TRAINING-FREE SPATIALLY GROUNDED 3D OBJECT ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present *X3DEnc*, a training-free framework for spatially grounded 3D objects encoding. Our central insight is that such a 3D object is characterized not only by its geometry, but also by its appearance and spatial pose. Different downstream tasks may require different relative emphasis on these components. *X3DEnc* decomposes a 3D object into normalized geometry and color fields within the unit ball and a pose transformation. It then lifts the finite-dimensional pose vector to a harmonic field and projects geometry, color, and pose onto a shared 3D Zernike basis. This common spectral coordinate system supports separate or joint encoding. The geometry-color-pose joint encoding allow emphasis-adjustable encoding. *X3DEnc* accommodates voxel, signed-tetrahedral, and surface-triangle representations, covering volumetric solids, watertight meshes, hollow objects, and open surfaces. Theoretical analysis and various experiments demonstrate that *X3DEnc* can redirect similarity toward geometry, color, or pose while preserving a single encoding formulation. Rather than competing with neural encoders, *X3DEnc* provides an alternative when training data are limited or when interpretable, jointly controllable multi-component encoding is required.

1 INTRODUCTION

Encoding a 3D object into a compact vector is a fundamental operation in recognition (Su et al., 2015; Qi et al., 2017a), retrieval (Novotni & Klein, 2003; Liu et al., 2023), generation (Nichol et al., 2022), and geometric reasoning (Zhao et al., 2021; Zhou et al., 2024). Unlike 2D shapes, a 3D object is often represented as an unordered point cloud (Geiger et al., 2012; Qi et al., 2017a; Wu et al., 2015; Dai et al., 2017), an irregular polygonal mesh (Hanocka et al., 2019), a volumetric grid (Maturana & Scherer, 2015) or a parametric CAD model (Wu et al., 2015). A spatially grounded 3D object additionally carries pose information such as translation, rotation and scale, and optionally contains color or other surface attributes. A desirable 3D object encoding framework should be able to encode all these information within a unified framework, reconciling heterogeneous data structures and retaining both intrinsic geometry and extrinsic spatial information.

Modern 3D object encoders are neural network based. Point-based networks process unordered samples, mesh networks exploit surface connectivity, and voxel or multi-view methods adapt convolutional architectures to process regular grids or rendered images (Su et al., 2015; Maturana & Scherer, 2015; Qi et al., 2017a; Hanocka et al., 2019; Zhao et al., 2021). More recent self-supervised and multimodal systems obtain increasingly transferable representations by pretraining on large collections of 3D objects, point clouds, images and text datasets (Yu et al., 2022; Pang et al., 2022; Liu et al., 2023; Zhou et al., 2024; Wu et al., 2025). Although the promising task performance, deep neural network based encoding lacks interpretability, encoding size adaptability and transferability as it merely reflects the training data distribution. Moreover, these methods fail to explicitly encode 3D object color and pose. The desired balance between 3D object geometry, color and pose often requires architectural modification and retraining the whole neural network from scratch.

In this paper, we complementarily investigate:

Can we construct a training-free encoding of a spatially grounded 3D object that unifies geometry, color, and pose while allowing their relative emphasis to be adjusted for different downstream tasks?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

More specifically, we seek a deterministic and interpretable framework for encoding spatially grounded 3D objects that remains applicable across diverse downstream tasks. The framework should represent geometry, color, and pose either separately or within a unified descriptor, while allowing their relative emphasis to be adjusted without architectural modification or retraining. By avoiding data-driven representation learning, it removes dependence on large-scale 3D training corpora and task-specific optimization, while making its inductive bias explicit through the selected mathematical basis and encoding construction.

To this end, we introduce *X3DEnc*, a training-free framework for encoding spatially grounded 3D objects. Building on classical 3D Zernike spectral descriptors (Novotni & Klein, 2003), *X3DEnc* separates an object into three conceptually distinct components: normalized geometry contained in the origin-centered unit ball, optional surface color, and a pose transformation comprising translation, rotation, and scale. Geometry and color are represented as fields on the unit ball, while the finite-dimensional pose vector is lifted to a harmonic field, allowing all three components to be projected onto a common Zernike coordinate system. In addition to producing separate geometry, color, and pose encodings, *X3DEnc* constructs a unified descriptor whose relative emphasis on these components can be adjusted without learning an additional fusion model. Rather than replacing task-optimized neural encoders, it provides a complementary, dataset-independent representation for settings with limited training data or requiring interpretable and controllable multi-component encoding. In summary, *X3DEnc* exhibits the following desirable properties:

X3DEnc Desirable Properties

1. **General-purpose and Training-Free**: *X3DEnc* involves no training and is task-agnostic.
2. **Multiple Format Support**: *X3DEnc* flexibly supports solid/hollow, close/open surface 3D object encoding.
3. **Mathematical Interpretability**: Each *X3DEnc* coefficient corresponds to a prescribed radial and angular Zernike mode, providing an explicit frequency-domain interpretation.
4. **Applicability**: *X3DEnc* is widely applicable to various 3D downstream tasks when multi-component encoding is required.
5. **Adaptability**: *X3DEnc* adaptively supports different encoding length, different encoding emphasis between geometry-color and pose.

Our main contributions are summarized as follows,

- We propose *X3DEnc*, a general-purpose and entirely training-free framework that encodes 3D object geometry, object color and spatial pose within the same 3D Zernike basis.
- We flexibly support voxel, signed tetrahedra, and surface triangles based 3D object representations, covering volumetric solids, watertight meshes, open surfaces, and hollow objects.
- We establish controllable joint geometry-color-pose encoding, resulting in the wide applicability to various downstream tasks.
- We demonstrate *X3DEnc* advantage through both theoretical foundation discussion, controlled experimental analysis and downstream tasks across 3D object retrieval, classification and cross-domain generalization test.

2 Related Work

3D Object Representation. 3D objects are commonly represented as point cloud (Geiger et al., 2012), polygonal meshes (Wu et al., 2015), voxel grids or parametric CAD models. While being compact and flexible, point cloud does not explicitly encode surface connectivity (Geiger et al., 2012; Dai et al., 2017; Qi et al., 2017a). Meshes augment vertices with edges and polygonal faces, thereby preserving local topology and supporting accurate surface rendering and processing (Hanocka et al., 2019). Voxel representations discretize space into a regular occupancy or attribute grid, making them directly compatible with volumetric convolutions at the cost of memory consumption that grows cubically with spatial resolution (Maturana & Scherer, 2015). CAD representations retain design-level structure through geometric primitives, parametric surfaces, boundary representations, or constructive operations. These formats expose different geometric information and computational trade-offs, motivating *X3DEnc* design that can operate consistently across different 3D representation.

Training-free 3D Object Encoding. Classic training-free 3D object encoders extensively incorporate spectral, statistical and view-based methods to encode 3D objects. Plain 3D Zernike descriptor projects 3D object onto orthogonal radial-angular bases (Novotni & Klein, 2003; Kazhdan et al., 2003); Spherical harmonic descriptor (SHD (Kazhdan et al., 2003)) expands occupancy sampled on concentric spherical shells; Shape Distributions (D2 (Osada et al., 2002)) summarize pairwise distances between sampled surface points, while the Ensemble of Shape Functions (ESF (Wohlkinger & Vincze, 2011)) combines multiple global geometric histograms (Osada et al., 2002; Wohlkinger & Vincze, 2011); The Light Field Descriptor (LFD (Chen et al., 2003)) represents an object through multiple rendered silhouettes, whereas the Viewpoint Feature Histogram (VFH) aggregates surface-normal relationships together with sensor-relative viewpoint information (Chen et al., 2003; Rusu et al., 2010). These classic training-free encoders primarily describe 3D shape geometry and do not support unified encoding accommodating both geometry, color, and spatial pose. Our *X3DEnc* flexibly supports either separate or joint encoding across 3D object geometry, color and pose, accommodating various encoding requests.

Neural 3D Object Encoding. The growing availability of 3D data corpora (Wu et al., 2015; Uy et al., 2019; Dai et al., 2017) has enabled neural encoders to learn task-discriminative object representations. PointNet (Qi et al., 2017a) directly processes unordered point sets through permutation-invariant aggregation; PointNet++ (Qi et al., 2017b) hierarchically aggregates local neighborhoods; DGCNN (Wang et al., 2019) dynamically constructs feature-space graphs; and Point Transformer (Zhao et al., 2021) models geometric interactions through attention. These approaches achieve strong empirical performance, but their latent coordinates generally have no prescribed geometric meaning, and their behavior remains coupled to the chosen architecture, training datasets. *X3DEnc* addresses a complementary objective: it projects geometry, color, and harmonic pose fields onto a fixed orthogonal Zernike coordinate system, producing a deterministic and coefficient-interpretable representation whose components can be encoded separately or jointly re-emphasized at inference time without learning a new fusion model.

3 X3DEnc: Training-free 3D Object Encoding

3.1 Problem Definition

A spatially grounded 3D object indicates a 3D object Ω_{3D} and the 3D object associates with geometric structure, color and spatial pose. The spatial position represents the 3D object's spatial localization and orientation in the 3D coordinate system. Our goal is to propose a training-free and general-purpose 3D encoding framework $\boldsymbol{\mathcal{F}}$ that encodes Ω_{3D} into high-dimensional space.

To ensure Ω_{3D} is encodable by *X3DEnc*, we decompose Ω_{3D} into three orthogonal components: the normalized 3D object geometry $\Omega_{\mathcal{G}}$ within the unit ball centering at origin point, 3D object color $\Omega_{\mathcal{C}}$ that also lies within the unit ball, 3D object pose $\Omega_{\mathcal{P}}$ that transforms the normalized 3D object geometry to its original spatial position and size, $\Omega_{3D} := [\Omega_{\mathcal{G}}; \Omega_{\mathcal{C}}; \Omega_{\mathcal{P}}]$. The 3D object geometry $\Omega_{\mathcal{G}}$ indicates the 3D object geometric structure. The 3D object color $\Omega_{\mathcal{C}}$ can be represented by either RGB or gray-scale color. The 3D object pose $\Omega_{\mathcal{P}}$ usually consists of a linear 3D position translation, rotation and size scaling. The original 3D object Ω_{3D} can be reconstructed from 3D object geometry $\Omega_{\mathcal{G}}$ and 3D object color $\Omega_{\mathcal{C}}$ by sequentially applying linear position translation, rotation and scaling, $\Omega_{3D} \xleftarrow{\Omega_{\mathcal{P}}} \Omega_{\mathcal{G}} + \Omega_{\mathcal{C}}$. *X3DEnc* is capable of encoding the 3D object geometry $\Omega_{\mathcal{G}}$, 3D object color $\Omega_{\mathcal{C}}$ and 3D object pose $\Omega_{\mathcal{P}}$ either jointly or separately.

$$E_{\mathcal{G}} = \boldsymbol{\mathcal{F}}(\Omega_{\mathcal{G}}); \quad E_{\mathcal{C}} = \boldsymbol{\mathcal{F}}(\Omega_{\mathcal{C}}); \quad E_{\mathcal{P}} = \boldsymbol{\mathcal{F}}(\Omega_{\mathcal{P}}); \quad E_{\mathcal{O}} = \boldsymbol{\mathcal{F}}(\Omega_{\mathcal{G}}, \Omega_{\mathcal{C}}, \Omega_{\mathcal{P}}); \tag{1}$$

where $E_{\mathcal{G}}$, $E_{\mathcal{C}}$, $E_{\mathcal{P}}$, and $E_{\mathcal{O}}$ denote the geometry, color, pose, and joint encodings, respectively. $E_{\mathcal{G}}, E_{\mathcal{C}}, E_{\mathcal{O}} \in \mathbb{R}^D$. By supporting voxel, signed-tetrahedral, and surface-triangle representations (Sec. 3.3), *X3DEnc* accommodates volumetric solids, watertight meshes, hollow objects, and open surfaces within a common formulation. The corresponding geometry, color, and pose fields are projected onto the same 3D Zernike basis, enabling them to be encoded within the same unified, training-free framework.

3.2 Encoding Mathematical Tool: 3D Zernike Bases

Defined in the 3D unit ball $\mathbb{B}$, 3D Zernike bases provide a set of basis in spherical coordinate. Each basis holds designated radial and spherical frequencies in the 3D space. These bases are mutually orthogonal and form a complete and compact basis set. By projecting the 3D data (see Sec. §3.1) onto the 3D Zernike bases, we obtain the per-basis coefficient describing the input at non-overlapping frequency bands. Flattening these coefficients into one vector gives us the final encoding. Formally, a typical 3D Zernike basis within the unit ball $\mathbb{B}$ spherical coordinate (r, θ, ϕ) is defined as,

$$Z_{nl}^{m}(r, \theta, \phi) = R_{nl}(r)\, Y_{l}^{m}(\theta, \phi) \tag{2}$$

where $r \in [0, 1]$, $\theta \in [0, 2\pi]$, and $\phi \in [0, \pi]$. $n, l, m \in \mathbb{Z}$. $R_{nl}(r)$ is a radial Zernike polynomial, and $Y_{l}^{m}(\theta, \phi)$ is a spherical harmonic of degree l and order m. Overall, n controls the polynomial order and spatial detail; l controls the angular frequency over the sphere; and $|m|$ controls the azimuthal variation around the polar axis, while the sign of m determines the corresponding complex phase. One 3D Zernike basis is shown in Fig. 1, from which we can see how the three indices jointly control basis. Detailed definition of $R_{nl}(r)$ and $Y_{l}^{m}(\theta, \phi)$ is given in Sec A.1 of Appendix.

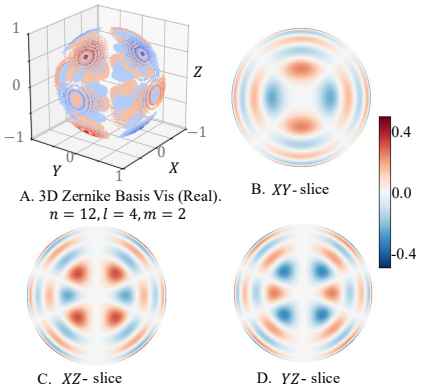


Figure 1: 3D Zernike basis real part with $n = 12$, $l = 4$ and $m = 2$ visualization in both Cartesian coordinate. We visualize in both 3D space (subfigure A) and three slices at three planes (subfigure B, C and D). More explanation is given in Sec. A.2 of Appendix.

If $n \geqslant 0$, $0 \leqslant l \leqslant n$, $n - l$ is even, and $-l \leqslant m \leqslant l$, any two 3D Zernike bases are orthogonal,

$$\int_0^1 \int_0^{2\pi} \int_0^{\pi} Z_{nl}^{m}(r, \theta, \phi)\overline{Z_{n'l'}^{m'}(r, \theta, \phi)}\, r^2 \sin\phi \, d\phi \, d\theta \, dr = \delta_{nn'}\delta_{ll'}\delta_{mm'} \tag{3}$$

where δ denotes the Kronecker delta function, $\delta_{nn'} = 1$ if $n = n'$, otherwise $\delta_{nn'} = 0$ (the same applies to l and m). The detailed mathematical proof is given in Sec. A.3 of Appendix. By iterating over all appropriate n, l and m from $n = 0$ and ensuring the orthogonality described in Eqn. (3), we can construct the 3D Zernike basis set $\mathcal{Z}$, $\mathcal{Z} = \{Z_{0,0}^{0}, Z_{2,0}^{0}, \cdots, Z_{n,l}^{m}, \cdots, Z_{N,L}^{M}\}$. For a given 3D object field f_{3D} within the unit ball, we can get the encoding coefficients by projecting it onto $\mathcal{Z}$,

$$E = [e_{0,0}^{0}, e_{2,0}^{0}, \ldots, e_{n,l}^{m}, \ldots, e_{N,L}^{M}], \quad e_{n,l}^{m} = \int_0^1 \int_0^{2\pi} \int_0^{\pi} f_{3D}\, \overline{Z_{n,l}^{m}}\, r^2 \sin\phi \, d\phi \, d\theta \, dr \tag{4}$$

where the 3D object field f_{3D} can be either 3D object geometric structure, color, or harmonic pose field. The resulting encoding in Eqn. (4) provides a compact multiscale description of the 3D object: lower-order coefficients capture coarse structure, whereas higher-order coefficients describe fine details. 3D Zernike basis exhibits favorable properties such as linearity, rotation equivariance and invertibility. More discussion are deferred to Sec. A.1 of Appendix.

3.3 3D Object Geometry and Color Representation and Encoding

Voxel Representation V_{3D} breaks down the 3D space into regular grid. Each grid is assigned with an occupancy, density or color value. Voxel representation is ideal for 3D convolutional processing and volumetric integration, but its memory cost grows cubically with resolution, and the resulting surfaces often appears blocky.

Signed Tetrahedra Representation T_{3D} decomposes an enclosed 3D object into non-overlapping oriented tetrahedra T_k with $\Omega = \sum_k \text{sign}(k) \cdot T_k$. Each tetrahedron is defined by four vertices: three vertices on the object surface and one origin vertex O, $T_k = (T_{A_k}, T_{B_k}, T_{C_k}, O)$. Tetrahedra-based representation introduces no voxelization process, it fully represents the original input 3D object because it directly adopts the on the surface sampled points to construct tetrahedrons. Moreover, since the 3D object is represented by the union of the signed tetrahedrons, it can only be used

to represent watertight and solid 3D object (Houdayer & Koehl, 2022). Specifically, $\text{sign}(k) = \text{sign}\big((T_{A_k} - O) \cdot \big((T_{B_k} - O) \times (T_{C_k} - O)\big)\big)$ is the sign of the tetrahedron's oriented volume. It equals $+1$ when the ordered face $(T_{A_k}, T_{B_k}, T_{C_k})$ is oriented outward relative to O and -1 otherwise, allowing negatively oriented tetrahedra to cancel excess volume in the signed decomposition.

Surface Triangle Representation F_{3D} models the boundary $\partial\Omega$ as a collection of non-overlapping, unsigned triangular faces $\{F_k\}$, where each face is defined by three surface vertices. This representation compactly captures 3D surface geometry and, because it does not require a signed volumetric decomposition, naturally accommodates both open surfaces and hollow structures.

In summary, the three 3D object representations can be mathematically represented as,

$$\Omega_{\mathcal{G}} = \begin{cases} \sum_k V_k, & \text{Voxel} \\ \sum_k \text{sign}(k) \cdot T_k, & \text{Signed Tetrahedra} \\ \sum_k F_k, & \text{Surface Triangles} \end{cases} \tag{5}$$

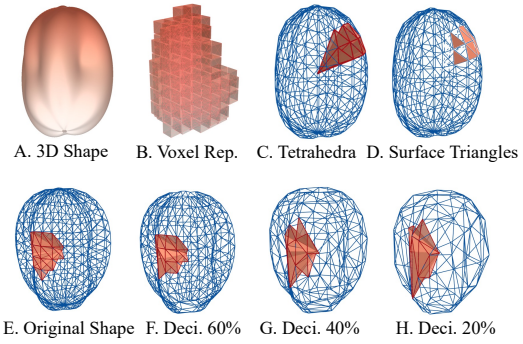


Figure 2: Top: three 3D object geometry representation methods visualiaztion. Bottom: Tetrahedra and Surface Triangle based representation can adjust the encoding complexity by controlling the triangles on the 3D object geometry surface.

The three representations are compared in Table 3 Sec. A.8 of Appendix and visualized in Fig. 2. Each representation offers a distinct trade-off among geometric fidelity, representation applicability, memory consumption, and computational cost. For the signed-tetrahedral and surface-triangle representations, encoding complexity can be adjusted directly by varying the number of retained surface triangles, as illustrated in the bottom row of Fig. 2. If the 3D object contains color information (either gray-scale or RGB), we construct a multichannel representation in which the first channel encodes geometry and the remaining channels encode color. For gray-scale input, the second channel stores intensity; for RGB input, the second through fourth channels store the red, green, and blue values, respectively.

Using any representation in Eqn. (5) and projecting to 3D Zernike basis in Eqn. (4), the 3D object geometry encoding can be expressed as,

$$E_{\mathcal{G}} = [g^0_{0,0}, g^0_{2,0}, \ldots, g^m_{n,l}, \ldots, g^M_{N,L}], \quad g^m_{n,l} \in \mathbb{C} \tag{6}$$

The 3D object color encoding can be accordingly expressed as,

$$E_{\mathcal{C}} = [c^0_{0,0}, c^0_{2,0}, \ldots, c^m_{n,l}, \ldots, c^M_{N,L}], \quad c^m_{n,l} \in \mathbb{C} \tag{7}$$

3.4 3D Pose Representation and Encoding

Following the harmonic field principle in XShapeEnc (He, 2026), we lift the 3D pose vector $\Omega_{\mathcal{P}} := \mathbf{p} = [p_1, \ldots, p_K]^\top \in \mathbb{R}^K$ to a scalar field within the unit ball $\mathbb{R}^{3D}$. We reserve the spherical harmonic channel (l_p, m_p) so that the resulting field and coefficients are real-valued,

$$f_{\mathcal{P}}(r, \theta, \phi; \mathbf{p}) = \left(\sum_{k=1}^{K} p_k w_k(r) \right) Y^{m_p}_{l_p}(\theta, \phi) \tag{8}$$

Each pose value p_k is associated with a radial window $w_k(r)$ initialized from a two-bump Gaussian. The window bank is orthonormalized by Gram–Schmidt under the 3D radial inner product $\langle u, v \rangle_{r^2} = \int_0^1 u(r)v(r)r^2\, dr$, so that $\langle w_i, w_k \rangle_{r^2} = \delta_{ik}$. We select L admissible radial orders $n_j = l_p + 2(j-1)$, $j = 1, \ldots, L$, and project the harmonic pose field in Eqn. (8) onto the modes $\{Z^{m_p}_{n_j l_p}\}_{j=1}^{L}$. The resulting 3D object pose encoding is,

$$E_{\mathcal{P}} = [p^{m_p}_{l_p,l_p}, p^{m_p}_{l_p+2,l_p}, \cdots, p^{m_p}_{l_p+2N,l_p}], \quad p^{m_p}_{l_p,l_p} \in \mathbb{R} \tag{9}$$

3.5 3D Geometry, Color and Pose Joint Encoding

To enable geometry-color-pose joint encoding with adjustable relative emphasis, we introduce three independent emphasis gain weights for geometry, color and pose separately,

$$w_g, w_c, w_p \in [0, 2], \qquad 0 \xleftarrow{\text{suppress}} 1 \xrightarrow{\text{emphasize}} 2 \tag{10}$$

where w_g, w_c and w_p indicates geometry, color and pose emphasis weight, respectively. The weight < 1, > 1 and $= 1$ indicates suppressing, strengthening and neutralizing the emphasis, respectively. As the three weights are mutually independent, we can emphasize any subset of the three components (either one, two or all three).

The 3D object pose encoding in Sec. 3.4 just encodes the pose vector within a reserved harmonic channel (l_p, m_p), resulting in zero encoding in all the non-reserved channels. To ensure each of the constructed 3D Zernike basis in Sec. 3.3 carries both 3D object geometry, color and pose encoding, we first encode the object pose across all curated harmonic channel. As a result, each 3D Zernike basis Z^m_{nl} encodes 3D object geometry coefficient g^m_{nl}, color c^m_{nl} and pose p^m_{nl} independently, where $g^m_{nl} \in \mathbb{C}$, $c^m_{nl} \in \mathbb{C}$ and $p^m_{nl} \in \mathbb{R}$. Moreover, we normalize color encoding $E_{\mathcal{C}}$ and pose encoding $E_{\mathcal{P}}$ to geometry encoding space for even energy distribution: $E_{\mathcal{C}} = \frac{\|E_{\mathcal{G}}\|_2}{\|E_{\mathcal{C}}\|_2} \cdot E_{\mathcal{C}}$, $E_{\mathcal{P}} = \frac{\|E_{\mathcal{G}}\|_2}{\|E_{\mathcal{P}}\|_2} \cdot E_{\mathcal{P}}$.

We first merge $E_{\mathcal{G}}$ and $E_{\mathcal{C}}$ to obtain appearance encoding $E_{\mathcal{U}}$ with w_g and w_c,

$$u^m_{nl} = \begin{cases} |g^m_{nl}|^{1-t}|c^m_{nl}|^t \exp(\mathrm{i}[(1-t)\phi_g + t\phi_c]), & t = \frac{w_c}{w_c+w_g}, t \in [0,1], \quad \text{if } w_g + w_c \neq 0 \\ 0 + 0 \cdot \mathrm{i}, & \text{if } w_g = w_c = 0 \end{cases} \tag{11}$$

where ϕ_g and ϕ_c indicate the phase of g^m_{nl} in Eqn. (6) and phase of c^m_{nl} in Eqn. (7), respectively. Equation (11) degenerates into c^m_{nl} when $t = 1$, and g^m_{nl} when $t = 0$. Based on Eqn. (11), we further get the final geometry-color-pose encoding,

$$o^m_{nl} = \begin{cases} \left(1 + \frac{d}{2}\right) \rho_d(u^m_{nl}) \, \exp\big(\mathrm{i}\, \vartheta_d(u^m_{nl})\big) \, \exp\big(-\mathrm{i}\, p^m_{nl}\big) \; - \; \frac{d}{2}\, p^m_{nl}, & d < 0, \\ u^m_{nl} \, \exp\big(-\mathrm{i}\, e^{-5d}\, p^m_{nl}\big), & d \geqslant 0. \end{cases} \tag{12}$$

The detailed definition of $\rho_d(u^m_{nl})$ and $\vartheta_d(u^m_{nl})$ is deferred to Eqn. (45) and Eqn. (46) respectively in Sec. A.10 of Appendix. By applying Eqn. (12) across all coefficients in $E_{\mathcal{G}}$, $E_{\mathcal{C}}$ and $E_{\mathcal{P}}$, we obtain the joint encoding $E_{\mathcal{O}}$,

$$E_{\mathcal{O}}|_{(w_g,w_c,w_p)} = [o^0_{0,0}, o^0_{2,0}, \ldots, o^m_{n,l}, \ldots, o^M_{N,L}], \tag{13}$$

where $o^m_{n,l} \in \mathbb{C}$. We visualize how the three weights jointly control the three components in Fig. 3, from which we can observe the three weights steer the joint encoding smoothly and predictably across the full geometry-color-pose spectrum. Moving toward a vertex makes one component dominates, moving along an edge trades off two components. The centroid observes the mixture of the colors, which equivalently indicating neutral emphasis. The smooth, streak-free gradients demonstrates that the transition between components is continuous rather than abrupt. In summary, the joint encoding in Eqn. (13) is task-adjustable – a downstream task that cares mostly about

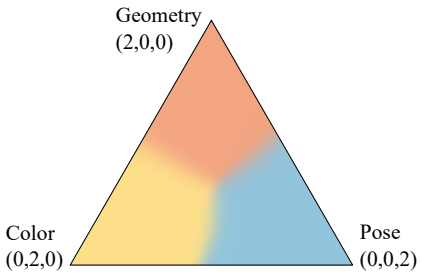


Figure 3: Geometry-Color-Pose emphasis controllability visualization. The three vertices with different color correspond to the pure one-emphasis settings. Interior color represents the blends of the three vertices colors, visually showing which component dominates the encoding at that emphasis.

geometry, color or pose can simply bias the weight toward the corresponding vertex, while the neutral point provides an unbiased, all-inclusive descriptor by default. More detailed discussion is provided in Sec. A.10 of Appendix.

3.6 X3DENC ENCODING RECAPITULATION

Motivated by XShapeEnc (He, 2026), we apply frequency propagation to the initial 3D object geometry, color encoding in Sec. §3.3 and the joint encoding in Sec. §3.5 to enrich the encoding frequency (mode discussion is provided in Sec. §A.11 of Appendix). Moreover, as we densely construct 3D Zernike basis from the lowest mode $Z^0_{0,0}$, a pre-defined encoding length inevitably loses the finer geometric or color detail because it truncates at the 3D Zernike basis reaching to the encoding length. To mitigate the detail-discarding issue, we propose to construct a much larger number of 3D Zernike basis than encoding length required to guarantee enough finer details are encoded, then adopt a data-independent and distance-preserving projection matrix $\mathbf{A}$ (Krahmer & Ward, 2011) to get the final real-valued encoding.

Algorithm 1: *X3DEnc* Encoding Workflow

Input : 3D object Ω; 3D Zernike basis $\{Z^m_{nl}\}$
Preprocess : decompose $\Omega \rightarrow \{\Omega_{\mathcal{G}}, \Omega_P, \Omega_{\mathcal{C}}\}$
if *encode 3D object geometry or color* **then**
  decide representation for $\Omega_{\mathcal{G}}$ or $\Omega_{\mathcal{C}}$ in Eqn. (5);
  get $E_{\mathcal{G}}$ from Eqn. (6) or get $E_{\mathcal{C}}$ from Eqn. (7);
  $E_{\text{raw}} \leftarrow E_{\mathcal{G}}$ or $E_{\mathcal{C}}$;
if *encode 3D object pose* **then**
  get 3D harmonic pose field $f_{\mathcal{P}}$ in Eqn. (8);
  project $f_{\mathcal{P}}$ to 3D Zernike basis in Eqn. (9);
  $E_{\text{raw}} \leftarrow E_{\mathcal{P}}$;
if *joint geometry-color-pose encode* **then**
  get emphasis weight: w_g, w_c, w_p in Eqn. (10);
  get joint encoding o^m_{nl} from Eqn. (12);
  $E_{\text{raw}} \leftarrow E_{\mathcal{O}}$;
get E_{out} from E_{raw} using Eqn. (14);
Output : Real-valued encoding E_{out}

$$E_{\text{out}} \leftarrow \mathbf{A} \cdot \text{flatten}(\text{FreqProp}(E_{\text{raw}})) \tag{14}$$

where $E_{\text{raw}} \in \{E_{\mathcal{G}}, E_{\mathcal{C}}, E_{\mathcal{O}}\}$ denotes the encoding selected by the requested mode, flatten denotes complex-to-real flattening (Eqn. (39) of Appendix), and FreqProp denotes frequency propagation (Eqn. (48) of Appendix). Further discussion of feature projection is provided in Sec. §A.9 of Appendix. The complete *X3DEnc* encoding workflow is summarized in Algorithm 1. Computationally, *X3DEnc* consists of deterministic field projection, linear-time frequency propagation over the retained coefficients. For a fixed spectral bandwidth, coefficient computation scales linearly with the number of evaluated Zernike modes. The Zernike bases can be precomputed and reused across objects. Encoding efficiency can be further improved by reducing the number of surface triangles through mesh decimation, as illustrated in Fig. 2 and quantified in Table 3.

4 EXPERIMENT

We evaluate *X3DEnc* from 3 main perspectives: 1. *X3DEnc* theoretical foundation and guarantee; 2. *X3DEnc* controllability on emphasis-adjustable encoding; 3. *X3DEnc* application in various tasks.

4.1 COMPARING BASELINES

We compare *X3DEnc* with 6 training-free baselines and 7 neural encoders. The 6 training-free baselines are: 1. plain 3D Zernike (plainZernike (Novotni & Klein, 2003)); 2. Spherical Harmonic Descriptor (SPH (Kazhdan et al., 2003)); 3. Shape Distributions (D2 (Osada et al., 2002)); 4. Light Field Descriptor (LFD (Chen et al., 2003)); 5. Ensemble of Shape Functions (ESF (Wohlkinger & Vincze, 2011)); 6. Shape Histograms (SH (Ankerst et al., 1999)). The 7 neural encoders are: 1. PointNet (Qi et al., 2017a); 2. PointNet++ (Qi et al., 2017b); 3. DGCNN (Wang et al., 2019); 4. Point-MAE (Pang et al., 2022); 5. Uni3D-B (Zhou et al., 2024); 6. Sonata (Wu et al., 2025); 7. Mamba3D-lite (Han et al., 2024). More information is provided in Sec. §A.13 of Appendix.

4.2 THEORETICAL FOUNDATION AND GUARANTEES

X3DEnc encoding is grounded in the orthonormality of the 3D Zernike bases in Eqn. (3): projecting the geometry, color, and harmonic pose fields through Eqn. (4) yields frequency-indexed coefficients

with linearity and rotation equivariance, while the norm of each (n, l) coefficient block is rotation invariant (Secs. A.4-A.6 of Appendix). The encoding before feature projection in Eqn. (38) in invertible in principle. The feature projection in Sec. §3.6 is data-independent and distance-preserving with mathematical rigor (see Sec. §A.9).

4.3 X3DENC GEOMETRY-COLOR-POSE EMPHASIS JOINT ENCODING CONTROLLABILITY

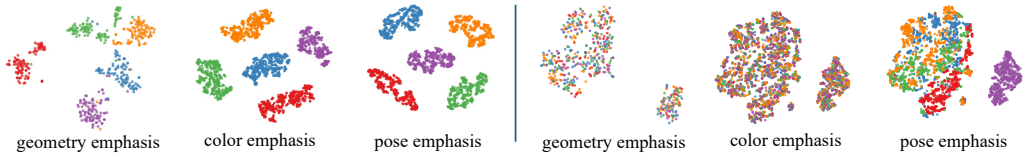


Figure 4: 3D object encoding relative-emphasis t-SNE clustering comparison between *X3DEnc* and Sonata (Wu et al., 2025). For clarity, we visualize only five object classes: airplane, vase, tv_stand, monitor, and chair. In each panel, different plot colors denote the ground-truth labels associated with the emphasized attribute. The same attribute-to-color mapping is used for *X3DEnc* and Sonata within each emphasis setting.

To evaluate *X3DEnc* controllable emphasis encoding across 3D shape geometry, color and pose, we construct a controlled variant of ModelNet40 (Wu et al., 2015) dataset, in which each object is assigned one of 5 colors (Indigo, Cyan, Teal, Sand and Rose), at one of 5 anchor positions (four corners and the center of a synthetic 100×100 m plane). The combined point cloud consists of the object's own points and uniformly-sampled context points spanning the whole plane. We evaluate all baselines on object-plus-context point cloud directly and ignore color if they do not support color encoding. *X3DEnc* encodes a 3D object's own geometry, color and spatial pose via the joint geometry–color–pose encoder. Retrieval performance (mean average precision, mAP) is then measured separately for 3D object geometry (aka class), color, and position relevance.

Table 1: Retrieval mAP (↑) under geometry, color and pose emphasis, respectively. - means the score is absent as the baseline does not support color encoding.

Method	Geometry (2,0,0)	Color (0,2,0)	Pose (0,0,2)
SPH (2003)	0.034	-	0.400
D2 (2002)	0.034	-	0.397
ESF (2011)	0.089	-	0.399
SH (1999)	0.035	-	0.364
LFD (2003)	0.034	-	0.754
PlainZernike (2003)	0.038	-	0.921
PointNet++ (2017b)	0.034	-	0.900
DGCNN (2019)	0.034	-	0.911
Point-MAE (2022)	0.034	-	0.858
Uni3D-B (2024)	0.034	0.650	0.552
Sonata (2025)	0.035	0.201	0.505
X3DEnc (voxel)	0.300	**0.968**	**1.000**
X3DEnc (tetrahedra)	0.323	0.957	**1.000**
X3DEnc (surface)	**0.324**	0.956	**1.000**

The quantitative evaluation result is in Table 1 and t-SNE (van der Maaten & Hinton, 2008) clustering visualization is shown Fig. 4. From Table 1 we can observe that the object-plus-plane point cloud leads to all baselines' geometric discriminability collapses toward chance (mAP on geometry emphasis 0.034-0.089). The color and pose information are relatively easier to measure than geometry. As *X3DEnc* encodes a 3D object geometry, color and pose via emphasis-adjustable encoding strategy (see Sec. §3.5), it is the only method that reaches the best on all three relevance axes. The t-SNE clustering in Fig. 4 further shows the superiority of *X3DEnc* – a clean and well-separated clusters along whichever axis is emphasized. All the experimental results demonstrate *X3DEnc*'s controllable disentanglement of geometry, color and pose from a unified encoding that none of the baselines compared here can replicate. More results are in Sec. §A.14 and Table 4 of Appendix.

4.4 APPLICATION IN VARIOUS DOWNSTREAM TASKS

Inter-Object Topological Relation Classification. To explicitly test *X3DEnc* inter-object spatial reasoning capability, we further propose an inter-object spatial topological relation classification task, in which we focus on 5 spatial topological relations arising from two spatially grounded 3D objects: *Touch*, *Overlap*, *Within*, *Equal* and *Disjoint*. Leveraging ScanObjectNN (Uy et al., 2019)



Figure 5: Inter-Object Topological Relation Illustration. For the sake of clear topological relation visualization, we use two contrastive colors to visualize two 3D objects, and ignore their RGB color.

real 3D objects dataset, we synthesize 3,000 training and 1,000 testing data from the training and testing split in ScanObjectNN data respectively. In each data synthesis, we randomly choose two 3D objects and spatially place them to reflect one particular topological relation (see Fig. 5). The task is to classify the topological relation. For each of the training-free and neural encoder baselines, we use it to encode feature from the combined two 3D objects data, while in *X3DEnc* we encode the two 3D objects separately and combine the two encoded features afterwards.

The classification accuracy is shown in Table 2. *X3DEnc* achieves the highest linear-probe accuracy of 0.84, outperforming the strongest neural baseline, and reaches 0.86 with an MLP probe, only 0.02 below DGCNN's best result. More importantly, the small improvement from the linear to the nonlinear probe indicates that the topological relations are already exposed in a directly decodable form rather than being recovered primarily through classifier capacity. This advantage follows *X3DEnc* design: encoding the two objects separately preserves each object's intrinsic geometry and explicit spatial pose before their representations are combined, avoiding the loss of object identity and relative-placement information that can occur when both objects are merged into a single input. The advantage of *X3DEnc* is not merely competitive classification accuracy, but a training-free and compositional representation in which inter-object spatial structure remains explicit, reusable, and readily accessible for relational reasoning.

Table 2: Inter-Object Topological Relation Classification Overall Accuracy. The MLP Probe has three hidden layers $[512, 256, 128]$.

Method	Linear Probe	MLP Probe
SPH (2003)	0.81	0.82
D2 (2002)	0.70	0.76
ESF (2011)	0.79	0.83
SH (1999)	0.59	0.60
LFD (2003)	0.80	0.81
PlainZernike (2003)	0.72	0.77
PointNet (2017a)	0.65	0.68
PointNet++ (2017b)	0.63	0.64
DGCNN (2019)	0.83	0.88
Mamba3D-lite (2024)	0.80	0.80
X3DEnc (surface)	**0.84**	**0.86**

The implementation is presented in Sec. §A.12 of Appendix. The experiments on *X3DEnc* cross-domain generalization capability is in Sec. §A.15, 3D object classification capability is in Sec. §A.16 of Appendix. We further provide source code in the supplementary material.

5 Conclusion and Limitations

We presented *X3DEnc*, a training-free framework for encoding spatially grounded 3D objects within a unified framework. It decouples 3D object's geometry, color and spatial pose and supports encoding them either separately or jointly, with emphasis-adjustable encoding strategy. We demonstrate the advantage of *X3DEnc* through extensive experiments and analysis. These results position *X3DEnc* as a complementary alternative to learned neural encoders, particularly when training data are limited or when deterministic, interpretable, and controllable multi-component encoding is required.

Limitations. *X3DEnc* assumes the 3D object data to be encoded exclusively represents the 3D object. The involvement of missing parts or background noise inevitably impact the encoding quality. Moreover, the adjustable emphasis weights act as steering parameters rather than calibrated measures of downstream importance, sometimes it requires task-specific tuning. Future work should therefore evaluate *X3DEnc* on large-scale real-world scenes with noisy, incomplete, or occluded observations.

AI USE STATEMENT

We used generative AI tools for the following tasks:

- **Paper polishing and writing assistance:** AI tools were used to assist with language polishing and drafting.
- **Implementation code review:** A code agent was used to assist with code review and bug fixing.

We have reviewed and validated all AI-assisted work and take responsibility for the final content of this paper, including its text, claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

X3DEnc is built entirely on open-source dataset, model. In the supplementary material, we provide the core code. We will release the code and configurations to reproduce results.

REFERENCES

Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. In *Advances in Spatial Databases: 6th International Symposium, SSD '99*, Lecture Notes in Computer Science, 1999. 7, 8, 9, 21

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, 2015. URL https://arxiv.org/abs/1512.03012. 22

Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On Visual Similarity Based 3D Model Retrieval. *Computer Graphics Forum*, 2003. 3, 7, 8, 9, 20

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2

Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3D: Enhancing Local Features for 3D Point Cloud Analysis via State Space Model. In *Proceedings of the 32nd ACM International Conference on Multimedia (ACM MM)*, 2024. 7, 9, 21, 22

Xu Han, Yuan Tang, Jinfeng Xu, and Xianzhi Li. MoST: Efficient Monarch Sparse Tuning for 3D Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 22

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: A Network with an Edge. *ACM Transactions on Graphics (ToG)*, 2019. 1, 2

Yuhang He. Training-free Spatially Grounded Geometric Shape Encoding (Technical Report). *arXiv preprint arXiv:2604.07522*, 2026. 5, 7

Jérôme Houdayer and Patrice Koehl. Stable Evaluation of 3D Zernike Moments for Surface Meshes. *Algorithms*, 2022. 5

Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In *Eurographics Symposium on Geometry Processing*. The Eurographics Association, 2003. 3, 7, 8, 9, 20

Felix Krahmer and Rachel Ward. New and Improved Johnson–Lindenstrauss Embeddings via the Restricted Isometry Property. *SIAM Journal on Mathematical Analysis*, 2011. 7, 17, 18

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. OpenShape: Scaling Up 3D Shape Representation Towards Open-World Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1

Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. 1, 2

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv preprint arXiv:2212.08751*, 2022. 1

Marcin Novotni and Reinhard Klein. 3D Zernike Descriptors for Content based Shape Retrieval. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, 2003. 1, 2, 3, 7, 8, 9, 20

Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape Distributions. *ACM Transactions on Graphics (ToG)*, 2002. 3, 7, 8, 9, 20

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked Autoencoders for Point Cloud Self-Supervised Learning. In *European Conference on Computer Vision (ECCV)*. Springer, 2022. 1, 7, 8, 21

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a. 1, 2, 3, 7, 9, 21, 22

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems (NIPS)*, 2017b. 3, 7, 8, 9, 21

Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010. 3

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 1

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 8, 21

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 2008. 8

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (ToG)*, 2019. 3, 7, 8, 9, 21, 22

Walter Wohlkinger and Markus Vincze. Ensemble of Shape Functions for 3D Object Classification. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011. 3, 7, 8, 9, 21

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-Supervised Learning of Reliable Point Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 7, 8, 21

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 8, 21, 22

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-Training 3D Point Cloud Transformers With Masked Point Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 3

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3D: Exploring Unified 3D Representation at Scale. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 7, 8, 21

A Appendix

A.1 3D Zernike Basis Definition

$$R_{nl}(r) = \sqrt{2n+3} \sum_{v=0}^{\frac{n-l}{2}} (-1)^{\frac{n-l}{2}+v} \frac{\binom{n-l}{k}\binom{\frac{n-l}{2}}{v}\binom{n+l+2v}{n-l}}{4^{\frac{n-l}{2}}\binom{\frac{n+l}{2}+v}{\frac{n-l}{2}}},$$
$$Y_l^m(\theta,\phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\phi)\, e^{im\theta}, \tag{15}$$

where $\sqrt{2n+3}$ is the radial normalization factor, $P_{(n-l)/2}^{(0,l+1/2)}(\cdot)$ is a Jacobi polynomial, and $P_l^m(\cdot)$ is the associated Legendre polynomial; negative orders follow the standard convention $Y_l^{-m} = (-1)^m \overline{Y_l^m}$.

3D Zernike basis exhibits several favorable properties as shown below:

3D Zernike Basis Based Encoding Properties

1. **Linearity**: for object fields V_1, V_2 and scalars α, β, $E(\alpha V_1 + \beta V_2) = \alpha E(V_1) + \beta E(V_2)$. Detailed proof is in Sec. A.4 of Appendix.
2. **Rotation Equivariance**: A 3D rotation preserves each (n, l) coefficient block and mixes only its m-indexed coefficients through the degree-l Wigner rotation matrix. A detailed proof is given in Sec. A.5 of Appendix.
3. **Rotation Invariance**: the block norm $s_{nl} = \left(\sum_{m=-l}^{l} |e_{n,l}^m|^2\right)^{1/2}$ is invariant to 3D rotations. Concatenating s_{nl} over all (n, l) therefore yields a rotation-invariant descriptor; the individual magnitudes $|e_{n,l}^m|$ are not generally invariant. Detailed proof is given in Sec. A.6 of Appendix.
4. **Invertibility**: The original object field can be reconstructed up to a radial order: $\widehat{\mathcal{S}_{3D\,N}} = \sum_{l,n,m} e_{n,l}^m \cdot Z_{n,l}^m$.

A.2 3D Zernike Basis Visualization Explanation

In Fig. 1, we visualized the 3D Zernike basis Z_{nl}^m with $n = 12$, $l = 4$ and $m = 2$. $n = 12$ indicates the radial rings, it counts the concentric oscillations from center to rim. That number is equals to $k = \frac{n-l}{2}$, so we can see 4 oscillations in sub-figure B; $m = 2$ indicates azimuthal lobes, the real part carries $\cos(m\phi)$, so you can see $2 \cdot m$ alternating-color wedges and m-fold rotational symmetry in sub-figure C; YZ-slice and YZ-slice are equatorial mirror symmetry, so we can see in sub-figure D $2 \cdot m$ alternating-color wedges but with symmetric pattern to sub-figure C.

A.3 3D Zernike Basis Orthogonality Proof

Equation (3) in main text can be rewritten as,

$$\int_0^1 \int_0^{2\pi} \int_0^{\pi} R_{nl}(r) Y_l^m(\theta,\phi) \overline{R_{n'l'}(r) Y_{l'}^{m'}(\theta,\phi)}\, r^2 \sin\phi\, d\phi\, d\theta\, dr \tag{16}$$

Because $R_{nl}(r)$ depends only on r and $Y_l^m(\theta,\phi)$ depends only on the angular variables, the integral factors in Eqn. (16) can be rewritten as,

$$\left(\int_0^1 R_{nl}(r)\overline{R_{n'l'}(r)} r^2\, dr\right)\left(\int_0^{2\pi}\int_0^{\pi} Y_l^m(\theta,\phi)\overline{Y_{l'}^{m'}(\theta,\phi)} \sin\phi\, d\phi\, d\theta\right) \tag{17}$$

The angular factor is the standard spherical-harmonic inner product on the unit sphere. Using the definition of Y_l^m in Eqn. (15), this inner product satisfies,

$$\int_0^{2\pi}\int_0^{\pi} Y_l^m(\theta,\phi)\overline{Y_{l'}^{m'}(\theta,\phi)}\sin\phi\,d\phi\,d\theta = \delta_{ll'}\delta_{mm'} \tag{18}$$

which follows from the Fourier orthogonality of $e^{im\theta}$ in θ and the associated-Legendre orthogonality in $\cos\phi$. Therefore Eqn. (18) equals to 0 whenever $l \neq l'$ or $m \neq m'$.

It remains to consider the case where $l = l'$ and $m = m'$. For a fixed admissible l, the normalized radial Zernike polynomials in Eqn. (15) are orthonormal on $[0,1]$ under the radial volume weight r^2:

$$\int_0^1 R_{nl}(r)\overline{R_{n'l}(r)}r^2\,dr = \delta_{nn'}. \tag{19}$$

Combining the radial and angular factors gives the Eqn. (3), which proves the 3D Zernike basis orthogonality condition in Eqn. (3).

A.4 Linearity Proof

Let $V_1, V_2 \in L^2(\mathbb{B};\mathbb{C})$ be two 3D object fields defined on the unit ball, and let $\alpha,\beta \in \mathbb{C}$. Following Eqn. (4), the coefficient associated with the basis $Z_{n,l}^m$ is the inner product

$$e_{n,l}^m[V] = \langle V, Z_{n,l}^m\rangle = \int_0^1\int_0^{2\pi}\int_0^{\pi} V(r,\theta,\phi)\,\overline{Z_{n,l}^m(r,\theta,\phi)}\,r^2\sin\phi\,d\phi\,d\theta\,dr. \tag{20}$$

For the linear combination $V = \alpha V_1 + \beta V_2$, substituting it into Eqn. (20) gives

$$\begin{aligned} e_{n,l}^m[\alpha V_1+\beta V_2] &= \int_0^1\int_0^{2\pi}\int_0^{\pi} \big(\alpha V_1(r,\theta,\phi)+\beta V_2(r,\theta,\phi)\big)\overline{Z_{n,l}^m(r,\theta,\phi)}\,r^2\sin\phi\,d\phi\,d\theta\,dr \\ &= \alpha\int_0^1\int_0^{2\pi}\int_0^{\pi} V_1(r,\theta,\phi)\overline{Z_{n,l}^m(r,\theta,\phi)}\,r^2\sin\phi\,d\phi\,d\theta\,dr \\ &\quad + \beta\int_0^1\int_0^{2\pi}\int_0^{\pi} V_2(r,\theta,\phi)\overline{Z_{n,l}^m(r,\theta,\phi)}\,r^2\sin\phi\,d\phi\,d\theta\,dr \\ &= \alpha e_{n,l}^m[V_1] + \beta e_{n,l}^m[V_2]. \end{aligned} \tag{21}$$

Because Eqn. (21) holds independently for every admissible index (n,l,m), stacking all coefficients into the encoding vector yields

$$E(\alpha V_1+\beta V_2) = \alpha E(V_1) + \beta E(V_2), \tag{22}$$

which proves that the 3D Zernike projection encoding is a linear operator. This property holds for either the complete basis or any fixed finite truncation of the basis set.

A.5 Rotation Equivariance Proof

Let $Q \in \mathrm{SO}(3)$ and define the rotated object field by

$$V_Q(\mathbf{x}) = V(Q^{-1}\mathbf{x}). \tag{23}$$

For fixed (n,l), collect the coefficients and basis functions over all orders m into

$$\mathbf{e}_{nl}[V] = [e_{n,l}^{-l}[V],\ldots,e_{n,l}^{l}[V]]^\top, \qquad \mathbf{Z}_{nl}(\mathbf{x}) = [Z_{n,l}^{-l}(\mathbf{x}),\ldots,Z_{n,l}^{l}(\mathbf{x})]^\top. \tag{24}$$

Because the radial coordinate is unchanged by rotation and the degree-l spherical harmonics form a rotation-invariant subspace, there exists a unitary degree-l Wigner matrix $D^{(l)}(Q)$ such that, under the adopted spherical-harmonic convention,

$$\overline{\mathbf{Z}_{nl}(Q\mathbf{x})} = D^{(l)}(Q)\overline{\mathbf{Z}_{nl}(\mathbf{x})}. \tag{25}$$

Using Eqn. (20) and changing variables from $\mathbf{x}$ to $\mathbf{y} = Q^{-1}\mathbf{x}$, we obtain

$$\begin{aligned}
\mathbf{e}_{nl}[V_Q] &= \int_{\mathbb{B}} V(Q^{-1}\mathbf{x})\overline{\mathbf{Z}_{nl}(\mathbf{x})}\, d\mathbf{x} \\
&= \int_{\mathbb{B}} V(\mathbf{y})\overline{\mathbf{Z}_{nl}(Q\mathbf{y})}\, d\mathbf{y} \\
&= D^{(l)}(Q) \int_{\mathbb{B}} V(\mathbf{y})\overline{\mathbf{Z}_{nl}(\mathbf{y})}\, d\mathbf{y} \\
&= D^{(l)}(Q)\mathbf{e}_{nl}[V].
\end{aligned} \tag{26}$$

Here the change of variables preserves both the unit ball and the volume measure because Q is orthogonal and $|\det Q| = 1$. Equation (26) shows that a rotation does not mix different radial orders n or angular degrees l; it mixes only the $2l + 1$ coefficients indexed by m within each fixed (n, l) block. Stacking these blocks yields a block-diagonal transformation of the full encoding, proving rotation equivariance.

A.6 Rotation Invariance Proof

Rotation equivariance alone does not make each coefficient $e^m_{n,l}$ invariant, because a rotation mixes the m-indexed coefficients within each fixed (n, l) block. We instead define the block magnitude

$$s_{nl}[V] = \|\mathbf{e}_{nl}[V]\|_2 = \left(\sum_{m=-l}^{l} \left| e^m_{n,l}[V] \right|^2 \right)^{1/2}. \tag{27}$$

From the equivariance relation in Eqn. (26), the coefficients of the rotated field satisfy

$$\mathbf{e}_{nl}[V_Q] = D^{(l)}(Q)\mathbf{e}_{nl}[V]. \tag{28}$$

Because the Wigner matrix $D^{(l)}(Q)$ is unitary, $D^{(l)}(Q)^{\mathrm{H}} D^{(l)}(Q) = I_{2l+1}$. Therefore,

$$\begin{aligned}
s_{nl}[V_Q]^2 &= \|\mathbf{e}_{nl}[V_Q]\|_2^2 \\
&= \left(D^{(l)}(Q)\mathbf{e}_{nl}[V] \right)^{\mathrm{H}} \left(D^{(l)}(Q)\mathbf{e}_{nl}[V] \right) \\
&= \mathbf{e}_{nl}[V]^{\mathrm{H}} D^{(l)}(Q)^{\mathrm{H}} D^{(l)}(Q)\mathbf{e}_{nl}[V] \\
&= \mathbf{e}_{nl}[V]^{\mathrm{H}}\mathbf{e}_{nl}[V] \\
&= s_{nl}[V]^2,
\end{aligned} \tag{29}$$

and hence $s_{nl}[V_Q] = s_{nl}[V]$. Concatenating these block magnitudes over all admissible (n, l) gives the descriptor

$$S(V) = [s_{00}[V], s_{20}[V], \ldots, s_{nl}[V], \ldots], \qquad S(V_Q) = S(V), \tag{30}$$

which proves rotation invariance. This construction applies to both the complete basis and any fixed finite truncation. Unlike the original equivariant coefficient vector $E(V)$, the invariant descriptor $S(V)$ discards orientation information because it retains only the norm of each (n, l) block.

A.7 3D Object Pose Encoding Invertibility Proof

Let $\mathbf{p} = [p_1, \ldots, p_K]^\top \in \mathbb{R}^K$ and consider the harmonic pose field from Eqn. (8),

$$f_{\mathcal{P}}(r, \theta, \phi; \mathbf{p}) = \left(\sum_{k=1}^{K} p_k w_k(r) \right) Y_{l_p}^0(\theta, \phi). \tag{31}$$

For the admissible radial orders $n_j = l_p + 2(j-1)$, $j = 1, \ldots, L$, define the projection coefficient

$$a_j = \left\langle f_{\mathcal{P}}, Z_{n_j l_p}^0 \right\rangle_{L^2(\mathbb{B})} \tag{32}$$

Substituting $Z_{n_j l_p}^0(r, \theta, \phi) = R_{n_j l_p}(r) Y_{l_p}^0(\theta, \phi)$ into Eqn. (32) gives

$$\begin{aligned} a_j &= \int_0^1 \int_0^{2\pi} \int_0^{\pi} \left(\sum_{k=1}^{K} p_k w_k(r) \right) Y_{l_p}^0(\theta, \phi) \overline{R_{n_j l_p}(r) Y_{l_p}^0(\theta, \phi)} r^2 \sin\phi \, d\phi \, d\theta \, dr \\ &= \sum_{k=1}^{K} p_k \left(\int_0^1 w_k(r) R_{n_j l_p}(r) r^2 \, dr \right) \left(\int_0^{2\pi} \int_0^{\pi} \left| Y_{l_p}^0(\theta, \phi) \right|^2 \sin\phi \, d\phi \, d\theta \right) \\ &= \sum_{k=1}^{K} C_{jk} p_k, \end{aligned} \tag{33}$$

where spherical-harmonic normalization makes the angular integral equal to one, and

$$C_{jk} = \int_0^1 w_k(r) R_{n_j l_p}(r) r^2 \, dr. \tag{34}$$

Stacking all L coefficients into $\mathbf{a} = [a_1, \ldots, a_L]^\top$ yields

$$\mathbf{a} = C\mathbf{p}, \qquad C \in \mathbb{R}^{L \times K}. \tag{35}$$

Assume $L \geqslant K$ and $\operatorname{rank}(C) = K$. Then $C^\top C$ is positive definite and invertible. Multiplying Eqn. (35) by the left pseudoinverse $C^\dagger = (C^\top C)^{-1} C^\top$ gives

$$\hat{\mathbf{p}} = C^\dagger \mathbf{a} = (C^\top C)^{-1} C^\top C \mathbf{p} = \mathbf{p}. \tag{36}$$

Equivalently, if two pose vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ produce the same encoding, then

$$C\mathbf{p}_1 = C\mathbf{p}_2 \quad \Longrightarrow \quad C(\mathbf{p}_1 - \mathbf{p}_2) = 0. \tag{37}$$

Since a full-column-rank matrix has a trivial null space, $\mathbf{p}_1 - \mathbf{p}_2 = 0$ and hence $\mathbf{p}_1 = \mathbf{p}_2$. Therefore, the pose encoding is injective and losslessly invertible in the noiseless setting.

The condition $K \leqslant L$ is necessary but not sufficient: the selected finite projection matrix must also have full column rank. Although Gram–Schmidt makes the radial windows mutually orthonormal in $L^2([0,1], r^2 dr)$, numerical implementations should still verify the rank or smallest singular value of the truncated matrix C. Once C is fixed, the harmonic field does not need to be explicitly sampled during encoding or decoding; the equivalent coefficient-domain operations $\mathbf{a} = C\mathbf{p}$ and $\mathbf{p} = C^\dagger \mathbf{a}$ are sufficient.

Table 3: Comparison between the three representations. r denotes the voxel resolution, $N_{\mathcal{Z}}$ the number of evaluated Zernike modes, $N_{\triangle}$ the number of surface triangles, and $q = \lfloor N_{\max}/2 \rfloor + 1$ the number of quadrature points per integration axis for maximum radial order $N_{\max}$.

Representation	Evaluation domain	Measure	Cost	Requirement
Voxel	voxel grid	volume	$\mathcal{O}(N_{\mathcal{Z}} r^3)$	point cloud or mesh
Signed Tetrahedra	tetrahedral cones	signed volume	$\mathcal{O}(N_{\mathcal{Z}} q^3 N_{\triangle})$	watertight closed surface
Surface Triangles	triangular faces	surface area	$\mathcal{O}(N_{\mathcal{Z}} q^2 N_{\triangle})$	closed or open surface

A.8 Three 3D Object Representations Comparison

Table 3 shows that no single representation is uniformly preferable; instead, each offers a distinct trade-off among input compatibility, geometric measure, and computational cost. The voxel formulation accepts either point clouds or meshes and provides direct volumetric encoding, but its cost grows cubically with the spatial resolution r, making fine discretizations expensive. Signed tetrahedra preserve a volume-based formulation without voxelization, yet require a watertight closed surface and incur three-dimensional quadrature cost proportional to $q^3 N_{\triangle}$. Surface triangles reduce the per-face quadrature dependence to q^2, scale linearly with the number of faces, and naturally support both closed and open surfaces, although they encode surface area rather than enclosed volume. Across all three choices, the cost remains linear in the number of evaluated Zernike modes $N_{\mathcal{Z}}$. Consequently, *X3DEnc* can select the representation according to the input topology and computational budget: voxels provide broad input compatibility, signed tetrahedra favor volume-aware encoding of watertight solids, and surface triangles provide the most flexible and typically more efficient option for mesh-based open or hollow objects.

A.9 Shared Structured Projection for Fixed-Length Encoding

The 3D object geometry, color encoding in Sec. §3.3 and joint encoding in Sec. §3.5 naturally prioritize coarse geometric or color information as we densely construct 3D Zernike basis starting from the lowest mode $Z^0_{0,0}$. A predefined encoding length inevitably loses the finer geometric or color detail because it truncates at the 3D Zernike basis that reaches to the length. For example, we just require 12 and 16 3D Zernike basis to get the encoding length of 512 and 1024, respectively, resulting in the nonexpressive encoding if the 3D geometric or color detail matters. To mitigate this issue, we explicitly disentangle the 3D Zernike basis number from the final encoding length. Specifically, we construct a much larger number of 3D Zernike basis than required to ensure enough finer details are encoded, followed by another data-independent and distance-preserving projection (Krahmer & Ward, 2011) to obtain the final fixed-length encoding.

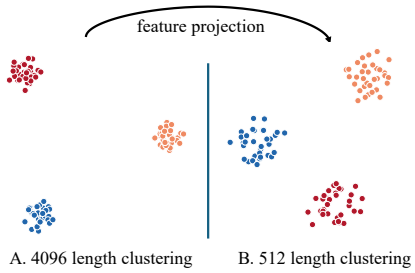


Figure 6: We visualize the t-SNE clustering for 100 features before and after projection. The 100 features are originally of 4096 size, and 512 after projection. We can observe that the three clustering pattern is maintained after projection, which attests the Walsh–Hadamard projection keeps the pairwise distance.

Let N_D and L indicate the 3D Zernike basis number and target encoding length, respectively, where $2 \cdot N_D \gg L$. We construct one data-independent matrix $\mathbf{A} \in \mathbb{R}^{L \times 2N_D}$ following Krahmer and Ward (Krahmer & Ward, 2011). For the complex-valued encoding $E_{\mathcal{G}}$ in Eqn. (6), $E_{\mathcal{C}}$ in Eqn. (7), $E_{\mathcal{O}}$ in Eqn. (12), we sequentially flatten each complex-valued coefficient's real and imaginary part in juxtaposition to form the real-valued encoding of size $2 \cdot N_D$. We then project the flattened $E_{\mathcal{G}}$, $E_{\mathcal{C}}$ and $E_{\mathcal{O}}$ to the target encoding length, respectively,

$$E_{\mathcal{G}} \leftarrow \mathbf{A} \cdot \text{flatten}(E_{\mathcal{G}}); \quad E_{\mathcal{C}} \leftarrow \mathbf{A} \cdot \text{flatten}(E_{\mathcal{C}}); \quad E_{\mathcal{O}} \leftarrow \mathbf{A} \cdot \text{flatten}(E_{\mathcal{O}}) \tag{38}$$

where $E_{\mathcal{G}}, E_{\mathcal{C}}, E_{\mathcal{O}} \in \mathbb{R}^D$. At a high level, the construction first applies independent random sign flips to the input coordinates and then uses a Walsh–Hadamard transform to distribute each original coefficient across the transformed vector. It subsequently retains L uniformly selected

transformed coordinates and applies an appropriate normalization, producing a structured projection from $2N_D$ dimensions to L dimensions. The random signs and selected coordinates are sampled once using a fixed seed, and the resulting $\mathbf{A}$ is shared across all objects and all encoding types rather than reconstructed for each individual encoding. This shared projection allows every retained Zernike mode to contribute to the compact representation while approximately preserving pairwise distances (see Fig. 6),

The postprocessing step introduced in Sec. §3.6 is motivated by a separation between *spectral resolution* and *descriptor length*. The former is controlled by the number N_D of retained 3D Zernike coefficients: increasing N_D extends the represented radial and angular frequencies and therefore permits finer shape or color variation to enter the encoding. The latter is the prescribed downstream dimension L, such as 512 or 1024. Direct truncation couples these two quantities and removes the highest-order coefficients first. Instead, *X3DEnc* computes a substantially longer spectral vector and subsequently maps all of its coordinates to a fixed-length descriptor using one shared, data-independent projection.

For object s and encoding type $\mathcal{X} \in \{\mathcal{G}, \mathcal{C}, \mathcal{O}\}$, let $E_{s,\mathcal{X}} = [e_{s,\mathcal{X},1}, \ldots, e_{s,\mathcal{X},N_D}] \in \mathbb{C}^{N_D}$ follow the same fixed ordering of (n, l, m) across all objects and encoding types. We juxtapose the real and imaginary parts coefficientwise,

$$\begin{aligned} \mathbf{x}_{s,\mathcal{X}} &= \mathrm{flatten}(E_{s,\mathcal{X}}) \\ &= \big[\Re(e_{s,\mathcal{X},1}), \Im(e_{s,\mathcal{X},1}), \ldots, \Re(e_{s,\mathcal{X},N_D}), \Im(e_{s,\mathcal{X},N_D})\big]^\top, \\ \mathbf{x}_{s,\mathcal{X}} &\in \mathbb{R}^M, \qquad M = 2N_D. \end{aligned} \tag{39}$$

A real-valued coefficient sequence can be handled by the same layout by setting every imaginary coordinate to zero. This convention keeps the coefficient positions aligned across geometry, color, and their joint encoding, which is necessary when the outputs are compared under different emphasis settings.

The Walsh–Hadamard transform requires a power-of-two input length. We therefore set $P = 2^{\lceil \log_2 M \rceil}$ and define the zero-padding injection $\Pi \in \mathbb{R}^{P \times M}$ by $\Pi\mathbf{x} = [\mathbf{x}^\top, \mathbf{0}^\top]^\top$. Zero padding introduces no additional information and preserves Euclidean distances. The projection is then constructed in three stages:

(i) **Random sign modulation.** Draw one Rademacher vector $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_P]^\top$ with independent $\xi_j \in \{-1, +1\}$ and form $\mathbf{D}_{\boldsymbol{\xi}} = \mathrm{diag}(\boldsymbol{\xi})$. This randomly changes the signs of the padded coordinates without changing their magnitudes or Euclidean norm.

(ii) **Orthogonal global mixing.** Apply the unnormalized Walsh–Hadamard matrix $\mathbf{H}_P \in \{-1, +1\}^{P \times P}$, which satisfies $\mathbf{H}_P^\top \mathbf{H}_P = P\mathbf{I}_P$. Each transformed coordinate consequently combines contributions from every packed Zernike coordinate, rather than corresponding to one selected low- or high-order mode.

(iii) **Uniform row selection.** Sample a subset $\Omega \subset \{1, \ldots, P\}$ of L row indices uniformly without replacement. Let $\mathbf{R}_\Omega \in \{0, 1\}^{L \times P}$ retain exactly those transformed coordinates.

Combining these operations gives the shared projection matrix,

$$\mathbf{A} = \frac{1}{\sqrt{L}} \mathbf{R}_\Omega \mathbf{H}_P \mathbf{D}_{\boldsymbol{\xi}} \Pi \in \mathbb{R}^{L \times M}, \qquad \widetilde{E}_{s,\mathcal{X}} = \mathbf{A}\mathbf{x}_{s,\mathcal{X}} \in \mathbb{R}^L. \tag{40}$$

The factor $1/\sqrt{L}$ compensates for retaining only L of the P Hadamard coordinates. Since the complete unnormalized transform has energy $P\|\mathbf{x}\|_2^2$ and every row is selected with probability L/P, the projected energy is unbiased:

$$\mathbb{E}_\Omega \left[\|\mathbf{A}\mathbf{x}\|_2^2 \right] = \|\mathbf{x}\|_2^2 . \tag{41}$$

The random sign modulation is central to the result of Krahmer and Ward (Krahmer & Ward, 2011): randomizing the column signs of a matrix satisfying an appropriate restricted isometry property

converts it into a Johnson–Lindenstrauss embedding for a fixed finite collection of vectors. In Eqn. (40), the partial Hadamard operator $\frac{1}{\sqrt{L}}\mathbf{R}_\Omega\mathbf{H}_P$ provides the structured parent matrix, while $\mathbf{D}_{\boldsymbol{\xi}}$ performs the required column-sign randomization. Under the corresponding restricted-isometry and output-dimension conditions, the resulting map satisfies, with high probability,

$$(1-\varepsilon)\left\|\mathbf{x}_{i,\mathcal{X}}-\mathbf{x}_{j,\mathcal{Y}}\right\|_2^2 \leqslant \left\|\widetilde{E}_{i,\mathcal{X}}-\widetilde{E}_{j,\mathcal{Y}}\right\|_2^2 \leqslant (1+\varepsilon)\left\|\mathbf{x}_{i,\mathcal{X}}-\mathbf{x}_{j,\mathcal{Y}}\right\|_2^2, \qquad \mathcal{X},\mathcal{Y}\in\{\mathcal{G},\mathcal{C},\mathcal{O}\}. \tag{42}$$

For each pair of dimensions (M, L), we sample $\boldsymbol{\xi}$ and Ω only once using a fixed random seed. The resulting $\mathbf{A}$ is then reused for every object, every dataset split, and each of $E_{\mathcal{G}}$, $E_{\mathcal{C}}$, and $E_{\mathcal{O}}$. It is not estimated from the coefficient values and is never reconstructed separately for an individual object. For example, if S objects produce the three encoding types, the $3S$ packed row vectors can be stacked into $\mathbf{X}\in\mathbb{R}^{3S\times M}$ and projected simultaneously as

$$\widetilde{\mathbf{X}} = \mathbf{X}\mathbf{A}^\top \in \mathbb{R}^{3S\times L}. \tag{43}$$

Sharing the matrix is essential: it places every result in the same projected coordinate system, makes cross-object distances meaningful, and preserves the linear relationships between geometry, color, and the emphasis-adjustable joint encoding. A different matrix is generated only when the packed input dimension M, target length L, or experimental random seed changes.

The projection does not synthesize detail that was absent from the original Zernike coefficients. Rather, it prevents a fixed output length from forcing the systematic removal of high-frequency modes: every retained mode participates in the globally mixed descriptor. Likewise, because $L < M$, the compact projection is not generally invertible and its individual coordinates no longer correspond to particular (n, l, m) modes. The uncompressed Zernike encoding is therefore retained whenever reconstruction or coefficient-level interpretation is required, while the projected encoding is used for fixed-length downstream comparison.

Finally, $\mathbf{A}$ need not be explicitly stored as a dense matrix. Given $\mathbf{x}$, we zero-pad it, apply the fixed sign vector, compute the fast Walsh–Hadamard transform in $\mathcal{O}(P\log P)$ operations, retain the entries indexed by Ω, and divide by $\sqrt{L}$. Thus, the projection remains training-free and efficient while allowing *X3DEnc* to choose its spectral bandwidth independently of the final descriptor length.

A.10 More Detail on Geometry-Color-Pose Joint Encoding

Let D denote the number of retained Zernike modes, and let $\mathbf{g}, \mathbf{c}\in\mathbb{C}^D$ denote the geometry and color encodings. We first balance them coefficient wise to obtain the appearance encoding $\mathbf{u} = [u_{nl}^m]\in\mathbb{C}^D$,

$$u_{nl}^m = \begin{cases} |g_{nl}^m|^{1-t}|c_{nl}^m|^t \exp(\mathrm{i}[(1-t)\phi_g + t\phi_c]), & t=\frac{w_c}{w_c+w_g}, t\in[0,1], \quad \text{if } w_g+w_c\neq 0 \\ 0+0\cdot\mathrm{i}, & \text{if } w_g=w_c=0 \end{cases} \tag{44}$$

where Eqn. (44) degenerates into c_{nl}^m when $t=1$, and g_{nl}^m when $t=0$. When $w_g=w_c=0$, every coefficient is the complex zero; hence $\mathbf{u}=\mathbf{0}_D+\mathrm{i}\cdot\mathbf{0}_D\in\mathbb{C}^D$, which has the same length as the original geometry encoding and has identically zero real and imaginary parts. The second control is the appearance ↔ pose balance, governed by the signed offset of the mean appearance gain from the pose gain:

$$d = \frac{1}{2}(w_g+w_c) - w_p, \quad d\in[-2,2] \tag{45}$$

where $d>0$ sharpens the appearance and weakens the pose effect, while $d<0$ does the opposite. For $d<0$, we interpolate the appearance phase from its original angle to the fixed angle 1 radian:

$$\vartheta_d(u_{nl}^m) = \begin{cases} (1+\frac{d}{2})\arg(u_{nl}^m) - \frac{d}{2}, & u_{nl}^m\neq 0, -2\leqslant d<0, \\ 1, & u_{nl}^m = 0, -2\leqslant d<0. \end{cases} \tag{46}$$

We simultaneously interpolate the appearance magnitude toward a unit carrier,

$$\rho_d(u_{nl}^m) = \left(1 + \frac{d}{2}\right)|u_{nl}^m| - \frac{d}{2}, \qquad -2 \leqslant d < 0. \tag{47}$$

A.11 Frequency Propagation

After projecting a shape's occupancy field onto the 3D Zernike basis, *X3DEnc* applies a *frequency propagation* (FreqProp) step that mixes energy between neighboring frequency bands before the coefficients are compacted into the final descriptor. Let $z_{nl}^m \in \mathbb{C}$ denote the raw Zernike moment of radial order n, angular degree l, and order m (with $n - l$ even and $|m| \leqslant l$). FreqProp propagates *from lower order to higher order* along two independent axes, each governed by a single decay parameter $\lambda \in (0, 1)$:

$$\begin{aligned} \text{Radial:} \quad & \tilde{z}_{nl}^m = z_{nl}^m + \lambda\, \tilde{z}_{n-2,\,l}^m, \\ \text{Angular:} \quad & \tilde{z}_{nl}^m = \tilde{z}_{nl}^m + \lambda\, \tilde{z}_{n,\,l-2}^m, \qquad l \geqslant |m| + 2 \end{aligned} \tag{48}$$

Radial propagation is applied first (for fixed (l, m), energy flows $n{-}2 \to n$), followed by angular propagation on the result (for fixed (n, m), energy flows $l{-}2 \to l$). Both passes traverse coefficients in ascending (n, l) order and use the *already-propagated* value $\tilde{z}$ as the source for the next hop, so the propagation is *cascaded*: a coefficient at order n implicitly aggregates a λ-weighted geometric contribution from all lower orders $n-2, n-4, \ldots$ (and similarly for l), rather than only its immediate neighbor. Coefficients with $m < 0$ are not propagated directly; they are recovered afterward via conjugate symmetry, $\tilde{z}_{nl}^{-m} = (-1)^m \overline{\tilde{z}_{nl}^m}$, which holds because the encoded field is real-valued. The resulting $\tilde{z}$, not the raw z, is what is compacted (real/imaginary interleaved, $m \geqslant 0$ only) into *X3DEnc* feature vector.

A.12 Implementation Detail

Emphasis-Conditioned Retrieval and Evaluation. The joint encoding is steered by three per-component emphasis gains $(w_g, w_c, w_p) \in [0, 2]^3$ (neutral value 1), and we evaluate the seven principal settings that emphasize geometry $(2, 0, 0)$, color $(0, 2, 0)$, position $(0, 0, 2)$, their pairwise combinations, and the neutral point $(1, 1, 1)$. For each emphasis setting we ℓ_2-normalize the descriptors and rank every gallery object per query by squared Euclidean distance. Relevance is defined under seven label criteria (same category; same color; same position; and their conjunctions), which exposes both the gain on the emphasized attribute and any leakage from the others. Retrieval quality is measured by *mean average precision* (mAP).

Inter-Object Topological Relation Classification Setting. All training-free descriptors (SPH, D2, ESF, SH, LFD, PlainZernike) are evaluated identically: features are normalized. then passed to two fixed probes. In the small MLP probe, we add three hidden linear layers $[512, 256, 218]$. Adam optimizer is used to train all classifiers with learning rate 10^{-3} and we train 10 epochs with 8 NVIDIA A100 GPUs. *X3DEnc* encoding dimension is 512, projected from 744-dim by 3D Zernike encoding. Two 3D objects' features are elementwise added before feeding to the probing layer. The training set has 3,000 instances, while the testing set has 1,000 instances.

A.13 More Discussion on Baselines

We compare *X3DEnc* with 6 training-free baselines and 7 neural network encoders. The 6 training-free baselines are: 1. **plain 3D Zernike** (plainZernike (Novotni & Klein, 2003)), which is the closest classical spectral baseline to *X3DEnc*; 2. **Spherical Harmonic Descriptor** (SPH (Kazhdan et al., 2003)). A centered, scale-normalized shape field is sampled on concentric spherical shells; 3. **Shape Distributions** (D2 (Osada et al., 2002)). D2 represents a shape by a normalized histogram of Euclidean distances between pairs of points sampled uniformly with respect to surface area. Pairwise distances are invariant to rigid transformations; scale invariance additionally requires distance normalization; 4. **Light Field Descriptor** (LFD (Chen et al., 2003)). LFD compares shapes through silhouettes rendered from multiple viewing directions. Each silhouette is encoded using 2D Zernike moments and Fourier contour descriptors, and shape similarity is obtained by matching view

sets over candidate camera-system alignments; 5. **Ensemble of Shape Functions** (ESF (Wohlkinger & Vincze, 2011)) provides a global point-cloud descriptor based on multiple geometric shape-function histograms; 6. **Shape Histograms** (SH (Ankerst et al., 1999)) partitions the space surrounding a 3D object's centroid into spatial bins and records the normalized distribution.

Neural Encoder Baselines. We select 7 representative neural encoders. 1. **PointNet** (Qi et al., 2017a) and 2. **PointNet++** (Qi et al., 2017b) are the early work learning from 3D point cloud; 3. **DGCNN** (Wang et al., 2019) repeatedly constructs k-nearest-neighbor graphs in feature space; 4. **Point-MAE** (Pang et al., 2022) learns transferable object representations by masking local point patches and reconstructing them; 5. **Uni3D-B** (Zhou et al., 2024) represents large-scale multimodal pretraining, aligning a point-cloud Transformer with pretrained image and language representations; 6. **Sonata** (Wu et al., 2025) uses self-distillation with a Point Transformer V3 backbone to learn transferable point representations; 7. **Mamba3D-lite** (Han et al., 2024) is a lightweight configuration of Mamba3D that combines Local Norm Pooling for neighborhood geometry with a bidirectional state-space module for global points. These baselines cover complementary supervised, self-supervised, graph-based, Transformer-based, and multimodal approaches to neural 3D object encoding.

A.14 More Experimental Result on X3DEnc Controllability

Unlike the comparison methods, which produce one fixed descriptor, *X3DEnc* exposes explicit weights (w_g, w_c, w_p) that steer the same joint representation toward the attribute required by a downstream task. Table 4 shows that the three pure-emphasis settings recover their intended relevance axes: geometry emphasis yields a geometry mAP of 0.324, color emphasis reaches 0.956, and position emphasis achieves a perfect 1.000. The geometry result is particularly informative: it is approximately $8.5\times$ the plain 3D Zernike result of 0.038, although both approaches rely on the same underlying Zernike geometry representation. The gain therefore comes from the *X3DEnc* architecture, which encodes undistorted canonical geometry and introduces position through a separate native channel, rather than forcing geometry and spatial context into one scene-level descriptor. Likewise, the perfect position result matches the strongest learned position-sensitive baselines while retaining a fixed, training-free encoder and an explicit user-controlled emphasis mechanism. The mixed settings reveal an additional nuance: geometry–color and neutral weighting remain strongly color-dominated, whereas position becomes discriminative only under dedicated position emphasis. Thus, the weights provide effective control over the representation, but their numerical values should be interpreted as steering parameters rather than as a linear prediction of downstream mAP; intermediate task-specific settings may require calibration.

Table 4: Retrieval mAP under different *X3DEnc* geometry, color, and position emphasis settings.

Emphasis weights	Geometry	Color	Position
Geometry $(2, 0, 0)$	**0.324**	0.201	0.201
Color $(0, 2, 0)$	0.093	**0.956**	0.201
Position $(0, 0, 2)$	0.034	0.201	**1.000**
Geometry–color $(1, 1, 0)$	0.073	0.900	0.201
Geometry–position $(1, 0, 1)$	0.173	0.201	0.201
Color–position $(0, 1, 1)$	0.081	0.271	0.201
Neutral $(1, 1, 1)$	0.083	0.949	0.201

A.15 Extra Experiment 1: Cross-Domain Generalization

One central premise of *X3DEnc* is that an encoding by *X3DEnc* can substitute for a pretrained neural encoders in settings where the labeled data is scarce. The most demanding test of this premise is *cross-domain* generalization: a 3D object encoding trained (or extracted by *X3DEnc*) on one distribution and evaluated on a completely different dataset, without any finetuning applied. We therefore compare *X3DEnc* against 3 neural encoders (PointNet (Qi et al., 2017a), DGCNN (Wang et al., 2019) and Mamba3D-lite (Han et al., 2024) variant) trained from scratch on ScanObjectNN (Uy et al., 2019) and then evaluated on ModelNet40 (Wu et al., 2015) without adaptation or finetuning – the `trainSONN→testMN40` cross-domain classification setting. The two datasets are restricted to their 11 shared object categories: ModelNet40-11 (5,227 objects; 4,381 train / 846 test, clean synthetic CAD meshes) and ScanObjectNN-11, hardest `PB_T50_RS` variant (11,955 objects; 9,535 train / 2,420 test, noisy real-world scans with background clutter and arbitrary orientation). Each of the 3 neural networks is trained from scratch for 8 epochs on ScanObjectNN-11 (1,024 points/object, mean-centered and unit-sphere normalized, with $\mathcal{N}(0, 0.01)$ jitter augmentation, Adam optimizer,

learning rate 10^{-3}, weight decay 10^{-4}, batch size 32), and its penultimate-layer embeddings are then evaluated with an external classification probe on ModelNet40-test. *X3DEnc*, undergoes no training (`res`$= 32$, $n_{\max} = 22$) reduced to its yaw-invariant ($SO(2)$) form (dimension 1,222), and only a lightweight external probe (logistic regression or a single-hidden-layer 256-unit MLP) is fit on top of it.

The cross-domain classification accuracy rate is shown in Table 5. We can observe that DGCNN (Wang et al., 2019), Mamba3D-lite (Han et al., 2024), and PointNet (Qi et al., 2017a) have achieved inferior performance than *X3DEnc*. This result indicates that a well-designed training-free descriptor remains a genuinely competitive, and in this case superior, alternative to several standard learned baselines under distribution shift. The more limited the training data, the more prominent advantage *X3DEnc* shows.

Table 5: `trainSONN`→`testMN40` cross-domain classification accuracy.

Method	Accuracy
DGCNN (2019)	11.6%
Mamba3D-lite (2024)	5.0%
PointNet (2017a)	4.0%
X3DEnc	**12.2%**

A.16 Extra Experiment 2: X3DEnc Performance on 3D Object Classification

Although *X3DEnc* focuses on 3D object geometry, color and pose encoding, we want to figure out how far a *X3DEnc* on object geometry encoding can be pushed on ModelNet40 (Wu et al., 2015) (official 9,843/2,468 train/test split, 40 categories) as progressively more design effort and learned components are layered on top, without abandoning the underlying spectral descriptor. The base descriptor is per-(n, l) rotation-invariant magnitude, computed for three geometric representations projected onto a shared 3D-Zernike basis ($n_{\max} = 30$ cache, truncated to the empirically-tuned sweet spot $n_{\max} = 20$): the mesh *surface* shell, a 64^3 voxelized *occupancy* shell, and the same shell *flood-filled to a solid*. Starting purely from *X3DEnc*, we further introduce self-supervised (SSL) masked-token pretraining - first on ModelNet40 itself, then on the larger and more diverse ShapeNet55 (Chang et al., 2015) (∼52k shapes).

Table 6: Overall Accuracy on ModelNet40 as design effort is progressively added on top of *X3DEnc*.

Configuration	Overall Accuracy
PlainZernike	76.1%
X3DEnc	85.8%
X3DEnc + SSL	**90.1%**
Reference: MoST (2025) (SOTA)	96.2%

Table 6 traces this progression. Two insights stand out. First, **a purely training-free X3DEnc pipeline is already a genuinely competitive result**: with zero learned backbone parameters reaches 85.8% overall accuracy, confirming that a deterministic spectral descriptor is a viable, non-trivial baseline rather than a weak strawman. Second, *X3DEnc* with self-supervised masked-token pretraining reaches to 90.1%. This confirms that, *if pretraining is permitted*, a carefully-designed *X3DEnc* based pipeline is competitive with much of the deep point-cloud literature, while a fully training-free variant remains a strong, non-trivial standalone result.