(r, \theta)\right)^* r\, d\theta\, dr = \frac{\pi}{n+1} \delta_{nn'} \delta_{mm'} \tag{4}$$

where $\delta_{mm'}$ and $\delta_{nn'}$ are Kronecker deltas. $\delta_{mm'} = 0$ if $m \neq m'$, otherwise 1. $\delta_{nn'} = 0$ if $n \neq n'$, otherwise 1. By selecting a set of radial order n and angular frequency m by following the $n - |m|$ is even and $|m| < n$ constraints, we can construct a Zernike basis set $\{V_n^m(r, \theta) | n = 0, 1, \cdots, N, m = -M, -M+2, \cdots, M\}$ (in Eqn. (2)). We visualize part of Zernike basis in Fig. 1, from which we can observe that the basis governed by different n and m exhibits different frequency variation along either the radial or angular direction. We exploit this characteristic to encode a 2D shape at different granularities. More discussion and mathematical proof are given in Sec. A.3 and Sec. A.16 of Appendix.

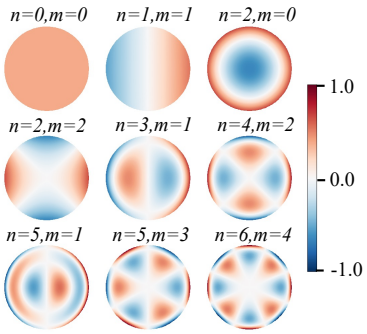


Figure 1: Zernike basis visualization, we visualize nine real-part basis governed by radial mode n and angular frequency m.

3.3 Shape Geometry Encoding

Given the shape geometry $S_{\mathcal{G}}$ expressed as a binary mask in polar coordinate system: $f_{\mathcal{G}}(r, \theta)$, we project it onto the Zernike basis set $\{V_n^m\}$ and treat the projection coefficients as the shape geometry encoding. Specifically, for the Zernike basis $V_n^m(r, \theta)$, the projection coefficient z_n^m is obtained by integrating the elementwise multiplication of the shape mask $f_{\mathcal{G}}(r, \theta)$ by $V_n^m(r, \theta)$ over the unit disk,

$$z_n^m = \frac{n+1}{\pi} \iint_{\mathbb{D}} f_{\mathcal{G}}(r, \theta) \, [V_n^m(r, \theta)]^* \, r \, dr \, d\theta; \quad z_n^m \in \mathbb{C} \tag{5}$$

Different Zernike basis V_n^m describes the shape geometry at different granularities: low-order coefficients describe the global outline, while higher-order coefficients encode localized and subtle details. By projecting onto Zernike basis, we obtain the shape geometry encoding $Z_{\mathcal{G}}$,

$$Z_{\mathcal{G}} = [z_0^0, \, z_1^{-1}, \, z_1^1, \, z_2^{-2}, \, \cdots, \, z_n^m, \, \cdots, \, z_N^M], \quad \text{where} \quad z_n^m \in \mathbb{C}, \; m \leq M, n \leq N \tag{6}$$

where $M \leq N, \quad N - M$ is even. The total number of complex coefficients in Eqn. (6) equals $\frac{N^2+M}{2} + 1$. Zernike basis based shape geometry encoding has two desirable properties that make the encoding in Eqn. (6) highly adaptive for various needs: **Linearity**, which means Zernike basis based encoding respects addition and scalar multiplication (Sec. A.17 of Appendix). **Rotation Equivariance**, which means rotating the shape geometry equals to rotating the final Zernike basis.

3.4 High-Frequency Enrichment

The encoding in Eqn. (6) is compact and invertible, but may suffer from high-frequency sparsity for simple shape geometries. For instance, the encoding for a circle only activates the lowest-order mode (*e.g.*, Z_0^0), resulting in the encoding dominated by 0. Such low-frequency-only encoding lacks frequency richness and is suboptimal for neural network learning [45, 52]. To resolve this challenge, we propose Frequency Propagation (*FreqProp*) that explicitly injects structural information informed by lower-frequency encoding to its adjacent higher-frequency encoding. Specifically, we introduce two kinds of frequency propagation strategies: radial frequency propagation (*rFreqProp*) and angular frequency propagation (*aFreqProp*). *rFreqProp* enriches each coefficient z_n^m by incorporating magnitude and phase information from its radial adjacent lower-frequency coefficient Z_{n-2}^m. *aFreqProp* enriches each coefficient z_n^m by incorporating magnitude and phase information from its angular adjacent lower-frequency coefficient Z_n^{m-2},

$$z_n^m \leftarrow z_n^m + \lambda_r |z_{n-2}^m| \cdot e^{i \arg(Z_{n-2}^m)}; \quad z_n^m \leftarrow z_n^m + \lambda_a |z_n^{m-2}| \cdot e^{i \arg(Z_n^{m-2})} \tag{7}$$

where $\arg(\cdot)$ indicates the phase information, λ_r is the radial propagation ratio deciding the frequency ratio propagating from lower-frequency coefficient (we set it as 0.6, see Sec. A.20 in Appendix), λ_a is the corresponding angular propagation ratio. As Zernike indices obey parity, stepping n by 2 (or m by 2) guarantees to land on valid Zernike index. Starting from the lower-frequency coefficient, we iteratively propagate the its frequency to the radial or angular higher-frequency coefficient by the chain rule in Eqn. (7). The propagation ratio ensures the long-range exponential frequency influence decay. As is shown in Fig. 2, a coefficient mostly impacts its adjacent higher frequency coefficient, the longer-range leads to exponentially decayed impact. This long range decay protects the encoding from contamination by frequency propagation and maintains the encoding discriminability. The whole propagation process is shown in Fig. 10.

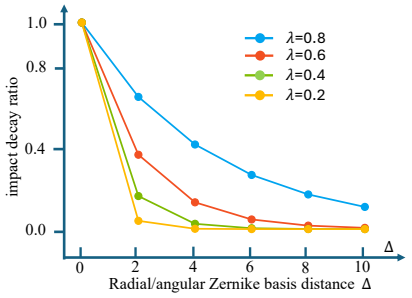


Figure 2: Frequency impact decay illustration in *FreqProp*. We show the impact decay ratio w.r.t. radial/angular basis distance Δ_n/Δ_m, under the different propagation ratio λ.

Our proposed *FreqProp* in Eqn. (7) has clear geometric interpretation. As is shown in Fig. 11, it can be viewed as we perturb the higher-order Zernike basis Z_n^m by *rotating* it by $\arg(Z_{n-2}^m)$ and *scaling* it by Z_n^m. The perturbed basis is then "projected" back

to itself to yield a non-zero coefficient as long as the lower-frequency is non-zero. Benefiting from the *Linearity* and *Rotation Equivariance* property of Zernike transform, we can directly obtain the propagated coefficient without rotating/scaling the Zernike basis in reality. Given the initial encoding in Eqn. (6). It fully accommodates the shape geometry's specificity as the propagation builds on top of the initial encoding. *FreqProp* naturally preserves the orthogonality of the Zernike basis as no Zernike basis is altered during the propagation process.

Invertibility. *FreqProp* is fully invertible, the original encoding coefficients can be precisely recovered by subtracting the propagated term, where the propagation term can be deterministically decided by starting from Z_0^0. With the recovered original coefficients, we can further recover the original shape geometry by $S_{\mathcal{G}} = \sum_{(n,m)\in(N,M)} Z_n^m \cdot V_n^m$.

3.5 Shape Pose Encoding

For the pose vector $\mathbf{p} \in \mathbb{R}^K$, we seek a pose encoding strategy that packs multiple scalar "pose" parameters into compact representation under the same Zernike basis. To this end, we define a harmonic pose field $f_{\mathcal{P}}(r, \theta; m_p, \mathbf{p})$ over the unit disk (r, θ) at the angular frequency m_p,

$$f_{\mathcal{P}}(r, \theta; m_p, \mathbf{p}) = \Big(\sum_{k=1}^{K} p_k w_k(r)\Big) \cos(m_p \theta) \tag{8}$$

where $p_k \in [-1, 1]$, $p_k \in \mathbf{p}$. For the sake of numerical stability, $\mathbf{p}$ is L_2-normalized. Each pose parameter p_k is associated with a separate radial window $w_k(r)$, which defines how to place pose parameter energy along the radial axis. $\cos(m_p\theta)$ is the angular harmonic, we directly instantiate it as the real part of Zernike basis $e^{im\theta}$ (the imaginary part is ignored). By projecting the harmonic pose field to Zernike basis $V_n^m(r, \theta)$, we only get non-zero projection coefficient $a_n^m = \langle f_{\mathcal{P}}(r, \theta, m_p), V_n^m(r, \theta)\rangle$ at Zernike basis whose angular frequency equals to m_p (The detailed proof is presented in Sec. A.19 Appendix),

$$a_n^m = \begin{cases} \pi\sqrt{\frac{n+1}{\pi}} \sum_{k=1}^{K} p_k \int_0^1 w_k(r)\, V_n^{|m|}(r)\, r\, dr, & m = \pm m_p, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Let $C_{n,m_p}^k = \int_0^1 w_k(r)\, R_n^{|m_p|}(r)\, r\, dr$, we stack C_{n,m_p}^k to get $\mathbf{C}$, stack $a_n^{m_p}$ to get $\mathbf{A}$, Eqn. (9) can re-written as,

$$\mathbf{A} = \mathbf{p} \cdot \mathbf{C}, \quad \text{where } \mathbf{A} \in \mathbb{R}^{1\times L}, \mathbf{p} \in \mathbb{R}^{1\times K}, C \in \mathbf{R}^{K\times L} \tag{10}$$

To ensure invertibility and robustness of $\mathbf{p}$, $\mathbf{C}$ has to be full rank $\texttt{rank}(\mathbf{C}) = K$ and well-conditioned so that $\mathbf{p} = \mathbf{A}\mathbf{C}^{-1}$. To this end, we add two constraints to Eqn. (8): First, $K \le L$, which is the prerequisite for $\mathbf{C}$ to be full-rank. In practice, we simply need to project the harmonic pose field to at least K Zernike basis. Second, we instantiate $w_k(r)$ to be a set of *radially orthonormal* windows under the Zernike weight $r\, dr$ across different pose parameters, with which each pose parameter p_k only contributes to one distinct direction in the radial space, $\langle w_i, w_j\rangle = \int_0^1 w_i(r) w_j(r)\, r\, d_r = \delta_{ij}$. Radially orthogonal windows make $\mathbf{C}$ well-conditioned and often diagonal after projection, thus ensure the invertibility of $\mathbf{p}$. To explicitly introduce both low and high frequency along the radial direction, we implement each radial window w_k to be two Gaussian bumps. The mutual radial orthonormal property are ensured by Gram-Schmidt based radial window generation (see Fig. 3). The final shape pose encoding can be represented as,

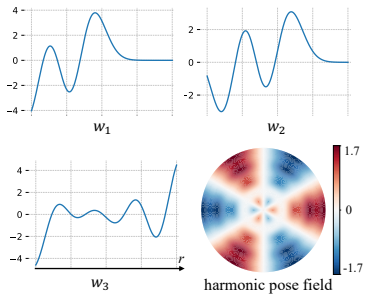


Figure 3: Three orthonormal radial windows and the harmonic pose field visualization.

$$Z_{\mathcal{P}} = [a_n^{m_p},\ a_{n+2}^{m_p},\ a_{n+4}^{m_p},\ \cdots,\ a_N^{m_p}],\ \ a_n^{m_p} \in \mathbb{R} \tag{11}$$

where $n = |m_p|,\ n < N,\ \ N - n$ is even. The proposed harmonic pose encoding is linear w.r.t. the pose field and Zernike projection (see Sec. A.22 of Appendix).

Pose Invertibility. Once we ensure $K \leq L$ and $w_k(r)$ is instantiated as a set of radially orthonormal windows, the original pose vector can be precisely recovered by Eqn. (10).

3.6 Shape Geometry and Pose Joint Encoding

Our key insight is that geometry and pose exhibit fundamentally different spectral behaviors: geometry energy is broadly distributed across Zernike basis, while pose energy is concentrated within a small set of angular frequency bands. To reconcile this mismatch, we embed pose as a phase modulation of geometry coefficients, which preserves geometric interpretability and avoids scale interference.

To ensure that shape pose admits non-zero projection onto the Zernike basis, we construct a harmonic pose field for each angular frequency band $m_p \in M$. By superposing these harmonic pose fields and combining them with the geometry mask, we form a composite geometry–pose field,

$$f_{\mathcal{GP}}(r,\theta) = \Big[f_{\mathcal{G}}(r,\theta),\ \sum_{m_p \in M} f_{\mathcal{P}}(r,\theta;m_p,\mathbf{p})\Big] \tag{12}$$

where $f_{\mathcal{G}}(r,\theta)$ denotes the shape geometry mask, $f_{\mathcal{P}}(r,\theta;m_p,\mathbf{p})$ denotes the harmonic pose field associated with angular band m_p. Projecting the geometry mask onto the Zernike basis yields complex-valued geometry coefficients, while projecting the composite pose field yields real-valued pose coefficients for the same basis. Consequently, each Zernike basis V_n^m is associated with a pair of coefficients: (z_n^m, a_n^m). We interpret the real-valued pose coefficient a_n^m as the rotation angle acting on the complex-valued geometry coefficient z_n^m and perform joint encoding via phase modulation,

$$Z_{\mathcal{GP}} = \Big[z_0^0 \cdot e^{-ia_0^0}, \ldots, z_n^m \cdot e^{-ia_n^m}, \ldots, z_N^M \cdot e^{-ia_N^M}\Big] \tag{13}$$

where $z_n^m \in \mathbb{C},\ a_n^m \in \mathbb{R}$. This formulation aligns the joint encoding to the geometry coefficient scale while embedding pose information purely in the phase. Importantly, this operation preserves the coefficient magnitude and is consistent with the rotational action of Zernike basis, yielding a geometrically interpretable and invertible fusion.

Beyond the default joint encoding, we introduce a tunable emphasis mechanism that allows controlled bias toward either geometry or pose. We define a relative emphasis parameter $\beta \in [0,2]$, where $\beta = 1$ indicates neutral emphasis, $\beta < 1$ emphasizes pose, and $\beta > 1$ emphasizes geometry. The farther β is from 1, the stronger the emphasis. Intuitively, emphasizing pose corresponds to degenerating geometry coefficients toward a unit complex carrier, such that pose-induced rotations dominate. Conversely, emphasizing geometry suppresses pose-induced rotations. This yields the formulation,

$$Z_{\mathcal{GP}} = \begin{cases} \Big[e^{\beta \cdot \ln(z_0^0)} \cdot e^{-i \cdot a_0^0}, \ldots, e^{\beta \cdot \ln(z_N^M)} \cdot e^{-i \cdot a_N^M}\Big], & \beta \in [0,1) \quad \leftarrow \text{emphasize pose} \\ \Big[z_0^0 \cdot e^{-i \cdot \eta(\beta) a_0^0}, \ldots, z_N^M \cdot e^{-i \cdot \eta(\beta) a_N^M}\Big], & \beta \in (1,2] \quad \leftarrow \text{emphasize geometry} \end{cases} \tag{14}$$

where $\eta(\beta) = \exp(-5 \cdot (\beta - 1))$ smoothly suppresses pose influence as β increases. $\beta \in [0,1]$ emphasizes shape pose, while $\beta \in (1,2]$ emphasizes shape geometry. We show how the relative emphasis is achieved in Fig. 4, from which we can observe that, ranging from 0.0 to 2.0, β continuously assigns different weight to shape geometry and shape pose respectively. Another qualitative example is shown in Fig. 5, in which we choose 4 exemplar shape geometries from *XShapeCorpus* with depth $d = 2$, and place each of them at the 4 sub-areas independently. As a result, each shape geometry has been duplicated at the 4 sub-areas with different shape poses, leading to a total of 16 spatially grounded shape geometries. By varying the modulation weight β in Eqn. (14), we can obtain the jointly encoding with different emphasis between shape geometry and shape pose. We test the discriminability by

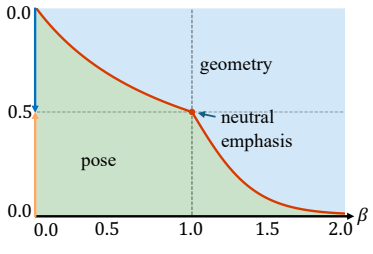
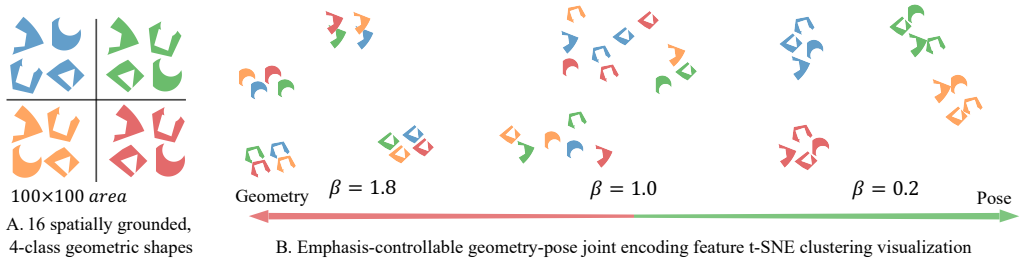


Figure 4: β controlled geometry-pose emphasis visualization.



A. 16 spatially grounded, 4-class geometric shapes

B. Emphasis-controllable geometry-pose joint encoding feature t-SNE clustering visualization

Figure 5: *XShapeEnc* controllable shape geometry-pose joint encoding visualization. By varying the emphasis modulation parameter β, we can emphasize either shape geometry or shape pose.

running t-SNE [47] clustering on the joint encoding under different modulation weight β to test if the joint encoding reflects the emphasis controlled by β. We can clearly see from Fig. 5 that we can successfully emphasize the shape geometry and shape pose by controlling β. While $\beta = 1$ strikes neutral emphasis where each geometric shape is individually separated, smaller $\beta = 0.2$ and larger $\beta = 1.8$ cluster them by shape pose and shape geometry correspondingly. Another comparison with two other post-hoc joint encoding is given in Fig. 12 in Sec. A.7 of Appendix.

Invertibility. Eqn. (14) entangles shape geometry and shape pose, we cannot recover any of them directly from the joint encoding, unless their own encoding is stored separately.

3.7 Encoding Recapitulation

The *XShapeEnc* workflow is shown in Algorithm 1 of Appendix. It supports shape geometry and shape pose encoding either separately or jointly. The Zernike basis set is precomputed beforehand. The projection of shape geometry or the harmonic pose field onto the Zernike basis can be accelerated through vectorized computation, while the optional *FreqProp* only requires a one-time sweep. Therefore, the overall encoding process is efficient. Since the shape geometry encoding in Eqn. (6) and the shape geometry-pose joint encoding in Eqn. (14) are complex-valued, we show how to convert them into compact real-valued encodings in Sec. A.4 of Appendix.

4 Experiment

We evaluate *XShapeEnc* from 3 main aspects: 1. **Theoretical Validity:** if the desired encoding properties such as invertibility, interpretability and generality are theoretically guaranteed? 2. **Discriminability:** if *XShapeEnc* is discriminative enough for shape geometry and shape pose encoding? 3. **Applicability:** if the encoding exhibit favorable applicability for various downstream tasks?

To test the capability of *XShapeEnc* in encoding arbitrary shape geometries, we curate a new 2D shape corpus *XShapeCorpus*, it contains 1000 complex shape geometries with hierarchical shape complexity denoted by construction depth (see Sec. A.8 of Appendix).

4.1 Comparing Baselines and XShapeEnc Implementation

We exhaustively compare *XShapeEnc* with 11 baselines that can be categorized into four main classes: boundary point based encoding [1, 19, 33, 49], boundary line based encoding [43], regular sampled point based encoding [29, 40] and neural network based encoding [11, 17, 38]. We briefly show them in Table 2, detailed discussion is deferred to Sec. A.9 and Fig. 14 of Appendix.

XShapeEnc Implementation: For *XShapeEnc*, the shape geometry mask and harmonic pose field are discretized into 300×300 resolution. *FreqProp* ratio $\lambda = 0.6$. The source code is in the supplementary material.

Table 2: Comparing Baseline Summary.

Category	Method
Boundary Point based	AngularSweep [19] 2DPE [49] ShapeDist [33] ShapeContexts [1]
Boundary Line	Poly2Vec [43]
Regular Sampled Points based	PointSet ShapeEmbed [40] Space2Vec [29]
Neural Network based	ResNet18 [17] ViT [11], CLIP [38]



Figure 6: *XShapeEnc* encoding invertibility illustration. We show two complex shape geometry (within unit disk) with depth = 6 and depth = 8 and their reconstructed shape geometry under various encoding length. More visualization is given in Fig. 17 of Appendix.

4.2 XShapeEnc Encoding Theoretical Validity

XShapeEnc is mathematically grounded. For *shape geometry*, the unit-disk mask is projected onto orthogonal Zernike basis (Sec. 3.3), whose orthogonality and linearity are proved in Sec. A.16 and Sec. A.17 of Appendix. This makes the encoding stable and interpretable, while also yielding rotation equivariance; taking coefficient magnitudes further gives rotation invariance. For *shape pose*, we construct a harmonic pose field (Eqn. 8), whose Zernike projection is derived in Sec. A.22 of the Appendix. With radially orthonormal windows and $K \leq L$, the pose can be recovered invertibly. Together, these results show that the full encoding is mathematically principled.

Shape Geometry Encoding Invertibility. We encode all the shape geometries in *XShapeCorpus* with encoding length of 64, 128, 256, 512, 1024, 2048, 4096, then further run inversion process to reconstruct the shape geometry. Two sets of reconstructed shape geometries w.r.t different encoding length in Fig. 6, from which we can clearly see that longer encoding length results in higher reconstruction accuracy. The quantitative reconstruction mean square error (MSE) under various encoding length is shown in Fig. 18 of Appendix, which shows longer encoding length corresponds to smaller MSE value and more complex shape geometry leads to larger MSE value. This observation is consistent with *XShapeEnc* encoding principle where the input shape geometry is essentially approximated by a set of orthogonal Zernike basis. More complex shape geometry requires more bases, and encoding length cutoff inevitably leads to higher reconstruction error (see Sec. A.11).

Shape Pose Encoding Invertibility. According to Sec. 10 and Eqn. (10), the pose can be precisely recovered as long as the encoding length is larger than the pose vector length.

4.3 XShapeEnc Encoding Discriminability

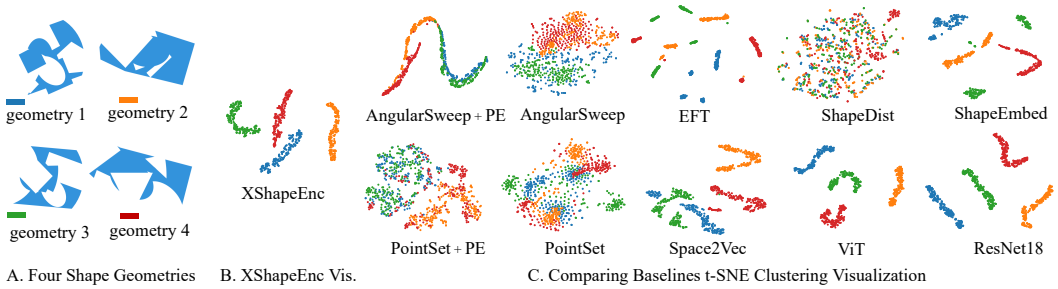


Figure 7: Shape geometry encoding clustering visualization between *XShapeEnc* and other baselines.

To assess shape geometry discriminability, we select four complex shapes with depth $= 10$ from *XShapeCorpus* and generate 200 augmented variants for each using rotation, shearing, vertex jitter, and elastic deformation. We then encode these samples with *XShapeEnc* and competing baselines using 512-d features, and visualize the resulting embeddings with t-SNE [47]. As shown in Fig. 7, *XShapeEnc* forms four compact and well-separated clusters, indicating strong intra-class cohesion and clear inter-class margins even under heavy perturbation. In contrast, classical training-free descriptors produce scattered and overlapping manifolds. Despite using no learning or external data, *XShapeEnc* achieves separability comparable to pretrained ResNet18 [17] and ViT [11], underscoring its strong

inductive bias and expressive power for shape geometry encoding. *XShapeEnc* shape pose encoding discriminability is given in Sec. A.12 and Fig. 19 of Appendix.

4.4 XShapeEnc Encoding Applicability

Table 3: 2D shape retrieval mAP result on Mendeley dataset [26] and MPEG-7 dataset [27].

Method Name	Mendeley	MPEG-7
PointSet	0.21	0.10
ShapeContexts [1]	0.55	0.45
Space2Vec [29]	0.30	0.47
ShapeDist [33]	0.25	0.11
AngularSweep [19]	0.42	0.26
ShapeEmbed [40]	0.73	0.41
ResNet18 [17]	0.40	0.56
ViT [11]	0.43	0.57
CLIP [38]	0.57	**0.64**
Plain Zernike	0.87	0.53
XShapeEnc	**0.91**	0.59

Table 4: Inter-shape topological relation classification accuracy (↑).

Method	Singapore	New York
PointSet	0.670	0.564
ShapeContexts [1]	0.581	0.525
AngularSweep [19]	0.606	0.546
Space2Vec [29]	0.706	0.632
ResNet18 [17]	0.674	0.753
ViT [11]	0.669	0.752
CLIP [38]	0.700	**0.779**
Poly2Vec [43]	0.702	0.684
XShapeEnc (w/o *FreqProp*)	0.740	0.753
XShapeEnc	**0.760**	0.768

Shape Retrieval. We evaluate shape retrieval on the Mendeley 2D shape benchmark [26] and the MPEG-7 CE-Shape-1 Part B dataset [27]. Mendeley consists of 9 primitive shape classes such as Triangle, Square and Pentagon. Each class is instantiated with 10,000 shapes with large variations in scale, rotation and deformation. MPEG-7 dataset [27] contains 70 organic shape classes, ranging from animal shape (*e.g.*, cattle, beetle, camel, butterfly), people (*e.g.*, children, face) to utensils (*e.g.*, fork, spoon, jar). In the main paper, we report magnitude-based *XShapeEnc* without *FreqProp*, together with plain Zernike encoding. As shown in Table 3, *XShapeEnc* achieves the best mAP on Mendeley (0.91) and the second-best mAP on MPEG-7 (0.59), substantially outperforming most training-free baselines. These results suggest that *XShapeEnc* captures global shape structure robustly under deformation, rotation, and scale variation. Although pretrained vision models perform slightly better on MPEG-7, *XShapeEnc* remains highly competitive across datasets with very different shape distributions. More details are given in Sec. A.13 of the Appendix.

Inter-Shape Topological Relation Classification. We follow Poly2Vec [43] and evaluate on OpenStreetMap polygons from New York and Singapore [28], where each polygon pair exhibits one of five topological relations [8]: *disjoint*, *touch*, *equal*, *overlap*, or *within* (see Fig. 20 of Appendix). This task is a strong test of joint geometry-pose encoding because the relation depends on both shape geometry and shape pose. Following Poly2Vec [43] setting, we construct 15,000/3,000 training/test pairs per city and use a two-layer MLP classifier. As shown in Table 4, *XShapeEnc* achieves the best result on Singapore (0.760) and remains highly competitive on New York (0.768), outperforming Poly2Vec [43] and most other baselines. It thus highlights the superiority of *XShapeEnc* for geometry-pose joint encoding, and its *FreqProp* component for frequency enrichment. More detail is in Sec. A.14 of Appendix. More experiments are given in Sec. A.15.

4.5 Conclusion and Limitation

We present a training-free framework, *XShapeEnc*, for encoding arbitrary spatially grounded 2D shapes. The core idea is to decompose it into normalized shape geometry and shape pose, then encode both with the same Zernike basis; frequency propagation (FreqProp) enriches the encoding that is suitable for neural network learning. We further enable geometry-pose joint encoding with adjustable emphasis. Experiments on both theoretical analysis and practical tasks demonstrate the mathematical rigor and wide applicability of *XShapeEnc*. We envision *XShapeEnc* as a foundational encoding framework whenever the 2D geometric shape is spatially grounded or the data is limited.

Limitation One limitation is the lack of testing *XShapeEnc* on wider range of tasks (*e.g.*, shape-aware generation, robotics area, embodied AI); Another limitation is that we assume the the 2D spatially grounded geometric shape has clear definition, the interference of noise has not been sufficiently discussed. Two future research directions: one is to incorporate semantic encoding (*e.g.*, texture, semantic label) into *XShapeEnc* encoding, the other one is to extend *XShapeEnc* to encode spatially grounded 3D object, and further 3D objects undergoing spatial motion.

Acknowledgement

We thank the reviewers for their constructive feedback and suggestions, which is important for us to improve this work.

References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape Matching and Object Recognition Using Shape Contexts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2002. 8, 10, 19, 22, 24

[2] Archana Boob and Mansi Radke. ElementaryCQT: A New Dataset and its Deep Learning Analysis for 2D Geometric Shape Recognition. In *SN Computer Science*, 2024. 18

[3] James Burgess, Jeffrey J. Nirschl, Maria-Clara Zanellati, Alejandro Lozano, Sarah Cohen, and Serena Yeung-Levy. Orientation-Invariant Autoencoders Learn Robust Representations For Shape Profiling of Cells and Organelles. In *Nature Communications*, 2024. 3

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1, 2, 3

[5] Shih-Fu Chang and John R. Smith. Extracting Multi-dimensional Signal Features for Content-based Visual Query. In *SPIE Symposium on Visual Communications and Signal Processing*, pages 995–1006, 1995. 3

[6] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3

[7] Zhiqin Chen and Hao Zhang. Learning Implicit Fields for Generative Shape Modeling. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[8] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *Advances in Spatial Databases*, 1993. 10, 22

[9] Andrew J. Davison. FutureMapping: The Computational Structure of Spatial AI Systems. *CoRR*, abs/1803.11288, 2018. 1, 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 19, 20, 22

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 8, 9, 10, 20, 22

[12] Yi Fang, Jin Xie, Guoxian Dai, Meng Wang, Fan Zhu, Tiantian Xu, and Edward Wong. 3D Deep Shape Descriptor. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[13] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[14] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[15] Rongzhi Gu and Yi Luo. ReZero: Region-Customizable Sound Extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 32:2576–2589, 2024. 1, 2, 3, 18, 23

[16] David Gutman, Noel C. F. Codella, Emre Celebi, Brian Helba, Michael Marchetti, Nabin Mishra, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1605.01397*, 2016. 24

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 8, 9, 10, 19, 22

[18] Yuhang He, Anoop Cherian, Gordon Wichern, and Andrew Markham. Deep Neural Room Acoustics Primitive. In *International Conference on Machine Learning (ICML)*, 2024. 1

[19] Yuhang He, Andrew Markham, and Okan Köpüklü. SoundTRC: DNN-based Acoustic Target Region Control. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025. 1, 2, 3, 8, 10, 18, 22, 23, 24, 27

[20] Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Ning Ding, Youbang Sun, Biqing Qi, Yuchen Fan, Xuekai Zhu, and Bowen Zhou. Fourier Position Embedding: Enhancing Attention's Periodic Extension for Length Generalization. In *International Conference on Machine Learning (ICML)*, 2025. 1

[21] Zihan Huang, Tao Wu, Wang Lin, Shengyu Zhang, Jingyuan Chen, and Fei Wu. AutoGeo: Automating Geometric Image Dataset Creation for Enhanced Geometry Understanding. *IEEE Transactions on Multimedia*, 27:3105–3116, 2025. 18

[22] Malek Itani, Tuochao Chen, Takuya Yoshioka, and Shyamnath Gollakota. Creating Speech Zones with Self-Distributing Acoustic Swarms. In *Nature Communications*, 2023. 2, 23

[23] A. Khotanzad and Y.H. Hong. Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1990. 3

[24] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 20, 24

[25] Ender Konukoglu, Ben Glocker, Antonio Criminisi, and Kilian M Pohl. WESD–Weighted Spectral Distance for measuring shape dissimilarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(9):2284–2297, 2013. 2

[26] Anas El Korchi and Youssef Ghanou. 2D Geometric Shapes Dataset–For Machine Learning and Pattern Recognition. *Data in Brief*, 32:106090, 2020. 2, 10, 18, 20, 22

[27] Longin Jan Latecki. Shape data for the mpeg-7 core experiment ce-shape-1, 2006. Dataset. 2, 10, 22

[28] Jae-Gil Lee and Minseo Kang. Geospatial Big Data: Challenges and Opportunities. In *Big Data Research*, 2015. 2, 10, 22

[29] Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-Scale Representation Learning for Spatial Feature Distributions using Grid Cells. In *International Conference on Learning Representations (ICLR)*, 2020. 2, 8, 10, 19, 22, 23, 24

[30] Gengchen Mai, Chiyu Jiang, Weiwei Sun, Rui Zhu, Yao Xuan, Ling Cai, Krzysztof Janowicz, Stefano Ermon, and Ni Lao. Towards General-Purpose Representation Learning of Polygonal Geometries. *GeoInformatica*, 27(2):289–340, 2023. 2

[31] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and Texture Analysis for Image Segmentation. In *International Journal of Computer Vision (IJCV)*, 2001. 1, 3

[32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[33] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape Distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002. 2, 3, 8, 10, 18, 19, 22, 24

[34] Frank P. Kuhl and Charles R. Giardina. Elliptic Fourier Features of a Closed Contour. *Computer Graphics and Image Processing*, 1982. 3

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[36] Eric Persoon and King-Sun Fu. Shape Discrimination Using Fourier Descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, 1977. 3, 24

[37] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations (ICLR)*, 2022. 1

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021. 8, 10, 20, 22, 23

[39] Richard Ralph and The Latecki Group. MPEG-7 Core Experiment CE-Shape-1 Test Set: Benchmarking Image Database for Shape Recognition Techniques, 1999. Accessed: 2025-12-05. 18

[40] Anna Foix Romero, Craig Russell, Alexander Krull, and Virginie Uhlmann. ShapeEmbed: A Self-Supervised Learning Framework for 2D Contour Quantification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2, 3, 8, 10, 19, 20, 22, 23, 24

[41] Robin Scheibler, Eric Bezzam, and Dokmanić Ivan. Pyroomacoustics: A python package for audio room simulations and array processing algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. 23

[42] Changyong Shu, Jiajun Deng, Fisher Yu, and Yifan Liu. 3DPPE: 3D Point Positional Encoding for Multi-Camera 3D Object Detection Transformers. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[43] Maria Despoina Siampou, Jialiang Li, John Krumm, Cyrus Shahabi, and Hua Lu. Poly2vec: Polymorphic Encoding of Geospatial Objects for Spatial Reasoning with Deep Neural Networks. *International Conference on Machine Learning (ICML)*, 2025. 1, 2, 3, 8, 10, 19, 22, 23, 24

[44] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. In *Neurocomputing*, 2024. 1, 2

[45] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 5

[46] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[47] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(86):2579–2605, 2008. 8, 9, 17, 20, 21, 28

[48] Rein van't Veer, Peter Bloem, and Erwin Folmer. Deep Learning for Classification Tasks on Geospatial Vector Polygons, 2019. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jacob Uszkoreit, Llion Jones Jone, Aidan N. Gomez, and Lukasz Kaiser. Attention is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, 3, 8, 18, 20, 21

[50] von F. Zernike. Beugungstheorie des Schneidenver-Fahrens Und Seiner Verbesserten form, der Phasenkontrastmethode. *Physica*, 1(7):689–704, 1934. 2, 3, 4

[51] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: a Line Segment Detector. In *Image Processing On Line (IPOL)*, 2012. 1

[52] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[53] Xinggang Wang, Bin Feng, Xiang Bai, Wenyu Liu, and Longin Jan Latecki. Bag of Contour Fragments for Robust Shape Classification. *Pattern Recognition*, 47(6):2116–2125, 2014. 1, 3

[54] Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. ReCo: Region-Controlled Text-to-Image Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[55] Dazhou Yu, Yuntong Hu, Yun Li, and Liang Zhao. PolygonGNN: Representation Learning for Polygonal Geometries with Heterogeneous Visibility Graph. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024. 1, 2, 3

[56] Huichao Zhang, Pengyu Wang, Manyi Li, Zuojun Li, and Yaguang Wu. Unit Region Encoding: A Unified and Compact Geometry-aware Representation for Floorplan Applications, 2025. 2

A Appendix

A.1 More Detail on XShapeEnc Encoding

Shape Pose Encoding. To verify that the proposed harmonic pose encoding is meaningful, we analyze how distances in the latent pose space correlate with the actual spatial effect of pose on the shape. For two pose vectors $\mathbf{p}_i$ and $\mathbf{p}_j$, we measure (i) their distance in the learned pose embedding $\|A(\mathbf{p}_i) - A(\mathbf{p}_j)\|_2$, and (ii) the L^2 distance between the corresponding pose fields $\|f_{\mathcal{P}}(\cdot; \mathbf{p}_i) - f_{\mathcal{P}}(\cdot; \mathbf{p}_j)\|_{L^2(\mathbb{D})}$, which quantifies how much the shape is displaced in the image domain. As shown in Fig. 8, these two quantities exhibit strong linear correlation across randomly sampled pose pairs. This behavior is not incidental: because our pose field construction and Zernike projection are both linear, the pose embedding constitutes an isometric mapping of the pose–field Hilbert space, and distances in the latent space exactly reflect differences in the induced spatial fields. Importantly, this implies that the encoding does not preserve Euclidean distances in raw pose–parameter space, but rather preserves the *functional effect of pose on the shape*, which is the quantity of interest for shape–pose reasoning. The observed correlation therefore confirms that our representation provides a geometrically meaningful and physically grounded embedding of pose.

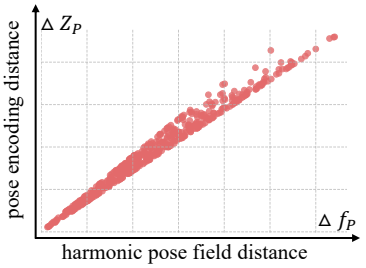


Figure 8: Correlation between harmonic pose field and the final shape pose encodings.

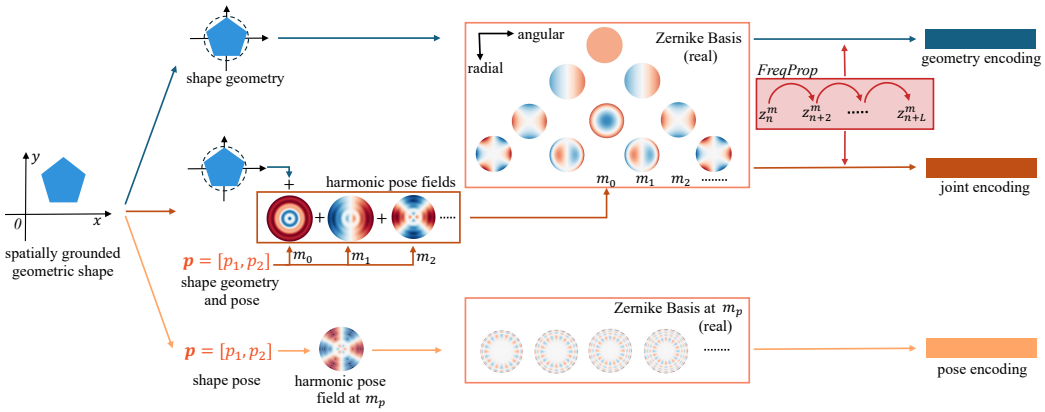


Figure 9: *XShapeEnc* pipeline visualization. The input spatially grounded shape is decomposed into its shape geometry within the unit disk and its shape pose vector. *XShapeEnc* flexibly supports shape geometry encoding, shape pose encoding either independently or jointly, under the same Zernike basis umbrella. The shape pose vector constructs a harmonic pose field lying within the unit disk so that it can be processed by Zernike basis. In the shape geometry encoding and shape geometry and pose joint encoding, a set of Zerkine basis across multiple radial and angular frequency bands are constructed, while in shape pose encoding a set of Zernike basis at particular angular frequency band is constructed. An optionalov frequency propagation (FreqProp) is further constructed to optionally enrich the high-frequencies in the encoding.

We visualize *XShapeEnc* encoding pipeline in Fig. 9, from which we can clearly see that *XShapeEnc* can flexibly encode shape geometry and shape pose either separately or jointly.

A.2 Zernike Basis Linearity and Rotation Equivariance

1. **Linearity**. Linearity means Zernike basis based encoding respects addition and scalar multiplication. The Zernike basis encoding on a composite shape geometry deriving from adding and/or scaling different shape geometries together equals of the same linear operation on the Zernike basis encodings of each individual shape geometry. The mathematical proof is given in Sec. A.17 in Appendix section. Benefiting from the *Linearity* property, we can compositionally derive new shape geometry Zernike basis encoding by linearly compositing Zernike encodings of shape geometries that composite the new shape geometry, without requiring to encode the new shape geometry from scratch.

2. **Rotation Equivariance**, rotating the input shape geometry equals to rotating the final Zernike basis, $Z_n^m(r, \theta + \phi) = Z_n^m(f) \cdot e^{-im\phi}$. Benefiting from the *Rotation Equivariance* property, we can either derive rotation-invariant shape geometry encoding by taking the magnitude $Z_\mathcal{G}$, or rotation-variant feature by keeping the original encoding in Eqn. (6).

A.3 Comparison between Zernike basis and Fourier transform

Compared to conventional 2D Fourier transform, Zernike basis based transform exhibits several notable advantages: 1. its basis functions are polynomials rather than sinusoids, enabling more localized and compact shape representation; 2. the separation of radial and angular components allows independent control over radial resolution and angular frequency, facilitating flexible and fine-grained shape analysis; 3. Zernike basis based encoding possesses favorable properties including *Linearity* which lends us the advantage to compositionally encode complex shape from simple shapes.

A.4 More Discussion on Encoding Recapitulation

Algorithm 1: *XShapeEnc* Algorithmic Workflow

Input : geometric shape $\mathcal{S}$; *FreqProp* coeff. λ; encoding len. L; modulation weight β; optional angular freq. band m_p; Zernike basis $\{V_m^n\}$ (or $\{V_{m_p}^n\}$)

1 decompose $\mathcal{S}$, $\mathcal{S} \to (S_\mathcal{G}, S_\mathcal{P})$, $S_\mathcal{G} \coloneqq f_\mathcal{G}$, $S_\mathcal{P} \coloneqq f_\mathcal{P}$
2 **if** *encode shape geometry* **then**
3 get geometry encoding $Z_\mathcal{G} \leftarrow \boldsymbol{\mathcal{F}}_\mathcal{G}(S_\mathcal{G})$ (Eqn. 6);
4 optionally run *FreqProp* with λ on $Z_\mathcal{G}$ (Eqn. 7);
5 $Z_\mathcal{G} \leftarrow$ complex2real$(Z_\mathcal{G})$ (Sec. 3.7);
6 **return** $Z_\mathcal{G} \in \mathbb{R}^L$;
7 **if** *encode shape pose* **then**
8 get harmonic pose field at m_p, $f_\mathcal{P}(m_p)$ (Eqn. 8);
9 get shape pose encoding $Z_\mathcal{P} \leftarrow \boldsymbol{\mathcal{F}}_\mathcal{P}(S_\mathcal{P})$ (Eqn. 11);
10 **return** $Z_\mathcal{P} \in \mathbb{R}^L$;
11 **if** *joint encode geometry and pose* **then**
12 get harmonic pose fields $\sum_{m_p \in M} f_\mathcal{P}(m_p)$ (Eqn. 8);
13 composite field $f_{\mathcal{GP}} = f_\mathcal{G} + \beta \sum_{m_p \in M} f_\mathcal{P}(m_p)$ (Eqn. 12);
14 joint encoding $Z_{\mathcal{GP}} \leftarrow \boldsymbol{\mathcal{F}}_{\mathcal{GP}}(f_{\mathcal{GP}})$ (Eqn. 14);
15 optionally run *FreqProp* with λ on $Z_{\mathcal{GP}}$ (Eqn. 7);
16 $Z_{\mathcal{GP}} \leftarrow$ complex2real$(Z_{\mathcal{GP}})$ (Sec. 3.7);
17 **return** $Z_{\mathcal{GP}} \in \mathbb{R}^L$;

Output : real-valued encoding: $Z_\mathcal{G}$ or $Z_\mathcal{P}$ or $Z_{\mathcal{GP}}$

Table 5: Relation between real-valued encoding length and required Zernike basis number in shape geometry Encoding and shape geometry and shape pose joint encoding (see Sec. 3.3 and 3.6).

Enc. Length	Basis Num.
256	22
512	31
1024	44
2048	63
4096	91

Table 6: Relation between real-valued encoding length and required Zernike basis number in shape pose encoding (see Sec. 3.5).

Enc. Length	Basis Num.
256	256
512	512
1024	1024
2048	4096
4096	4096

XShapeEnc encoding workflow is shown in Algorithm 1. We can learn that both *Line 3*, *Line 9* and *Line 14* enable vectorized computation. *Line 4*, *Line 9* and *Line 15* just requires a one-time sweep along the initial shape geometry encoding, so the whole workflow is extremely efficient with almost linear time complexity. Moreover, the Zernike basis $\{V_m^n\}$ is pre-constructed. Together with the pre-defined radial frequency propagation coefficient λ and angular frequency m_p, the Zernike basis $\{V_m^n\}$ is used to encode arbitrary shapes.

The shape geometry encoding $Z_\mathcal{G}$ in Eqn. (6) is complex-valued; it serves as the foundational encoding that is fully invertible and frequency-rich. On top of $Z_\mathcal{G}$, we further extract real-valued encoding for practical needs. For example, if the goal is to obtain rotation-invariant encoding, we can simply take magnitude $|Z_\mathcal{G}|$. For rotation-variant encoding, we can adopt any pre-defined rule to flatten the $Z_\mathcal{G}$ encoding into real-valued vector encoding as long as the original $Z_\mathcal{G}$ can be recovered by reversing the rule. In order to preserve as much information as possible in Eqn. (6) in the final real-valued encoding, we exploit the conjugate symmetry $z_n^{-m} = \overline{z_n^m}$ and retain only non-negative angular orders. Moreover, we flatten each complex-valued coefficient by putting its real and imaginary parts in juxtaposition. The relation between the final real-valued encoding length and the required Zernike basis number is given in Table 5, from which we can learn that we can obtain the commonly

used encoding length with limited Zernike bases. For the shape pose encoding $Z_{\mathcal{P}}$ in Eqn. (11), no complex-to-real value conversion is needed as it is already real-valued. To obtain the same encoding length as shape geometry encoding, it essentially requires twice the number of Zernike bases required by shape geometry encoding $Z_{\mathcal{G}}$.

A.5 More Discussion on FreqProp

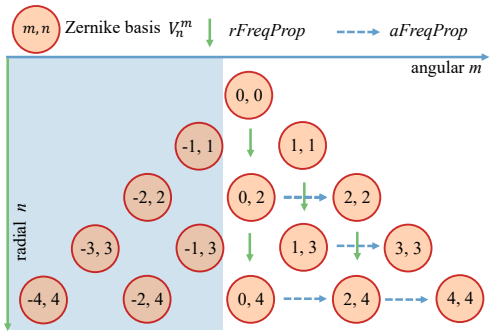


Figure 10: *FreqProp* visualization. *rFreqProp* and *aFreqProp* propagate along fixed-angular and and fixed-radial Zernike basis, respectively. *FreqProp* is invertible as we reverse the propagation process. We do not need to run *FreqProp* on negative angular Zernike basis ($m < 0$, overlaid by light blue) because their projection coefficients show conjugate symmetry ($z_n^{-m} = \overline{z_n^m}$) with their positive angular Zernike basis counterpart. We ignore the negative angular part when deriving the final encoding (see Sec. 3.7).

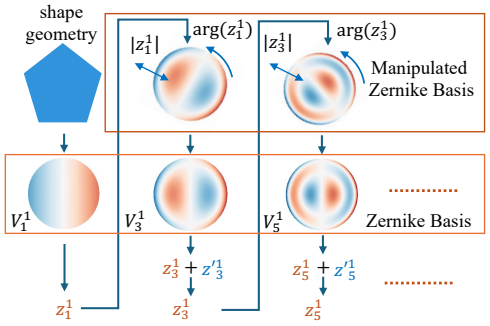


Figure 11: *rFreqProp* geometric interpretation illustration. We use the angular frequency band $m = 1$ as the propagation direction. The input shape geometry is projected onto the Zernike basis V_1^1 to get the projection coefficient z_1^1, which is then used to manipulate the next adjacent Zernike basis V_3^1 by scaling it by $|z_1^1|$ and rotating it by $\arg(z_1^1)$ before projecting onto V_3^1. The same rule applies to all other follow-up Zernike bases, including Z_5^1, Z_7^1, *etc*.

The frequency propagation is visualized in Fig. 10 and its geometric interpretation is shown in Fig. 11.

A.6 More Discussion on Shape Geometry Encoding Invertibility

Given the constructed shape corpus *XShapeCorpus*, we exhaustively test shape geometry reconstruction error (mean square error MSE) under various encoding length and shape geometry complexity. To this end, first, we encode each shape geometry by setting the rasterization resolution as 300, and encode the target length covers commonly used length: 64, 128, 256, 512, 1024, 2048, 4096. After applying the inversion process, we compute the shape geometry reconstruction error under various target length and shape complexity informed by depth. Five qualitative shape geometry reconstruction under the 7 encoding lengths on 2 complex shapes (depth=$1, 4, 6, 6, 8, 10$) are shown in Fig. 17, from which we can learn that the input complex geometric shape can be reconstructed from the shape geometry encoding. The larger the encoding length, the higher the reconstruction accuracy. The quantitative reconstruction MSE variation w.r.t. different shape complexity (indicated by depth) under various encoding length is shown in Fig. 18 in the main paper. We observe from this figure that longer encoding length results in smaller MSE value, while more complex shapes lead to higher MSE than simpler ones. This observation is consistent with *XShapeEnc* encoding principle where the input shape geometry is essentially approximated by a set of orthogonal Zernike basis. More complex shape geometry requires more bases, and encoding length cutoff inevitably leads to higher reconstruction error.

A.7 More Discussion on Geometry and Pose Join Encoding

We present shape geometry encoding in Sec. 3.3 and shape pose encoding in Sec. 3.5 separately. Although both relying on Zernike basis, they require to construct different Zernike basis for encoding. While shape pose encoding requiring to construct the Zernike basis at predefined fixed angular frequency band, shape geometry encoding instead require to construct the Zernike basis across multiple angular frequencies. To jointly encode shape geometry and pose into one representation $Z_{\mathcal{GP}}$ and with exactly the same Zernike basis, we propose a novel shape geometry and shape pose joint encoding framework.

Two most straightforward approaches are to jointly encode in the final feature space via either *Addition* or *Concatenation*. For *Addition*, the geometry encoding $Z_{\mathcal{G}}$ is combined with the pose encoding $Z_{\mathcal{P}}$ through elementwise addition, weighted by a scalar α. For *Concatenation*, the two encodings are simply concatenated:

$$\text{Add: } Z_{\mathcal{GP}} = Z_{\mathcal{G}} + \alpha \cdot Z_{\mathcal{P}}; \quad \text{Concate: } Z_{\mathcal{GP}} = [Z_{\mathcal{G}}; \alpha \cdot Z_{\mathcal{P}}]. \tag{15}$$

However, both *Addition* and *Concatenation* in Eqn. (15) suffer from fundamental limitations. First, neither yields a compact representation. The Zernike bases used for geometry encoding (Eqn. 6) and pose encoding (Eqn. 11) are inherently mismatched: geometry encoding spans multiple angular frequency bands, whereas pose encoding is restricted to a single designated angular frequency. This basis mismatch leads to inefficient use of the encoding space. Second, elementwise *Addition* inevitably entangles geometry and pose, causing mutual interference between the two components. In contrast, *Concatenation* preserves disentanglement but imposes a strict encoding-length constraint: to maintain a fixed-length representation, the available dimensional budget must be split between $Z_{\mathcal{G}}$ and $Z_{\mathcal{P}}$, resulting in reduced expressiveness for both. These limitations motivate us to design a more principled and compact joint encoding strategy.

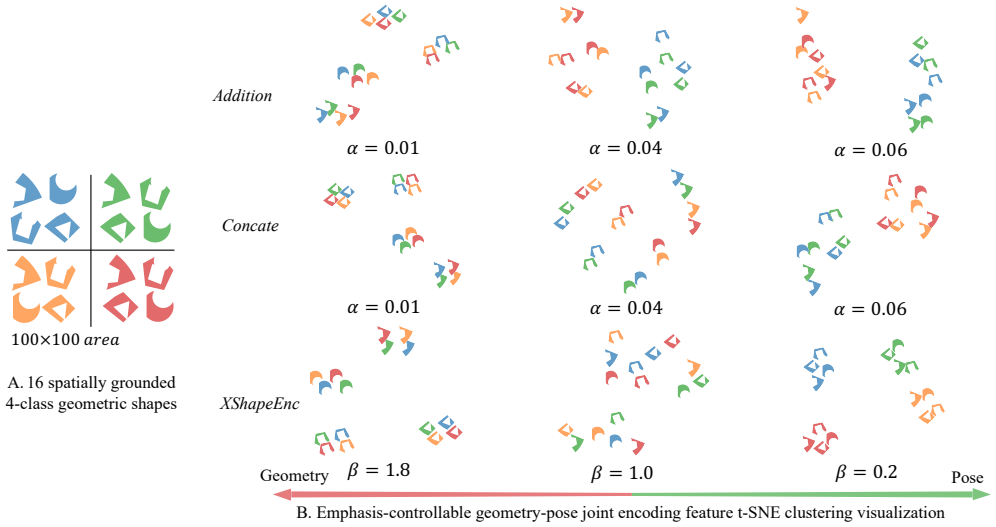


Figure 12: Shape Geometry and Pose Joint Encoding t-SNE [47] clustering result visualization

To assess the joint encoding discriminability, we choose 4 exemplar shape geometries from *XShapeCorpus* with depth $d = 2$, and further place each of the 4 shape geometries at the 4 sub-areas independently (100 $m \times 100$ m area, divided into 4 sub-areas: topleft, topright, bottomleft and bottomright). As a result, each shape geometry has been duplicated at the 4 sub-areas but with different shape pose (*e.g.*, x- and y- coordinates), leading to a total of 16 spatially grounded shape geometries. We then encode each of the spatially grounded shape geometry by the shape geometry and pose jointly encoding strategy as presented in Sec. 3.6. By varying the modulation weight β in Eqn. 14, we can obtain the jointly encoding with different emphasis between shape geometry and shape pose. We test the discriminability by running t-SNE [47] clustering on the joint encoding under different modulation weight β to test if the joint encoding reflects the emphasis governed by β. Meanwhile, we compare the other two straightforward joint encoding methods presented in Eqn. 15: *Addition* and *Concatenation*.

The clustering resulting is shown in Fig. 12, from which we can clearly see that: smaller β results in the joint encoding emphasizing on shape geometry, while larger β instead emphasizing on shape pose. It thus shows the discriminability in our proposed shape geometry and shape pose joint encoding. However, *Addition* and *Concate* can just emphasize geometry. The increased α confuses the shape geometry and shape pose, resulting meaningless clustering (see the two clustering with $\alpha = 0.06$).

A.8 XShapeCorpus Dataset Detailed Introduction

Currently there is no public dataset in which each 2D shape is an arbitrary geometric shape (aka shape geometry) and further paired with a spatial position (aka shape pose). Existing datasets such as ElementaryCQT [2], AutoGeo [21], CE-Shape-1 [39] and mendeley 2D geometric shapes dataset introduced in [26] contain just simple geometric shapes (*e.g.*, triangle, square, circle) and have been associated with no spatial pose information. We hereby curate **XShapeCorpus** - a novel 2D geometric shape corpus highlighting shape diversity and pose diversity. Specifically, we depend on 8 shape primitives that are commonly seen in our daily lives:

circle, *square*, *rectangle*, *triangle*, *diamond*, *ellipse*, *pentagon*, *sector*

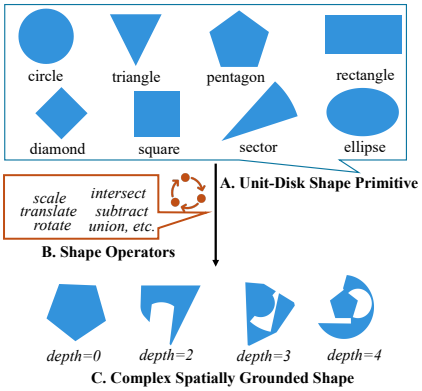


Figure 13: *XShapeCorpus* curation: Shape primitive and shape curation process visualization. More complex shape can be created by consecutively running shape operations. Higher depth (operation number) indicates higher Shape complexity. Each each is independently associated with spatial pose.

Each of the 8 shape primitives is normalized to lie within the unit disk. More complex 2D geometric shapes can be constructed by iteratively running either unary or binary shape operators. The unary shape operator operates on a single shape and contains 3 operations: *scale*, *translate*, *rotate*. The binary shape operator operates on two shapes to create a new shape, and we incorporate 5 binary shape operators: *subtract*, *union*, *intersect*, *symmetric difference* (*xor*), *convex hull*.

The shape curation pipeline is shown in Fig. 13, an intermediate shape created by the preceding operation is fed to the next randomly chosen operation to generate the new shape. As more operations generally result in higher shape complexity, we use the "depth", which indicates the number of operations used to construct a shape, to measure a shape's complexity. To address the potential collapse back to plain shape primitive or the existence of primitive-shaped "tiny hole" of the created shape, we avoid directly using raw shape primitive during intermediate shape creation steps but instead wrap it to match the intermediate shape fraction. Moreover, we further enforce binary operation number to be executed to be proportional to "depth" to guarantee higher depth value is mostly associated with more complex shape. In practice, we curate *XShapeCorpus* for depth ranging from 1 to 10. For each depth, we construct 100 shapes, resulting in a total of 1,000 shapes.

A.9 Detailed Comparing Baseline Discussion

For boundary point based encoding, we incorporate 4 baselines:

1. **AngularSweep**. The angular sweep encoding strategy has been initially proposed and adopted by SoundTRC [19] and ReZero [15] to encode a 2D convex geometric shape. The underlying idea is to shoot a line from the origin and sweep counterclockwise (or clockwise) at a predefined angular interval. The final shape geometry encoding is obtained by sequentially appending the distance value between the origin and shape boundary at each sweep angle. The original angular sweep in SoundTRC [19] and ReZero [15] can only encode convex shape geometry (they assume the shooting line just intersects with the shape geometry once). We extend it to accommodate arbitrary shape geometry encoding by appending all distance values in close-to-far order at one angular interval.
2. **2DPE**. Given shape geometry represented by a binary mask, we discretize the mask into finite grid points and adopt sinusoidal positional encoding [49] to encode each point position. The final shape geometry encoding can be obtained by mean-pooling all point encodings. **2DPE** can be used to encode both shape geometry and shape pose.
3. **ShapeDist** [33], Shape distributions provide a classical training-free approach for characterizing object geometry by analyzing statistical relationships between sampled points on the object's surface. In particular, the D2 distribution computes distances between randomly selected point pairs to form a global signature of shape. To adapt this concept to 2D silhouettes in our setting, we detect the boundary of the binary shape mask and

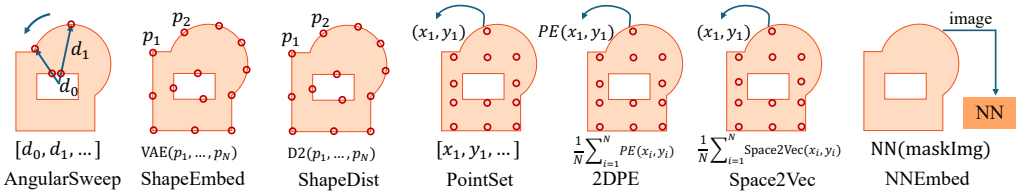


Figure 14: Typical baselines illustration: *AngularSweep*, ShapeEmbed [40], ShapeDist [33] are based on shape geometry boundary points, the other three baselines (PointSet, 2DPE and Space2Vec [29]) are based on regularly sampled points.

deterministically sample boundary points at a uniform stride. We then compute all pairwise Euclidean distances among the sampled points, normalize them to ensure scale invariance, and convert them into a fixed-length histogram. The resulting descriptor acts as a compact and permutation-invariant shape representation that captures global boundary geometry without requiring learning or specialized basis functions. This serves as a strong classical baseline complementary to our proposed XShapeEnc framework.

4. **ShapeContexts** [1]. Serving as a classic 2D shape description, Shape Context [1] characterizes a shape by sampling contour points and recording the relative spatial distribution of other points in a log-polar histogram around each reference point. It captures local and global geometric structure, offering robustness to moderate deformation and making it a strong baseline for shape description.

For boundary line based encoding, we incorporate one baseline,

1. **Poly2Vec** [43], a recent method for encoding spatially grounded polygon shapes. To extend it to arbitrary shapes, we intentionally approximate non-linear shape boundaries using sets of line segments.

For regular-sampled point based encoding, we incorporate 3 baselines,

1. **PointSet**. We approximate the geometric shape by point set by grid-sampling a set of 2D points on the shape mask. Flattening each point's $[x, y]$ coordinates and further concatenating all points' coordinates together in pre-defined order (*e.g.*, scanning each row from top to the bottom) gives us the geometric shape encoding. Point set is a well-established and flexible approach in graphics, shape analysis. In our experiment, we adopt it to represent the shape geometry.
2. **ShapeEmbed**. ShapeEmbed [40] first converts the shape mask boundary into a translation-, rotation-, and scale-invariant Euclidean distance matrix, then trains a variational autoencoder (VAE) to encode the matrix into latent space representation. We adopt the ShapeEmbed [40] to encode the shape geometry.
3. **Space2Vec**. Space2Vec [29] encodes spatial locations into fixed-dimensional vectors using a set of multi-scale sinusoidal basis functions inspired by biological grid cell responses. We incorporate it as a baseline to encode both shape geometry and shape pose. Given a point (x, y) sample, Space2Vec projects it onto multiple spatial frequencies and directions to generate the encoding. To adapt Space2Vec to our setting, we approximate the shape mask by point set and compute per-point Space2Vec encoding. The final shape geometry encoding is obtained by mean average pooling all point encodings. Unlike the original Space2Vec, which is designed for general geospatial representation with optional neural projection layers, our adaptation removes all learnable components and uses a simple point sampling strategy with mean pooling, yielding a lightweight geometry representation suitable as a competitive training-free baseline for *XShapeEnc*.

For neural network based encoding (*NNEmbed*), we test 3 image-based neural networks,

1. **ResNet18** [17], which is a widely used convolutional neural network model pretrained on ImageNet dataset [10].

2. **ViT** [11], which is Transformer [49] based neural network model pretrained on ImageNet dataset [10].
3. **CLIP** [38], which is a neural network model pre-trained on massive aligned text-image paired data. It lays emphasis on the input image's semantics when encoding a shape.

We illustrate the encoding procedures underlying a subset of the baselines in Fig. 14. For ShapeEmbed [40], we train separate ShapeEmbed for different shape geometry mask dataset in self-supervised manner. For example, we train different ShapeEmbed model for *XShapeCorpus* dataset, Mendeley 2D shape data [26] and target region control dataset, separately. We refer to the original ShapeEmbed [40] for the implementation. In training, we adopt Adam optimizer [24] with learning rate 0.001, and we train 100 epochs on NVIDIA A100 GPU.

A.10 More Ablation Study

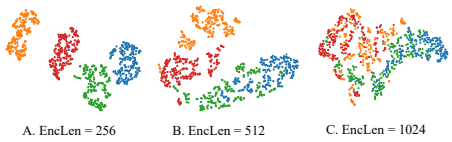


Figure 15: t-SNE [47] clustering result visualization on various encoding length

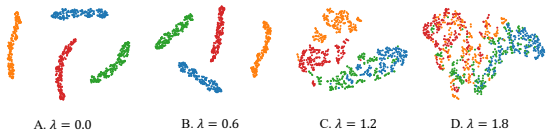


Figure 16: t-SNE [47] clustering result visualization on frequency propagation coefficient λ.

Ablation on Encodng Length. In the experiment in the main section, we used the encoding length 512 for the shape geometry discriminability test. To further test the impact of encoding length, we further test the clustering result with encoding length in 256, 512 and 1024. From the result shown in Fig. 15, we can observe that, under shape augmentation, longer shape encoding length introduces more encoding feature difference. This is anticipated because longer encoding length captures more localized shape feature. Shape augmentation pronounces such localized shape feature variation.

Ablation on FreqProp λ. We further ablate the frequency propagation coefficient λ on the shape geometry discriminability. By varying $\lambda \in [0.0, 0.6, 1.2, 1.8]$, we visualize the corresponding clustering result in Fig. 16. We can clearly see higher λ introduces larger portion of low-frequency coefficient to high-frequency coefficient, the resulting encoding gradually loses shape discriminability.

A.11 More Discussion on Shape Geometry Invertibility

Shape Geometry Encoding Invertibility. We encode all the shape geometries (see Eqn. (6)) in *XShapeCorpus* with encoding length of 64, 128, 256, 512, 1024, 2048, 4096, then further apply the inversion procedure to reconstruct the shape geometry. Two sets of reconstructed shape geometries w.r.t different encoding length in Fig. 6, from which we can clearly see that larger encoding length results in higher reconstruction accuracy. The quantitative reconstruction mean square error (MSE) is shown in Fig.

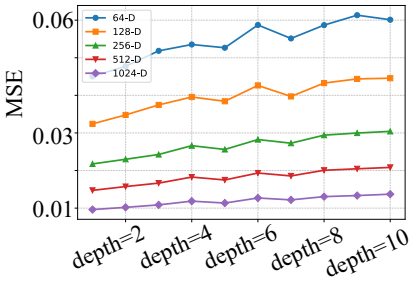


Figure 18: MSE variation for encoding length over depth shape complexity on *XShapeCorpus* dataset.

Given the constructed shape corpus *XShapeCorpus*, we exhaustively test shape geometry reconstruction error (mean square error MSE) under various encoding lengths and shape geometry complexity. To this end, first, we encode each shape geometry by setting the rasterization resolution as 300, and use target encoding lengths covering 64, 128, 256, 512, 1024, 2048, 4096. After applying the inversion process, we compute the shape geometry reconstruction error under various target lengths and shape complexity informed by depth. Five qualitative shape geometry reconstructions under the 7 encoding lengths on 5 complex shapes (depth=$1, 4, 6, 8, 10$) are shown in Fig. 6, from which we can learn that the input complex geometric shape can be reconstructed from the shape geometry encoding. The larger the encoding length, the higher the reconstruction accuracy. The quantitative reconstruction MSE variation w.r.t. different shape complexity (indicated by depth) under various encoding length is shown in Fig. 18.

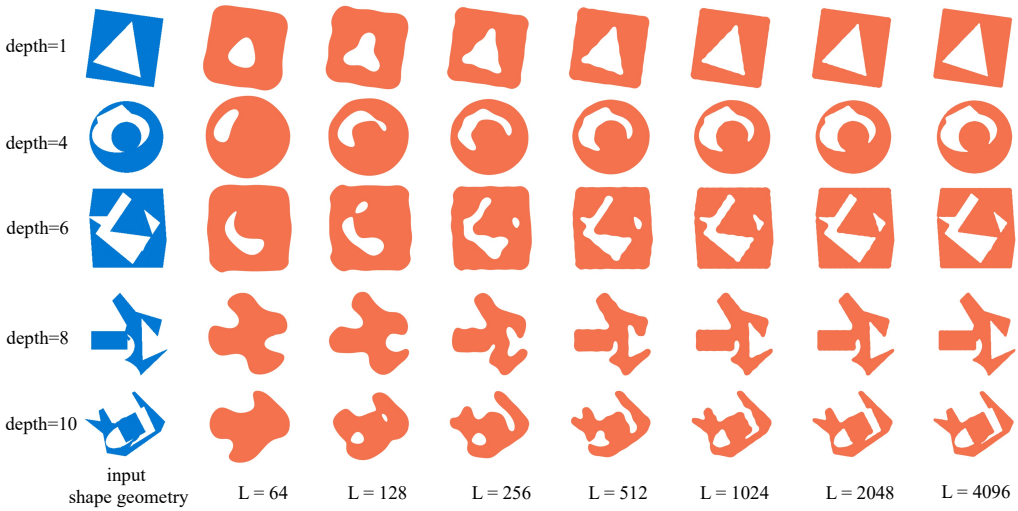


Figure 17: *XShapeEnc* encoding invertibility illustration. We show two complex shape geometry (within unit disk) with depth = 6 and depth = 8 and their reconstructed shape geometry under various encoding length. Note that the original reconstructed shape is soft-masked, we binarize it with the threshold 0.2 for better visualization.

We observe from this figure that longer encoding length results in smaller MSE values, while more complex shapes lead to higher MSE than simpler ones. This observation is consistent with *XShapeEnc* encoding principle where the input shape geometry is essentially approximated by a set of orthogonal Zernike basis. More complex shape geometry requires more bases, and encoding length cutoff inevitably leads to higher reconstruction error.

We provide more shape geometry invertibility visualization in Fig. 17.

A.12 XShapeEnc Shape Pose Encoding Discriminability

To assess *XShapeEnc* shape pose encoding discriminability, we construct a 100×100 m^2 area and evenly divide it into 4 sub-areas: *topleft*, *topright*, *bottomleft* and *bottomright*. Within each sub-area, we randomly sample 200 shape poses and use *XShapeEnc* and 2D positional encoding [49] to encode each shape pose into 512 d features. The t-SNE [47] clustering results for *XShapeEnc* and *2D PE* encoding are shown in Fig. 19. From this figure, we can learn that both *XShapeEnc* and *2D PE* show shape pose discriminability. Compared to *2D PE*, which mainly preserves absolute coordinate values, *XShapeEnc* organizes poses along continuous low-dimensional manifolds (belt-like trajectories in t-SNE). This reveals that *XShapeEnc* embeds spatial transformations in a geometrically consistent manner—small pose perturbations correspond to small latent displacements, while different pose branches remain clearly separated. In summary, *XShapeEnc* exhibits strong encoding discriminability for both shape geometry and pose encoding.

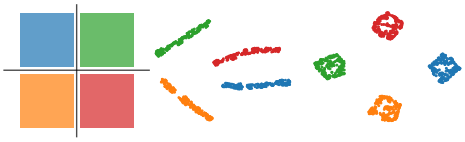


Figure 19: Shape pose t-SNE [47] clustering visualization. We divide a region into four main sub-regions and sample 200 points at each sub-region (subfig. A). Both *XShapeEnc* and classic 2D positional encoding can successfully cluster shape poses based on their spatial position.

A.13 More Discussion on Shape Retrieval

We first evaluate *XShapeEnc* shape geometry encoding capability on the classic 2D shape task: shape retrieval. In shape retrieval, each shape (in our case, shape geometry) is represented within an image. Given a query shape image, the target is retrieve images of the same class from the gallery set by computing pairwise encoded feature similarity (in our case, we use $\cos()$ similarity). For evaluation metric, we adopt mean average precision (mAP) which aggreates the average precision (AP) of

Table 7: 2D shape retrieval result on Mendeley 2D shape dataset [26] and MPEG-7 CE-Shape-1 Part B dataset [27]. We report mean Average Precision (mAP) by aggregating average prevision (AP) of querying each shape in the datasets. The encoding length is 512. The top-1, top-2 and top-3 performing methods are labeled with different color.

Description	Method Name	Mendeley Shape	MPEG-7 Shape
Training-free	PointSet	0.21	0.10
	ShapeContexts [1]	0.55	0.45
	Space2Vec [29]	0.30	0.47
	ShapeDist [33]	0.25	0.11
	AngularSweep [19]	0.42	0.26
Training-required	ShapeEmbed [40]	0.73	0.41
Pretrained Model	ResNet18 [17]	0.40	0.56
	ViT [11]	0.43	0.57
	CLIP [38]	0.57	0.64
Ours	*XShapeEnc* (FreqProp+Comp2Real)	0.53	0.54
	XShapeEnc (FreqProp+Mag.)	0.88	0.58
	XShapeEnc (Mag.)	**0.91**	**0.59**

querying each shape image. It reflects both retrieval accuracy and ranking quality, and serves as a general and widely adopted metric for shape retrieval performance.

We run shape retrieval on two public datasets: Mendeley 2D shape benchmark [26] and MPEG-7 CE-Shape-1 Part B dataset [27]. Mendeley 2D shape benchmark consists of 9 primitive shape classes: Triangle, Square, Pentagon, Hexagon, Heptagon, Octagon, Nangon, Circle and Star. Each class is instantiated with 10,000 shapes with large variations in scale, rotation and deformation. MPEG-7 CE-Shape-1 Part B dataset [27] contains 70 organic shape classes, ranging from animal shape (*e.g.*, cattle, beettle, camel, butterfly), people (*e.g.*, children, face), utensils (*e.g.*, fork, spoon, jar) to devices (*e.g.*, watch, cellular phone). Each class associates with 20 images, resulting in a total of 1,400 images. The two datasets test the shape geometry encoding generalization capability in handling both shape geoemtry primitives and real-world shape geometries. We compare *XShapeEnc* with 9 baselines, comprehensively covering training-free, training-required and pre-trained models. Within *XShapeEnc*, we evaluate three variants: 1. *XShapeEnc* with FreqProp (see Sec. 3.4) and complex-to-real conversion (FreqProp + Comp2Real), the resulting feature is rotation-variant and thus not ideal for the shape retrieval task by design; 2. *XShapeEnc* with FreqProp followed by taking the magnitude of the complex encoding (Eqn. 7) as the final feature, which is approximately rotation invariant because FreqProp process relatively breaks the rotation invariance (FreqProp + Mag.); 3. *XShapeEnc* without FreqProp and taking the magnitude from the complex encoding (Eqn. 6) as the final feature (Mag.).

The quantitative retrieval results are reported in Table 7. Several clear findings emerge. First, *XShapeEnc* (Mag.) achieves the best mAP (0.91) on the Mendeley shape dataset and the second-best mAP (0.59) on the MPEG-7 CE-Shape-1 Part B dataset, substantially outperforming most baselines. This indicates that *XShapeEnc* captures global shape structure more consistently under large intra-class variation. Second, *XShapeEnc* (FreqProp+Mag.) also performs strongly (mAP 0.88), suggesting that frequency enrichment remains effective even when propagation introduces slight rotation sensitivity. Third, while all non-pretrained baselines show performance drops on MPEG-7 CE-Shape-1 Part B [27], all three pretrained models show performance gains. We attribute this trend to the stronger correlation between MPEG-7 categories and natural-image semantics (e.g., ImageNet [10]) used during pretraining. Nevertheless, *XShapeEnc* remains highly competitive across datasets with very different shape distributions.

A.14 More Discussion on Inter-Shape Topological Relation Classification

To evaluate shape geometry and shape pose joint encoding capability, we follow Poly2Vec [43] to run experiments on the inter-shape topological relation classification task [8]. The data is from geospatial OpenStreetMap [28] in two cities: New York and Singapore, in which each shape is a 2D polygon associated with a spatial position. The 2D polygons indicate the building contour structure from the bird's-eye view (BEV), so they vary significantly w.r.t. polygon size, shape structure and orientation. Two such polygon shapes jointly present 5 potential topological relations: *disjoint*, *touch*, *equal*,

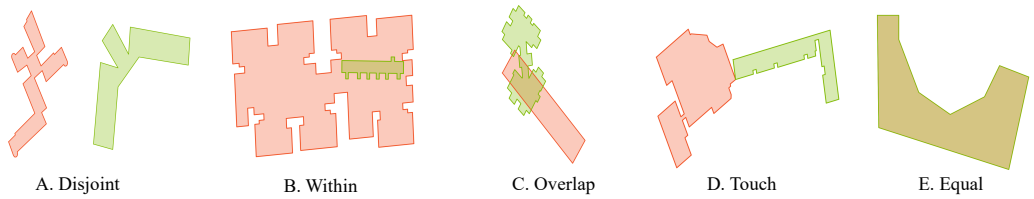


Figure 20: Polygon-polygon topological relation visualization: we visualize 5 inter-shape topological relations in OpenStreetMap Singapore dataset: *Disjoint*, *Within*, *Overlap*, *Touch* and *Equal*. *Equal* means two polygons are identical. From these visualizations, we can learn that these polygons vary drastically in terms of shape geometry complexity, size and scale.

overlap and *within* (see Fig. 20). The shape geometry and shape pose of the two shapes intertwine to inform the topological relation, so the topological relation classification task serves as an ideal testbed to test the geometry-pose joint encoding capability.

Following the configuration of Poly2Vec [43], for each topological relation we construct 3,000 polygon-polygon pairs for training and 1,000 pairs for testing, resulting in 15,000/3,000 training/test pairs for both New York and Singapore. For all baselines except Poly2Vec [43], we represent the two shape masks involved in a topological relation within a single image. When the two masks are too far apart (*e.g.*, in the *disjoint* case), we intentionally move them closer so that both can be enclosed within one image while preserving their topological relation. In *XShapeEnc*, each spatially grounded polygon is encoded into a 512-d feature with $\beta = 0.2$. For fair comparison, we follow Poly2Vec [43] and use a two-layer multi-layer perceptron (MLP) to predict the logits. The whole network is trained with cross-entropy loss using the Adam optimizer with a learning rate of 0.001. The GPU is A100 and we train 100 epochs.

The classification accuracies are reported in Table 4 in the main paper. Several observations support the effectiveness of *XShapeEnc*. First, on the Singapore split, *XShapeEnc* achieves the best accuracy (0.760), outperforming the strongest competing baseline CLIP [38] (0.700) by 0.060, as well as Poly2Vec [43] (0.702) and Space2Vec [29] (0.706). This margin is meaningful because the task depends not only on local contour cues but also on jointly reasoning about global geometry and relative pose. Second, on the New York split, *XShapeEnc* reaches 0.768, remaining highly competitive with the best-performing CLIP model (0.779) and outperforming Poly2Vec (0.684) by a large margin. These results suggest that *XShapeEnc* provides a strong geometry-pose representation without relying on large-scale pretraining, and that its advantage is especially pronounced when topological reasoning must be inferred directly from structured spatial shape relations rather than high-level semantic priors.

A.15 More Experiments

Acoustic Spatial Target Region Control Spatial acoustic target region control task has been recently proposed in [15, 19, 22], in which it aims to extract the spatial audio within a pre-specified 2D target region from the audio mixture. This task serves as an ideal scenario to test the 2D spatially grounded geometric shape encoding capability because its output is solely conditioned on the specified target region that accounts for both shape geometry and shape pose. We follow the setting used by SoundTRC [19] to focus on three kinds of spatially grounded geometric shapes: *Angle*, *Distance* and their combination *Angle-Distance*. By varying the angle and distance for each geometric shape, we can construct a set of indoor environment 2D geometric shapes on the floorplan plane, each geometric shape is spatially grounded to the 3D indoor environment center. Each spatially grounded 2D geometry shape's shape geometry and shape pose are jointly encoded with neural geometry-pose emphasis (where $\beta = 1$). While following the setting in [19], we constrain all sound sources to lie on the same plane and adopt Pyroomacoustics [41] to synthesize the data.

Table 8: Quantitative result on TRC task across three shapes: *Angle*, *Distance* and *Angle-Dist*.

Encoding Method	SDR (↑)	STOI (↑)	PESQ (↑)
AngularSweep [19]	14.10	0.82	2.31
Space2Vec [29]	15.00	0.83	3.40
Poly2Vec [43]	15.10	0.89	3.90
ShapeEmbed [40]	15.37	0.97	4.26
XShapeEnc (w/o *FreqProp*)	15.11	0.91	3.99
XShapeEnc	**17.62**	**0.98**	**4.38**

Specifically, we have synthesized 30-hour and 6-hour audio training and test, respectively. The sampling frequency is 16 kHz and each audio sample is 3 seconds long. The IRM mask (ideal ratio mask) to be learned is of shape 160×320. For both ShapeEmbed [19] and XShapeEnc, we directly use SoundTRC [19] neural network. For Poly2Vec [43], we follow its original setting to add an MLP layer to fuse the magnitude and phase features. We train all models on 8 NVIDIA A100 GPUs and train them for 150 epochs (afterwards, we find the performance gradually decays). Adam optimizer [24] is used with an initial learning rate 0.001, which decays every 50 epochs with a decay rate of 0.5.

The quantitative results are reported in Table 8 in the main paper, where *XShapeEnc* consistently achieves the best performance across all three evaluation metrics. In particular, *XShapeEnc* reaches an SDR of 17.62, outperforming the strongest baseline ShapeEmbed [40] (15.37), as well as Poly2Vec [43] (15.10), Space2Vec [29] (15.00), and SoundTRC [19] (14.10). Similar advantages are also observed for STOI and PESQ, where *XShapeEnc* achieves the top scores of 0.98 and 4.38, respectively. These gains are particularly meaningful because the target region control task depends on precise modeling of both shape geometry and spatial pose: even small encoding errors can degrade the alignment between the specified region and the extracted audio. The strong and consistent improvements therefore provide direct evidence that *XShapeEnc* offers a more faithful and task-relevant representation of spatially grounded geometric shapes, making it highly practical for downstream tasks built upon 2D geometric shape modeling.

Medical Science We depend on ISIC 2016 Task 3B [16], which is medical segmented lesion classification task. It provides 900 segmented lesion mask images, the each mask indicates one of the two classes: benign and malignant. For all the masks, we first extract XShapeEnc and other baselines feature (length is 512), then we train a linear probing layer to classify the feature. The 900 mask images are splited into 720 and 180 for training and testing, respectively, the accuracy rate is shown in Table 9. We can clearly learn that *XShapeEnc* outperforms all the 5 baselines, 2% over EFT [36] and 4% over ShapeDist [33].

Table 9: Accuracy rate comparison.

Method	**Accuracy Rate (↑)**
ShapeContexts [1]	68.33%
EFT [36]	73.89%
Shape2Vec [29]	71.11%
AngularSweep [19]	62.78%
ShapeDist [33]	72.22%
XShapeEnc	**76.11%**

A.16 Zernike Basis Orthogonality Proof

Proof. Let the Zernike basis be defined as,

$$V_n^m(r,\theta) = R_n^{|m|}(r)\, e^{\mathrm{i}m\theta}, \tag{16}$$

where $R_n^{|m|}(r)$ is the radial polynomial and (r,θ) are polar coordinates with $r \in [0,1]$, $\theta \in [0, 2\pi)$. We aim to prove,

$$\int_0^1 \int_0^{2\pi} V_n^m(r,\theta) \left(V_{n'}^{m'}(r,\theta)\right)^* r\, d\theta\, dr = \frac{\pi}{n+1}\, \delta_{nn'}\delta_{mm'} \tag{17}$$

The integrand in the above equation can be separated as,

$$\int_0^1 \int_0^{2\pi} V_n^m(r,\theta) \left(V_{n'}^{m'}(r,\theta)\right)^* r\, d\theta\, dr = \left(\int_0^{2\pi} e^{\mathrm{i}(m-m')\theta}\, d\theta\right) \left(\int_0^1 R_n^{|m|}(r)\, R_{n'}^{|m'|}(r)\, r\, dr\right). \tag{18}$$

The angular integral yields,

$$\int_0^{2\pi} e^{\mathrm{i}(m-m')\theta}\, d\theta = 2\pi\, \delta_{mm'} \tag{19}$$

Hence it remains to show that, for fixed $m \geq 0$,

$$\int_0^1 R_n^m(r)\, R_{n'}^m(r)\, r\, dr = \frac{\delta_{nn'}}{2(n+1)} \tag{20}$$

Let $k = \frac{n-m}{2} \in \mathbb{N}_0$. A standard identity for Zernike radial polynomials is

$$R_{m+2k}^m(r) = r^m\, P_k^{(0,m)}\big(2r^2 - 1\big)\,, \tag{21}$$

where $P_k^{(\alpha,\beta)}$ is the Jacobi polynomial. Set $x = 2r^2 - 1$, so that $r\, dr = \frac{1}{4}\, dx$ and $r^2 = \frac{1+x}{2}$. The radial integral becomes

$$\begin{aligned}\int_0^1 R_{m+2k}^m(r)\, R_{m+2k'}^m(r)\, r\, dr &= \frac{1}{4}\int_{-1}^1 \left(\frac{1+x}{2}\right)^m P_k^{(0,m)}(x) P_{k'}^{(0,m)}(x)\, dx \\ &= 2^{-(m+2)} \int_{-1}^1 (1+x)^m P_k^{(0,m)}(x) P_{k'}^{(0,m)}(x)\, dx.\end{aligned} \tag{22}$$

The Jacobi polynomial orthogonality relation (for $\alpha = 0$, $\beta = m$) is

$$\int_{-1}^1 (1-x)^0 (1+x)^m P_k^{(0,m)}(x) P_{k'}^{(0,m)}(x)\, dx = \frac{2^{m+1}}{2k+m+1}\, \delta_{kk'} \tag{23}$$

Therefore,

$$\int_0^1 R_{m+2k}^m(r)\, R_{m+2k'}^m(r)\, r\, dr = 2^{-(m+2)} \cdot \frac{2^{m+1}}{2k+m+1}\, \delta_{kk'} = \frac{1}{2(2k+m+1)}\, \delta_{kk'} \tag{24}$$

Since $n = m + 2k$, we have $2k + m + 1 = n + 1$, giving

$$\int_0^1 R_n^m(r)\, R_{n'}^m(r)\, r\, dr = \frac{\delta_{nn'}}{2(n+1)} \tag{25}$$

Combining with the angular part,

$$\int_0^1 \int_0^{2\pi} V_n^m(r,\theta)\, \big(V_{n'}^{m'}(r,\theta)\big)^*\, r\, d\theta\, dr = (2\pi\, \delta_{mm'}) \cdot \left(\frac{\delta_{nn'}}{2(n+1)}\right) = \frac{\pi}{n+1}\, \delta_{nn'} \delta_{mm'}. \tag{26}$$

This completes the proof. In particular, if we define the normalized basis

$$\widetilde{V}_n^m(r,\theta) = \sqrt{\frac{n+1}{\pi}}\, R_n^{|m|}(r)\, e^{\mathrm{i}m\theta}, \tag{27}$$

then $\{\widetilde{V}_n^m\}$ forms an orthonormal set on the unit disk with respect to the measure $r\, dr\, d\theta$. □

A.17 Zernike Basis Shape Linearity Proof

Let $f_1(r,\theta)$ and $f_2(r,\theta)$ be two functions defined over the unit disk $r \in [0,1], \theta \in [0, 2\pi)$, and let $f(r,\theta) = a f_1(r,\theta) + b f_2(r,\theta)$, where $a, b \in \mathbb{R}$.

The Zernike moment of order (n,m) of a function f is defined as:

$$Z_n^m(f) = \int_0^1 \int_0^{2\pi} f(r,\theta)\, V_n^m(r,\theta)^*\, r\, d\theta\, dr \tag{28}$$

where $V_n^m(r,\theta)$ is the complex-valued Zernike basis function and * denotes complex conjugation.

Start with:

$$Z_n^m(f) = \int_0^1 \int_0^{2\pi} (a f_1(r,\theta) + b f_2(r,\theta))\, V_n^m(r,\theta)^*\, r\, d\theta\, dr \tag{29}$$

$$= a \int_0^1 \int_0^{2\pi} f_1(r,\theta) V_n^m(r,\theta)^*\, r\, d\theta\, dr + b \int_0^1 \int_0^{2\pi} f_2(r,\theta) V_n^m(r,\theta)^*\, r\, d\theta\, dr \tag{30}$$

$$= a Z_n^m(f_1) + b Z_n^m(f_2) \tag{31}$$

Therefore, Zernike moments are linear operators:

$$Z_n^m\left(af_1 + bf_2\right) = aZ_n^m(f_1) + bZ_n^m(f_2) \tag{32}$$

A.18 Frequency Propagation Linearity Proof

We use radial frequency propagation as an example to derive the *Linearity* proof, the angular frequency propagation *Linearity* can be derived similarly. For any complex number $u \in \mathbb{C}$, the polar-form identity, $|u|e^{i\arg(u)} = u$ holds, with the natural convention that the expression evaluates to zero when $u = 0$. Therefore, the propagation rule in Eqn. (7) can be equivalently rewritten as,

$$z_n^m \;\leftarrow\; z_n^m + \lambda\, z_{n-2}^m. \tag{33}$$

Equation (33) defines a linear mapping from the coefficient vector $\{z_{n-2}^m,\, z_n^m\}$ to the updated coefficient z_n^m. The full radial frequency propagation is executed by composing a finite sequence of such linear updates along the radial chain $\{n, n-2, n-4, \ldots\}$. Since a finite composition of linear operators remains linear, the linear frequency propagation is linear on the Zernike coefficient space.

Let $\mathcal{Z}$ denote the Zernike basis transform that maps a real-valued shape geometry field f to its Zernike coefficients $\{z_n^m\}$, $\mathcal{P}_\lambda$ defines the radial frequency propagation operation that is conditioned on the propagation coefficient λ. From Eqn. (33), we can learn that $\mathcal{P}_\lambda$ is a linear operation. The final shape geometry encoding $E_\lambda(f)$ can be represented as,

$$E_\lambda(f) := \mathcal{P}_\lambda\big(\mathcal{Z}(f)\big) \tag{34}$$

We can then derive that E_λ is a linear functional of the input field f. That is, for any scalars $a, b \in \mathbb{R}$ and any geometry fields f_1, f_2,

$$E_\lambda(af_1 + bf_2) = a\, E_\lambda(f_1) + b\, E_\lambda(f_2). \tag{35}$$

Therefore, the composition $E_\lambda = \mathcal{P}_\lambda \circ \mathcal{Z}$ is linear, which completes the proof.

A.19 Harmonic Pose Field Derivation

In this section, we derive the Zernike projection coefficient of a harmonic pose field $f_{\text{pose}}(r, \theta; m_p)$ onto the Zernike basis $V_n^m(r, \theta)$. The complex Zernike basis is defined as,

$$V_n^m(r, \theta) = R_n^{|m|}(r) \cdot e^{im\theta} \tag{36}$$

The harmonic pose field is defined as,

$$f_{\text{pose}}(r, \theta; m_p, \mathbf{p}) = \left(\sum_{k=1}^{K} p_k w_k(r)\right) \cos(m_p\theta) \tag{37}$$

Adopting Euler's formula: $\cos(m_p\theta) = \frac{1}{2}\left(e^{im_p\theta} + e^{-im_p\theta}\right)$, we can rewrite Eqn. (37) as:

$$f_{\text{pose}}(r, \theta; m_p, \mathbf{p}) = \frac{1}{2}\sum_{k=1}^{K} p_k w_k(r)\left(e^{im_p\theta} + e^{-im_p\theta}\right) \tag{38}$$

Project f_{pose} onto V_n^m,

$$a_n^m = \int_0^{2\pi}\int_0^1 f_{\text{pose}}(r, \theta; m_p) \cdot \overline{V_n^m(r, \theta)} \cdot r\, dr\, d\theta \tag{39}$$

Substitute f_{pose} and V_n^m,

$$a_n^m = \frac{1}{2}\sum_{k=1}^{K} p_k \int_0^{2\pi}\int_0^1 w_k(r)\left(e^{im_p\theta} + e^{-im_p\theta}\right)\cdot R_n^{|m|}(r)e^{-im\theta}\cdot r\,dr\,d\theta \tag{40}$$

Separate integrals,

$$a_n^m = \frac{1}{2}\sum_{k=1}^{K} p_k \left(\int_0^{2\pi} e^{i(m_p-m)\theta}d\theta + \int_0^{2\pi} e^{-i(m_p+m)\theta}d\theta\right)\cdot \int_0^1 w_k(r)R_n^{|m|}(r)r\,dr \tag{41}$$

We now evaluate:

$$\int_0^{2\pi} e^{ia\theta}d\theta = \begin{cases} 2\pi, & a = 0 \\ 0, & a \neq 0 \end{cases} \tag{42}$$

We can learn from Eqn. (42) that we get non-zero values only when $m = \pm m_p$. So,

$$a_n^m = \pi\sum_{k=1}^{K} p_k \int_0^1 w_k(r)R_n^{|m|}(r)r\,dr, \quad \text{if } m = \pm m_p \tag{43}$$

By further adding the Zernike normalization factor $\sqrt{\frac{n+1}{\pi}}$ we can finally get,

$$a_n^m = \begin{cases} \pi\sqrt{\frac{n+1}{\pi}}\sum_{k=1}^{K} p_k \int_0^1 w_k(r)\,R_n^{|m|}(r)\,r\,dr, & m = \pm m_p \\ 0, & \text{otherwise} \end{cases} \tag{44}$$

A.20 Frequency Propagation Ratio λ Empirical Selection

We enrich higher radial orders within each angular harmonic by cascading from the lower neighbor while preserving rotation behavior. The update is

$$z_n^m \leftarrow z_n^m + \lambda|z_{n-2}^m|\cdot e^{i\arg(Z_{n-2}^m)}, \tag{45}$$

applied along fixed m with valid n (*e.g.*, $n - |m|$ even). Writing $n = |m| + 2k$ and letting z_k^m denote the coefficient at radial index k, the cascaded forward map admits the closed form

$$z_k^{m,\text{mod}} = \sum_{j=0}^{k} \lambda^j\, z_{k-j}^m, \qquad k = 0, 1, \ldots \tag{46}$$

and is exactly invertible via the nilpotent shift operator S along k:

$$\mathbf{z}^m = (I - \lambda S)\,\mathbf{z}^{m,\text{mod}}. \tag{47}$$

We set $\lambda = 0.6$. This keeps the per-chain gain bounded by $\|(I - \lambda S)^{-1}\| \leq 1/(1-\lambda) = 2.5$ (well-conditioned), while providing a controlled high-frequency tail: the contribution propagated K steps decays as λ^K, e.g., $\lambda^{10} \approx 6 \times 10^{-3}$. Thus $\lambda = 0.6$ yields a meaningful spread across higher n without excessive amplification, and the exact demodulation $(I - \lambda S)$ remains numerically stable. We experimentally find that other λ setting yields similar experiment results on the acoustic target region control task [19].

A.21 Ablations on XShapeEnc Encoding Discriminability

To analyze the effect of FreqProp on shape geometry discriminability, we further compare t-SNE clustering results with and without frequency propagation. As shown in Fig. 21, both radial FreqProp and angular FreqProp preserve clear inter-class separation while maintaining compact intra-class structure, indicating that propagation does not distort the underlying geometric identity captured by the encoding. More importantly, FreqProp improves frequency diversity without sacrificing class-level organization in the latent space, suggesting a favorable trade-off between representational richness and discriminative stability. This behavior is consistent with our formulation: propagation redistributes information across neighboring Zernike modes while retaining structured harmonic relationships. In other words, Zernike basis encoding provides a stable geometric backbone, and FreqProp acts as a controlled enhancement mechanism that strengthens downstream learnability while preserving shape-aware discriminability.

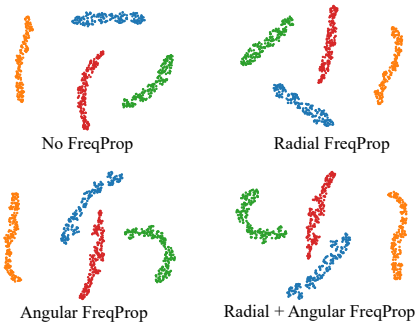


Figure 21: Shape geometry encoding comparison w/ w/o FreqProp.

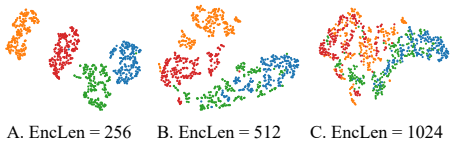


Figure 22: t-SNE [47] clustering result visualization on various encoding length

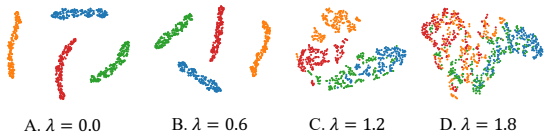


Figure 23: t-SNE [47] clustering result visualization on frequency propagation coefficient λ.

To further test the impact of encoding length, we further test the clustering result with encoding length in 256, 512 and 1024. From the result shown in Fig. 22, we can observe that, under shape augmentation, longer shape encoding length introduces more encoding feature difference. This is anticipated because longer encoding length captures more localized shape feature. Shape augmentation pronounces such localized shape feature variation. We further ablate the frequency propagation coefficient λ on the shape geometry discriminability. By varying $\lambda \in [0.0, 0.6, 1.2, 1.8]$, we visualize the corresponding clustering result in Fig. 23. We can clearly see higher λ introduces larger portion of low-frequency coefficient to high-frequency coefficient, the resulting encoding gradually loses shape discriminability.

A.22 Pose Encoding Linearity and Superposition Proof

The proposed harmonic pose encoding is linear with respect to the pose field and the subsequent Zernike projection. In particular, the resulting pose coefficients are linear functions of the pose vector, and therefore obey superposition. However, we do not enforce linearity with respect to physical pose parameters in Euclidean space (*e.g.*, translation), as such transformations are inherently non-linear in any fixed orthogonal basis on a bounded domain. Instead, our formulation embeds pose as a band-limited harmonic signal aligned with the Zernike basis, yielding a representation that is linear, controllable, and compatible with joint geometry–pose encoding.

Given the Zernike basis $V_n^{m_p}$ and two pose vectors $\mathbf{p_1}$ and $\mathbf{p_2}$, their linearly combined shape pose encoding by two scalars α and β can be represented by

$$a_n^{m_p}(\alpha \mathbf{p}_1 + \beta \mathbf{p}_2) = \langle f_{\mathcal{P}}(\cdot; \alpha \mathbf{p}_1 + \beta \mathbf{p}_2),\, V_n^m \rangle \tag{48}$$

Using the linear construction of the pose field in Eq. (8), we have

$$f_{\mathcal{P}}(r,\theta;m_p,\alpha\mathbf{p_1}+\beta\mathbf{p_2}) = \big(\sum_{k=1}^{K}(\alpha\cdot p_{1,k}+\beta p_{2,k})w_k(r)\big)\cdot\cos(m_p\theta) \tag{49}$$

$$= \alpha\sum_{k\in 1}^{K}\big(p_{1,k}w_k(r)\big)\cos(m_p\theta) + \beta\sum_{k\in 1}^{K}\big(p_{2,k}w_k(r)\big)\cos(m_p\theta) \tag{50}$$

$$= \alpha\cdot f_{\mathcal{P}}(r,\theta;\mathbf{p_1}) \;+\; \beta\, f_{\mathcal{P}}(r,\theta;\mathbf{p_2}). \tag{51}$$

Substituting Eqn. (51) into the inner product and using linearity of the integral (hence linearity of $\langle\cdot,\cdot\rangle$ in its first argument), we obtain,

$$a_n^{m_p}(\alpha\mathbf{p_1}+\beta\mathbf{p_2}) = \langle\alpha f_{\mathcal{P}}(\cdot;\mathbf{p}_1)+\beta f_{\mathcal{P}}(\cdot;\mathbf{p}_2),\, V_n^{m_p}\rangle \tag{52}$$

$$= \alpha\,\langle f_{\mathcal{P}}(\cdot;\mathbf{p}_1),\, V_n^{m_p}\rangle \;+\; \beta\,\langle f_{\mathcal{P}}(\cdot;\mathbf{p}_2),\, V_n^{m_p}\rangle \tag{53}$$

$$= \alpha\, a_n^{m_p}(\mathbf{p}_1) + \beta\, a_n^{m_p}(\mathbf{p}_2). \tag{54}$$

Thus the linearity and superposition property of shape pose encoding hold.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explicitly list the core contribution and scope of *XShapeEnc* in both abstract and introduction by text description, table and item list presentation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitation is presented in Sec. 4.5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: This paper can be treated as a theoretical paper, we provide detailed mathematical derivation and proof through the whole paper, especially in the Appendix (*e.g.*, A.19).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided detailed description of the implementation in Sec. A.15, Sec. A.14. We also provide the source code in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the source code of our method. All the datasets are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provided detailed experimental setting in Sec. A.15, Sec. A.14. We also provide the source code in the supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: In the source code provided, we explicitly show how the experiment is set and executed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Sec. A.14 and Sec. A.15, we provide all relevant details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is theoretical analysis, without potential societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets used (*e.g.*, inter-shape topological relation classification) are properly cited, and their usage follows the original terms and conditions.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Detailed *XShaoeCorpus* creation is given in Sec. A.8, we also provide the source code for it.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: the core method development of *XShapeEnc* in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.