Training-free Spatially Grounded Geometric Shape Encoding

Yuhang He

Senior Researcher
Microsoft Research, Vancouver Lab
Website: https://yuhanghe01.github.io
Email: yuhanghe@microsoft.com

Dec. 6, 2026

Contents

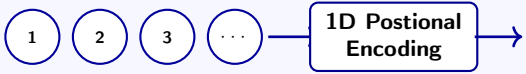
1. Motivation
2. Spatially Grounded 2D Shape Encoding
   - Mathematical Foundation: 2D Zernike Basis
   - Encoding Process
   - Applications
3. Summary

Spatially Discrete Point Encoding: Well Studied

Microsoft

1D: NLP

Word-token sequence



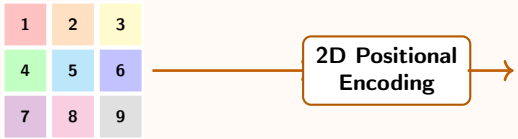
High-dimensional token embeddings



$\mathbf{e}_i \in \mathbb{R}^d$

2D Postional Encoding

$N \times N$ patch tokens

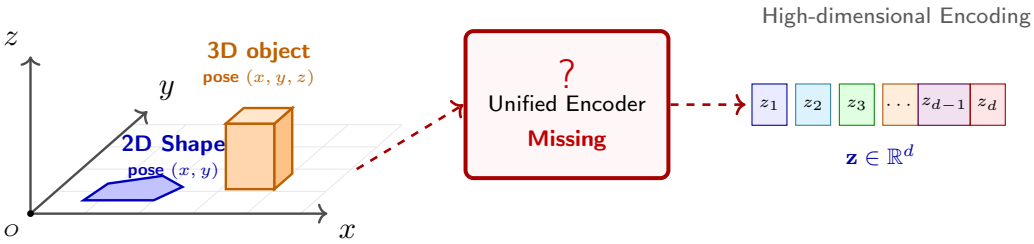


One high-dimensional vector per patch

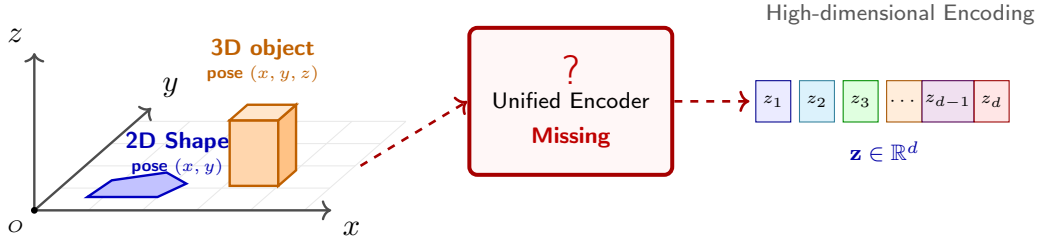


$\mathbf{v}_i \in \mathbb{R}^d$

What if Spatially Grounded 2D Shape or 3D Object?



What if Spatially Grounded 2D Shape or 3D Object?



why ignored!

Encoding by treating them as 2D image or 3D point is a lazy expedient. What if two shapes/objects have too large scale variance and spatial translate?

Five Properties We Anticipate

Microsoft

1. **General-purpose & training-free** – no learning required.
2. **Invertible & interpretable** – we can recover the shape from encoding.
3. **Controllable** – tunable emphasis between multi-components.
4. **General & efficient** – arbitrary shape/object, fast to compute.
5. **Adaptive** – applicable to downstream tasks.

XShapeEnc is built to satisfy all five simultaneously!

Representation: Geometry × Pose

$$S = A \cdot S_{\mathcal{G}} + b$$

- $S_{\mathcal{G}}$: the shape's (or 3D object) normalized geometry, sit inside a unit disk.
- A (2×2 matrix, or 3×3 matrix) and b (translation): the pose – scale, rotation, position.
- Flatten A, b into a pose vector $\mathbf{p}$.

Why split?

Geometry and pose behave differently – geometry is about *shape*, pose is about *where*. Encoding them with a shared, principled basis lets us mix them back together under control.

2D Zernike Basis: The Common Language



Frits Zernike: In the 1930s, he developed these polynomials for analyzing wavefronts on a circular lens. He received the 1953 Nobel Prize in Physics for demonstrating the phase-contrast method and inventing the phase-contrast microscope.

- A complete, orthogonal basis defined on the unit disk (r, θ):

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

- n: radial order (how much detail radially)
- m: angular frequency (how much detail around the circle)

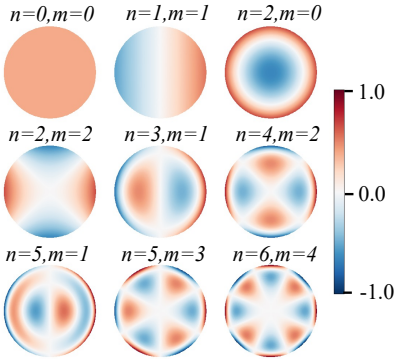


Figure: 2D Zernike basis visualization

Why Zernike, Not Fourier?

- Basis functions are polynomials, not sinusoids → more localized, compact shape description.
- Radial and angular detail are decoupled – tune one independently of the other.
- Orthogonality ⇒ Linearity: encoding of a composite shape = combination of its parts' encodings. No need to re-encode from scratch.

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

Encoding Geometry: Just a Projection

Given the geometry mask $f_{\mathcal{G}}(r, \theta)$, project it onto each Zernike basis function:

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

- Low-order coefficients $\rightarrow$ global outline.
- High-order coefficients $\rightarrow$ fine local detail.
- Stack all coefficients: $Z_{\mathcal{G}} = [z_0^0, z_1^{-1}, z_1^1, \ldots, z_N^M]$.

Encoding Geometry: Just a Projection

Given the geometry mask $f_{\mathcal{G}}(r, \theta)$, project it onto each Zernike basis function:

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

- Low-order coefficients $\rightarrow$ global outline.
- High-order coefficients $\rightarrow$ fine local detail.
- Stack all coefficients: $Z_{\mathcal{G}} = [z_0^0, z_1^{-1}, z_1^1, \ldots, z_N^M]$.

Two Desirable Properties

Linearity: encoding two composed shape geometries equals to linearly composing their separate encoding.

Rotation Equivariance: take the magnitude for a rotation-invariant encoding.

A Problem: Simple Shape Geometry Lacks High-Frequency

Microsoft

- Simple Shape Geometry leads to low-frequency dominated encoding (not good for neural network learning).
- A perfect circle only activates the *lowest* mode z_0^0 – everything else is zero.
- We need a trick that enriches high frequencies without breaking invertibility.

FreqProp: Borrowing Detail from Neighbor

Microsoft

$$z_n^m \leftarrow z_n^m + \lambda_r \, |z_{n-2}^m| \cdot e^{i \arg(z_{n-2}^m)} \text{ (radial FreqProp)}$$
$$z_n^m \leftarrow z_n^m + \lambda_a \, |z_n^{m-2}| \cdot e^{i \arg(z_n^{m-2})} \text{ (angular FreqProp)}$$

- Each coefficient borrows magnitude & phase from its lower-frequency radial neighbor ($n-2$) or angular neighbor ($m-2$).
- Geometric view: rotate & scale the higher basis by the lower one's phase/magnitude, then re-project.
- Influence decays exponentially with distance – nearby frequencies talk, far ones don't.
- Fully invertible: undo it by subtracting the propagated term, starting from z_0^0.

FreqProp: Borrowing Detail from Neighbor

$$z_n^m \leftarrow z_n^m + \lambda_r \, |z_{n-2}^m| \cdot e^{i \arg(z_{n-2}^m)} \text{ (radial FreqProp)}$$

$$z_n^m \leftarrow z_n^m + \lambda_a \, |z_n^{m-2}| \cdot e^{i \arg(z_n^{m-2})} \text{ (angular FreqProp)}$$

- Each coefficient borrows magnitude & phase from its lower-frequency radial neighbor $(n-2)$ or angular neighbor $(m-2)$.
- Geometric view: rotate & scale the higher basis by the lower one's phase/magnitude, then re-project.
- Influence decays exponentially with distance – nearby frequencies talk, far ones don't.
- Fully invertible: undo it by subtracting the propagated term, starting from z_0^0.

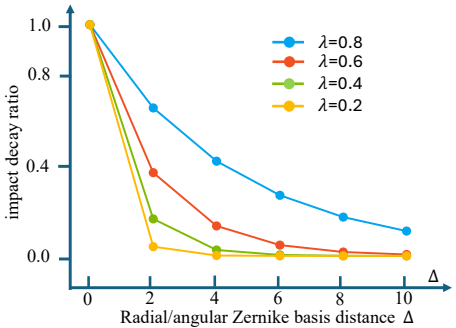


Figure: Exponential decay visualization.

FreqProp: Borrowing Detail from Neighbor

$$z_n^m \leftarrow z_n^m + \lambda_r \, |z_{n-2}^m| \cdot e^{i \arg(z_{n-2}^m)} \text{ (radial FreqProp)}$$

$$z_n^m \leftarrow z_n^m + \lambda_a \, |z_n^{m-2}| \cdot e^{i \arg(z_n^{m-2})} \text{ (angular FreqProp)}$$

- Each coefficient borrows magnitude & phase from its lower-frequency radial neighbor ($n-2$) or angular neighbor ($m-2$).
- Geometric view: rotate & scale the higher basis by the lower one's phase/magnitude, then re-project.
- Influence decays exponentially with distance – nearby frequencies talk, far ones don't.
- Fully invertible: undo it by subtracting the propagated term, starting from z_0^0.

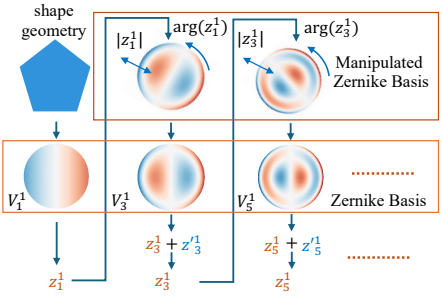


Figure: z_1^1 rotates and scales the next basis V_3^1 before it gets projected – a clean geometric picture of what Eqn. (FreqProp) is doing.

Pose Encoding: Pose as a Field

- Pose $\mathbf{p} \in \mathbb{R}^K$ is a handful of scalars (translation, scale, rotation).
- Trick: turn it into a harmonic pose field living on the unit disk, so it can be Zernike-encoded too:

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

- Each pose parameter p_k gets its own radial window $w_k(r)$ – mutually orthogonal, so nothing interferes.

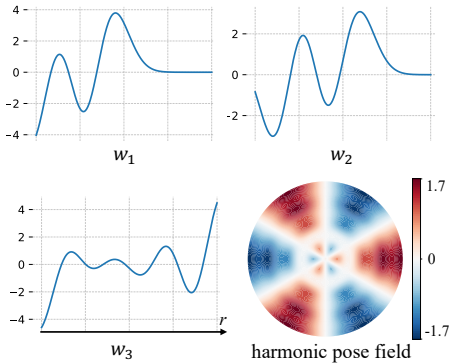


Figure: Orthonormal radial windows $w_k(r)$ (each capture one pose parameter) and the harmonic pose field.

Geometry-Pose Joint Encoding: Pose Effects on Phase

$$Z_{\mathcal{GP}} = \Big[\overset{\text{Geometry}}{\underset{\downarrow}{z_0^0}} \cdot e^{-i \overset{\text{Pose}}{\underset{\downarrow}{a_0^0}}}, \ldots, z_n^m \cdot e^{-i a_n^m}, \ldots \Big] \tag{1}$$

- Geometry coefficient z_n^m (magnitude) carries shape detail.
- Pose coefficient a_n^m rotates that coefficient in the complex plane – it modulates *phase*, not magnitude.
- No scale mismatch, no mutual interference, and the shared Zernike basis is used once, efficiently.

Controllable Emphasis via β

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

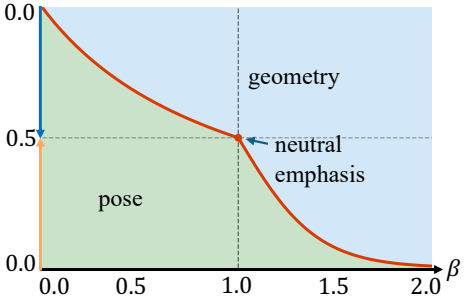
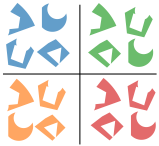


Figure: β controlled geometry-pose encoding emphasis

- $\eta(\beta) = \exp(-5 \cdot (\beta - 1))$;
- $\beta = 0$, it degenerates to pose encoding;
- $\beta = 2$, it is close to geometry encoding;

Joint Encoding Responds to β as Designed



100×100 *area*

A. 16 spatially grounded
4-class geometric shapes

Joint Encoding Responds to β as Designed

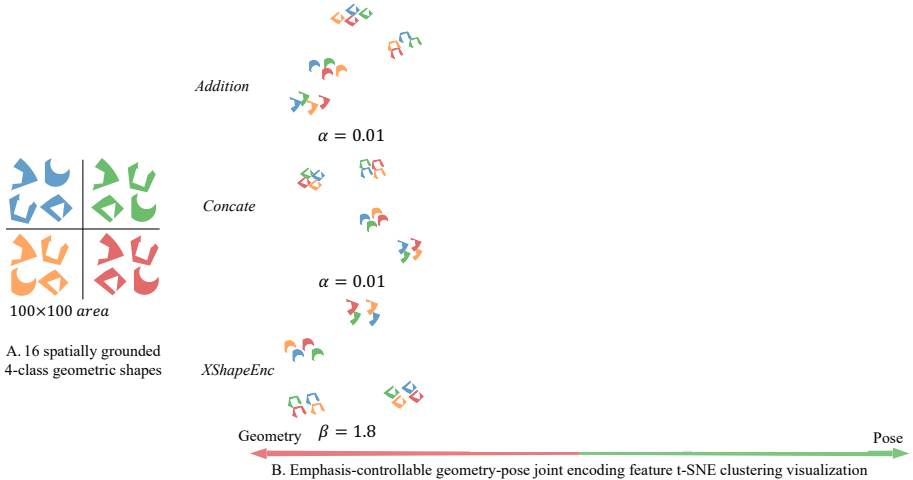


Figure: Small β clusters by *pose*; large β clusters by *geometry* – the control knob works exactly as intended.

Joint Encoding Responds to β as Designed

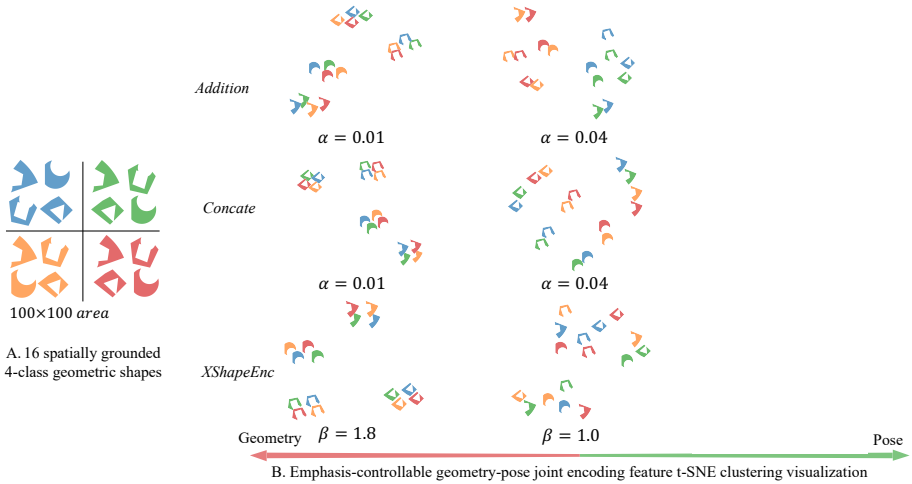


Figure: Small β clusters by *pose*; large β clusters by *geometry* – the control knob works exactly as intended.

Joint Encoding Responds to β as Designed

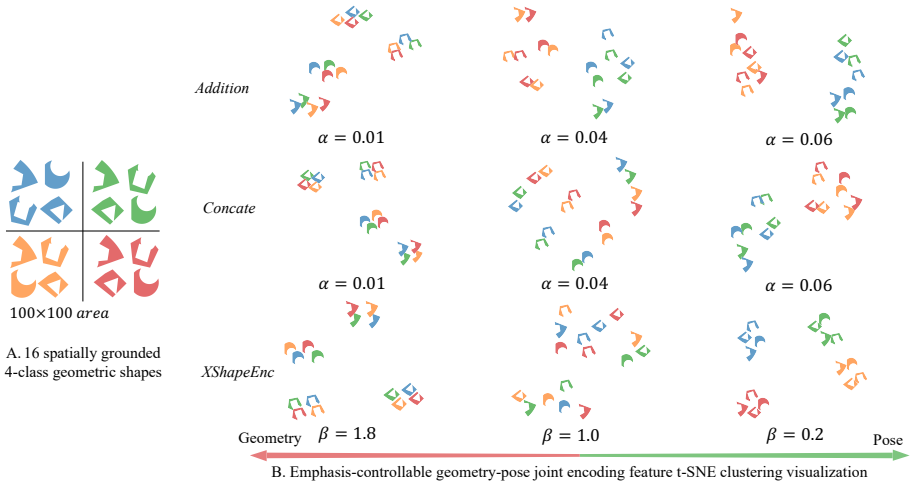
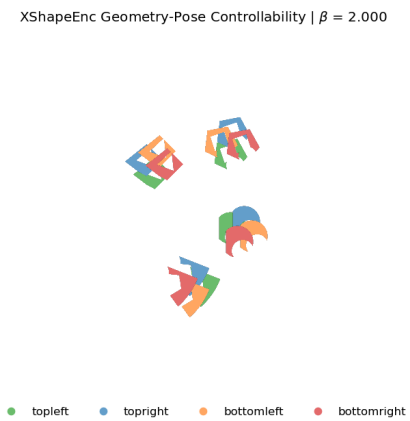
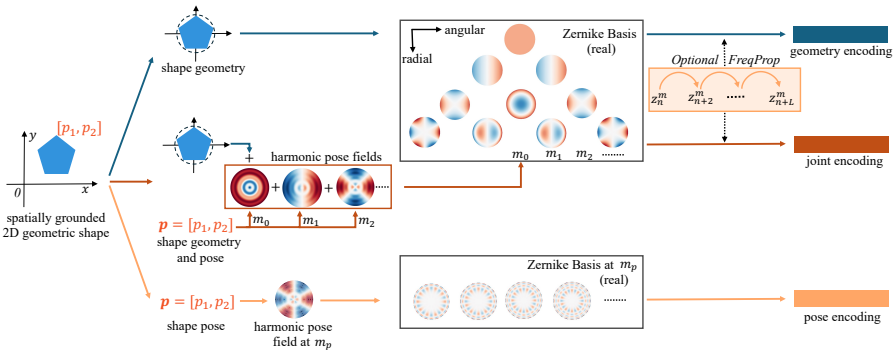


Figure: Small β clusters by *pose*; large β clusters by *geometry* – the control knob works exactly as intended.

An Inline Animation of β Controllability



The Full Algorithm, End to End

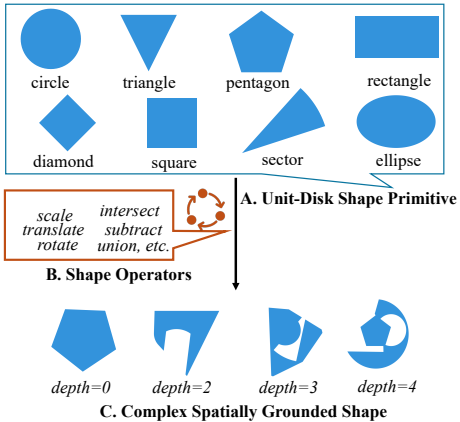


- Decompose $S \to (S_{\mathcal{G}}, S_{\mathcal{P}})$
- Encode geometry and/or pose, optionally fuse via phase modulation
- Optionally run *FreqProp* to enrich high frequencies
- All steps vectorized $\Rightarrow$ near-linear time, no training

XShapeCorpus: A New Benchmark

Microsoft

- No existing dataset pairs *arbitrary* geometry with *pose*.
- 8 shape primitives (circle, square, triangle, . . .)
- Built up via unary ops (scale/translate/rotate) and binary ops (union, subtract, intersect, xor, convex hull).
- "Depth" = number of operations → complexity knob.
- 10 depths × 100 shapes = 1,000 shapes.



Invertibility: Can We Get the Shape Back?

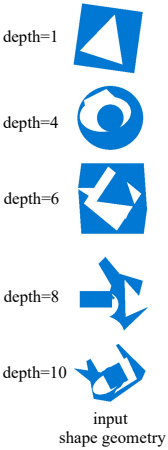


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

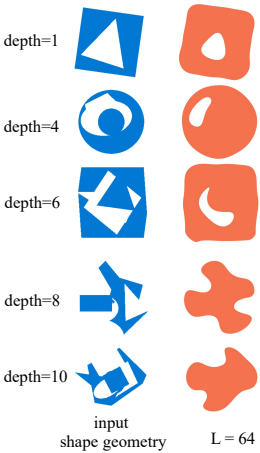


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

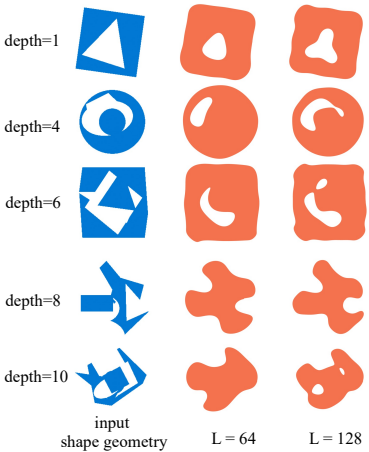


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

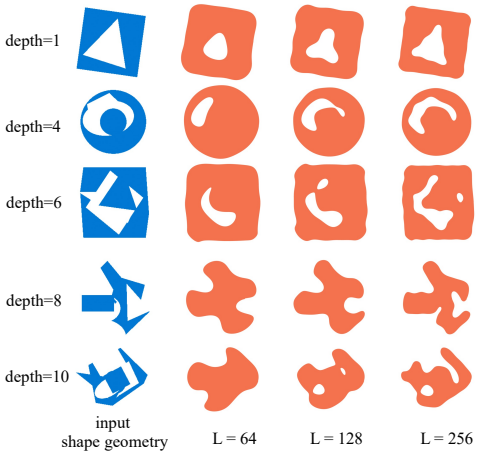


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

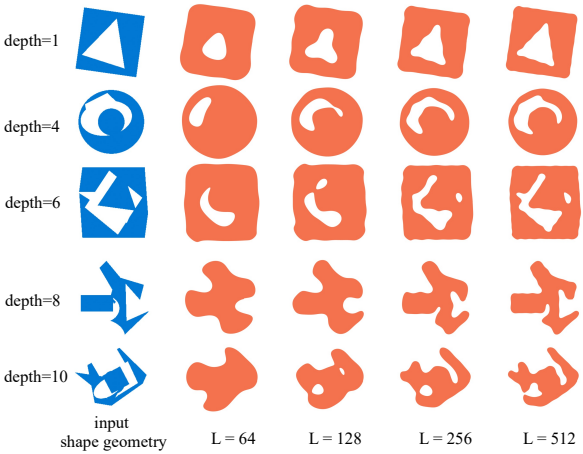


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

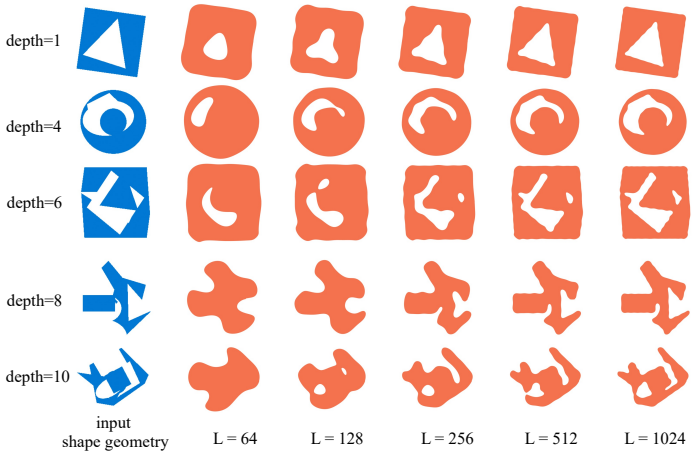


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

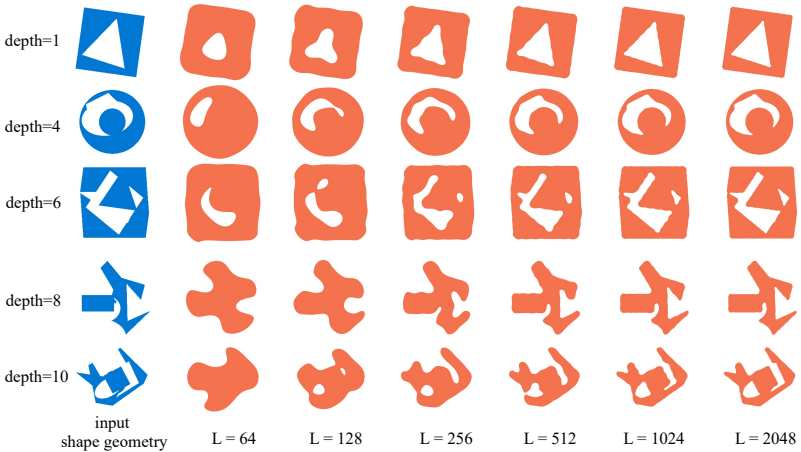


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Invertibility: Can We Get the Shape Back?

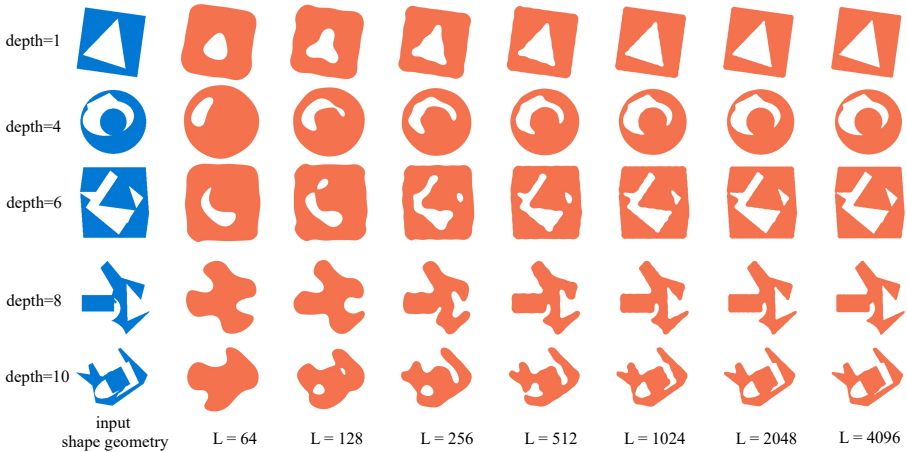


Figure: Reconstructions at increasing encoding length. More coefficients → sharper recovery.

Discriminability: Geometry Survives Augmentation

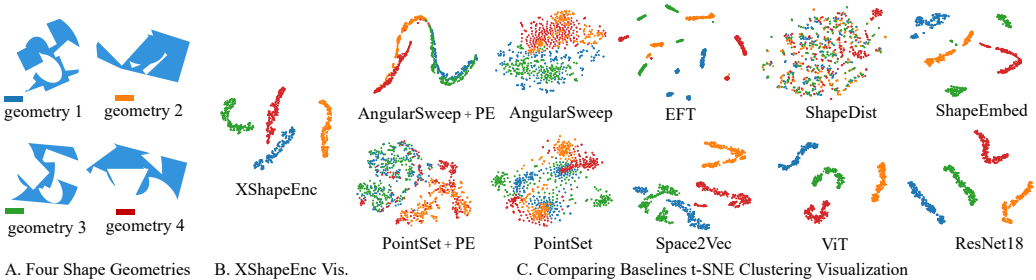
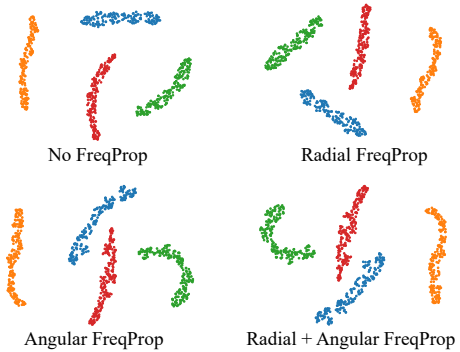


Figure: t-SNE of 4 complex shapes $\times$ 200 augmentations each. XShapeEnc keeps classes tightly separated; plain 2D positional encoding collapses everything together.

FreqProp Doesn't Hurt Discriminability



- Both radial and angular *FreqProp* preserve clean inter-class separation.
- Enriches frequency content *without* destroying the structure that makes shapes distinguishable.

Pose Discriminability: Shapes Cluster by Location

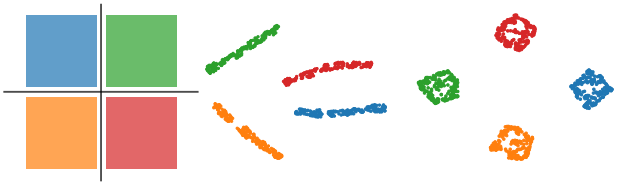


Figure: Each region is randomly place 100 2D positions. Unlike 2D PE (raw coordinates), XShapeEnc organizes poses along smooth, continuous manifolds – nearby poses stay nearby in code space.

- **MPEG-7 Dataset**: 70 classes, like animal shape (*e.g.*, cattle, beetle, camel), people (*e.g.*, children, face) to utensils (*e.g.*, fork, spoon).

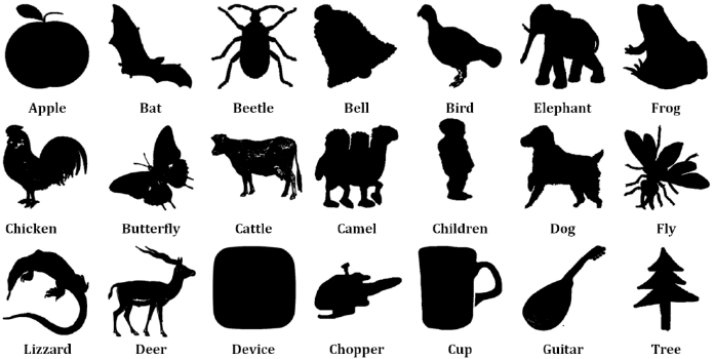
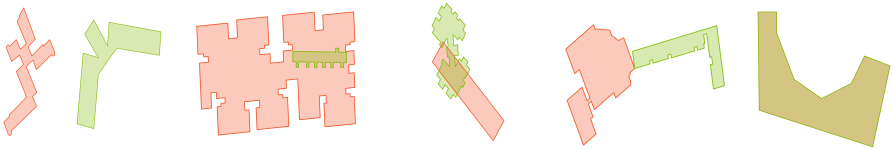


Figure: Sample Shapes from MPEG-7 Dataset.

Application 1: Shape Retrieval

Method Name	MPEG-7 (mAP)
PointSet	0.10
ShapeContexts (PAMI'02)	0.45
Space2Vec (ICLR'20)	0.47
ShapeDist (ToG'02)	0.11
AngularSweep (ICASSP'25)	0.26
ShapeEmbed (NeurIPS'25)	0.41
ResNet18 (CVPR'16)	0.56
ViT (ICLR'21)	0.57
CLIP (ICML'21)	**0.64**
Plain Zernike	0.53
XShapeEnc	0.59

Application 2: Topological Relation Classification



- Building topdown views from Singapore and New York;
- Five inter-shape relationship classes;
- Each accounts for both shape geometry and pose;
- 15,000 training set and 3,000 testing set per city;
- We train a two-layer MLP classification layer.

Application 2: Topological Relation Classification

Microsoft

Method	Singapore	New York
PointSet	0.670	0.564
ShapeContexts (PAMI'02)	0.581	0.525
AngularSweep (ICASSP'25)	0.606	0.546
Space2Vec (ICLR'20)	0.706	0.632
ResNet18 (CVPR'16)	0.674	0.753
ViT (ICLR'21)	0.669	0.752
CLIP (ICML'21)	0.700	**0.779**
Poly2Vec (ICML'25)	0.702	0.684
XShapeEnc (w/o *FreqProp*)	0.740	0.753
XShapeEnc	**0.760**	0.768

Application 2: Topological Relation Classification

Microsoft

Method	Singapore	New York
PointSet	0.670	0.564
ShapeContexts (PAMI'02)	0.581	0.525
AngularSweep (ICASSP'25)	0.606	0.546
Space2Vec (ICLR'20)	0.706	0.632
ResNet18 (CVPR'16)	0.674	0.753
ViT (ICLR'21)	0.669	0.752
CLIP (ICML'21)	0.700	**0.779**
Poly2Vec (ICML'25)	0.702	0.684
XShapeEnc (w/o *FreqProp*)	0.740	0.753
XShapeEnc	**0.760**	0.768

Warning

Image based inter-shape topological data representation is fragile because it does not account for shape scale, physical distance.

Summary

1. **General-purpose & training-free** – *consider it if data is limited.*
2. **Invertible & interpretable** – *consider it if require interpretability.*
3. **Controllable** – *consider it if require controlled encoding.*
4. **General & efficient** – *consider it if to build a unified layer.*

Things I am Thinking About

- *Extend to encode anything (Encode Anything)?*
- *Extend to encode 4D?.*
- *Application in Physical AI.*

Microsoft

Thank You

Paper: https://arxiv.org/pdf/2604.07522 (accepted by NeurIPS 26)
Code: https://github.com/yuhanghe01/XShapeEnc