SIEVE: Overcoming Topological Obstruction in Equivariant Self-Supervised Learning

Jongann Lee *
SolverX, UIUC
Urbana, Illinois, USA
jongann2@illinois.edu

Sun Woo Park *
Max Planck Institute of Mathematics
Bonn, Germany
s.park@mpim-bonn.mpg.de

Yun Young Choi †
SolverX
Seoul, Republic of Korea
young@solverx.ai

Abstract

We identify a **theoretical incompatibility** between 3D point cloud masked autoencoding and SE(3)-equivariant networks. We prove that standard masking triggers a *topological collapse*, effectively forcing the model to hallucinate orientation without a reference frame. To resolve this, we introduce **SIEVE** (Scalar Invariant Extraction Via Equivariance), an SE(3)-equivariant self-supervised framework that compresses point cloud geometry into invariant scalar embeddings. At its core is the *spherical projection mask*, which preserves global pose while obfuscating local semantics, sidestepping the obstruction by construction. Experiments confirm that SIEVE circumvents the predicted collapse where constant and noise masks fail. On downstream point cloud tasks, the resulting embeddings improve over coordinate-based baselines and hand-crafted geometric descriptors, with the largest gains on generalization to unseen shape categories. Code is available at https://github.com/voltwin-dev/SIEVE_NeurIPS26

1 Introduction

Point clouds serve as the fundamental data structure for 3D understanding in various domains, including robotics [18] and autonomous driving [6]. Unlike 2D images, where *position* (the pixel grid) and *content* (RGB values) are naturally disentangled, point clouds present a unique challenge: the geometry *is* the feature. The coordinate $\mathbf{x} \in \mathbb{R}^3$ encodes both the local structure and the global spatial location. This entanglement necessitates representations that are not only topologically rich, a goal also pursued in graph representation learning [12, 13], but also consistent under global rigid-body transformations.

This separation naturally motivates the use of SE(3)-equivariant neural networks, whose architectures are constrained to ensure that a rigid-body transformation applied to the input results in an identical transformation of the output [3–5, 16, 39]. However, because these models cannot rely on absolute coordinate "shortcuts" [43, 47], they often face more difficult optimization landscapes than their non-equivariant counterparts [49]. We propose self-supervised learning (SSL) as the path forward: the equivariant pipeline can distill geometric structure into SE(3)-invariant embeddings that enrich downstream representations, sidestepping the need to impose equivariance on the downstream model.

However, we identify a fundamental theoretical flaw when applying standard masking strategies to SE(3)-equivariant neural networks. Conventional SSL methods replace masked features with a learnable, isotropic token [24, 32, 50]. Utilizing the theory of pullbacks of vector bundles, we demonstrate that the use of such constant tokens induces a topological obstruction, leading to a collapse in the vector pipeline that inhibits the reconstruction of the input point cloud. We empirically

*Equal contribution.
†Corresponding author: young@solverx.ai.

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

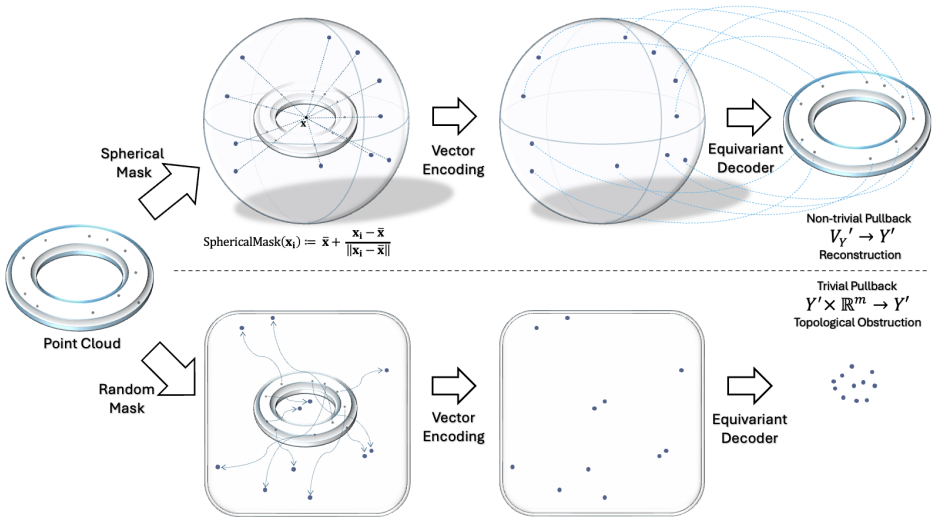

Figure 1: **Random masking causes representational collapse; spherical projection masking does not.** Random masking functions ignore subtle geometric data inherent in point clouds. This imposes topological obstructions in reconstructing the desired non-trivial vector bundles characterizing the underlying geometry of point clouds, as demonstrated in Theorem 2.6. As a result, representational collapse can occur as shown in Section 4.2. SIEVE utilizes a spherical projection mask, thereby possessing capabilities in reconstructing a class of pullbacks of vector bundles defined over a wide variety of 2 and 3-dimensional manifolds as demonstrated in Theorem 3.1.

validate this obstruction through a controlled experiment on a synthetic dataset, highlighting the catastrophic failure of isotropic masking in equivariant settings.

To resolve this, we propose the equivariant SSL framework **SIEVE** (Scalar Invariant Extraction Via Equivariance). Instead of masking information completely, SIEVE employs a *spherical projection masking strategy* that *sieves* the information: it projects the 'where' (global pose) onto the sphere while forcing the encoder to compress the 'what' (intrinsic semantics) into invariant scalars. This masking scheme is a direct consequence of our vector bundle analysis: it materializes the pullback of non-trivial vector bundles that standard masking cannot reconstruct. We validate these embeddings on several downstream point cloud tasks, where SIEVE improves over coordinate-based baselines and hand-crafted descriptors with the largest gains appearing on generalization to unseen categories.

2 Theoretical Framework

2.1 Preliminaries: Vector Bundles

Effective analysis of properties of 3D point clouds rests upon understanding entanglement between *position* and *content*. For this purpose, we seek suitable mathematical concepts or frameworks that encompass both topological features of 3D point clouds as well as contents measured over such point clouds as real vectors. We pay particular focus on using vector bundles from algebraic topology among many possible candidates that can harbor the subtlety of analyzing such entanglements.

Definition 2.1. Let X be a topological space. An n-dimensional vector bundle $\pi : V \to X$ is a topological space satisfying the following conditions. **(1)** The map π is a continuous surjection. **(2)** For every $x \in X$, the fiber $\pi^{-1}(x)$ is isomorphic to $\mathbb{R}^n$. **(3)** For every $x \in X$, there exists an open neighborhood $U \subset X$ such that satisfies the following three conditions. **(3a)** There exists a homeomorphism $\varphi : U \times \mathbb{R}^n \to \pi^{-1}(U)$. **(3b)** For every $v \in \mathbb{R}^n$, we have $\pi(\varphi(x, v)) = x$. **(3c)** Every isomorphism $\mathbb{R}^n \to \pi^{-1}(x)$ can be rewritten as an invertible $n \times n$ real matrix.

Interested readers may refer to Appendix A.1 for a detailed motivation for using the notion of vector bundles to formulate strengths and capabilities of deep learning algorithms. Given a vector bundle $\pi : V \to X$, a continuous function $f : X \to V$ such that $\pi(f(x)) = x$ for every $x \in X$ is called a global section. With these definitions, we propose the following topological reformulation of 3D

point clouds, up to some perturbations or negligible errors. **(1)** *3D point clouds* correspond to points on a smooth compact orientable 3-dimensional *manifold* Y embedded inside $\mathbb{R}^3$, whose boundary $X := \partial Y$ is a smooth compact orientable surface embedded inside $\mathbb{R}^3$. **(2)** Because 3D Point clouds consist of discrete set of points, we may require all the point clouds to lie over a complement of a small tubular neighborhood of the wedge sum of some circles $\vee_{i=1}^{g} S^1$ inside Y, where g is the genus of X. Such a space, denoted as Y', is homotopy equivalent to $X \times I$. **(3)** *Measurements* consisting of n-dimensional feature vectors over each point cloud $x \in Y'$ corresponds to *global sections* of an n-dimensional vector bundle $V \to Y'$.

It is natural to understand vector bundles that can harbor non-trivial entanglements between geometry and intrinsic semantics which 3D point clouds contain. This brings us to the notion of trivial vector bundles over topological spaces.

Definition 2.2. Let X be a topological space. An n-dimensional trivial vector bundle $V \to X$ is a vector bundle over X which is isomorphic to the vector bundle $X \times \mathbb{R}^n \to X$. Any continuous function $f : X \to \mathbb{R}^n$ is a global section of the trivial vector bundle.

The geometry of X does not impose any restrictions on the image of the global section $f : X \to \mathbb{R}^n$. We hence obtain the following insight: **trivial vector bundles** over topological spaces are suitable for analyzing data whose *position* and *content* are **disentangled**. We also observe that vector bundles over topological spaces which are **non-trivial** are potential candidates that can illuminate **entanglements** between geometry and intrinsic semantics of 3D point clouds. In light of these insights, we ask the following question: *What are effective ways to learn global sections of real oriented non-trivial vector bundles over a given topological space Y (or Y')?*

2.2 Interpretation: Simplifications with Projections to S^2 using Pullback of Vector Bundles

To understand the question, we analyze vector bundles over Y (or Y') that may be underlying spaces for 3D point clouds. The problem of classifying real oriented vector bundles over smooth manifolds, however, is a highly non-trivial problem. Nevertheless, we can construct some of these desired vector bundles by using pullbacks of vector bundles between topological spaces.

Definition 2.3. Given a continuous map $f : X \to Y$, and a vector bundle $\pi : V \to Y$, the pullback of V is the vector bundle $f^*\pi : f^*V \to X$, where $f^*V := \{(x, e) \in X \times V : f(x) = \pi(e)\}$. The fiber of f^*V at $x \in X$ is isomorphic to the fiber of V at $f(x) \in Y$.

Without loss of generality, we consider an auxiliary (possibly unknown) n-dimensional vector bundle $V_Y \to Y'$ underlying 3D point clouds, where $n \geq 3$. Because Y' and X are homotopy equivalent, there exists a bijection of isomorphism classes of vector bundles of Y' and X (see Corollary 1.8 of [23]). Hence, we can identify the n-dimensional vector bundle $V_Y \to Y'$ with an n-dimensional vector bundle $V_X \to X$. (This is one of the reasons why we chose Y' instead of Y, because we may reduce the dimension of the underlying topological space of our interest from 3 to 2.)

We can use pullbacks of vector bundles to construct trivial and non-trivial vector bundles over X. Given any continuous map $f : X \to S^2$, we can pullback some n-dimensional vector bundle $V \to S^2$ to construct a vector bundle $f^*V \to X$. Thankfully, the classification of real oriented vector bundles over S^2 is very well understood, whose proof is in Appendix A.2.

Theorem 2.4. *Let $V \to S^2$ be a real oriented vector bundle of dimension $n \geq 3$. Suppose V is a non-trivial vector bundle, i.e. V contains the tautological bundle $\pi_t : T \to S^2$ as a sub-vector bundle (see Appendix A.2 for its definition). Then V is isomorphic to the **unique** non-trivial vector bundle of dimension n, whose fibers are isomorphic to $\mathbb{R}^{n-2} \oplus \pi_t^{-1}(x)$ for every $x \in S^2$.*

There are two isomorphism classes of vector bundles over X we can construct using pullback of vector bundles of S^2: the trivial vector bundle $X \times \mathbb{R}^n \to X$, obtained as a pullback of the trivial vector bundle $S^2 \times \mathbb{R}^n \to S^2$, and a non-trivial vector bundle $V_X \to X$, obtained as a pullback of the unique non-trivial vector bundle $V \to S^2$.

2.3 Applications: Topological Obstructions in Constant and Randomized Noise Masking

The main application of our paper will focus on effectively approximating 3D point clouds, which shall be regarded as a set of discrete points $\mathcal{X}$ over Y'. Given an injection $i : \mathcal{X} \hookrightarrow Y'$, we consider

the restriction of the global section $f \circ i : \mathcal{X} \to Y' \to V'_Y$. We regard the problem of approximating 3D point clouds as a problem of approximating the restriction of the global section $f \circ i$.

To make the notion of approximation $f \circ i$ rigorous, it suffices to clarify what it means for an $\mathrm{SE}(3)$-equivariant neural network Ψ to effectively approximate a global section $f : Y' \to V'_Y$. To do this, we identify a global section $f : Y' \to V'_Y$ of an N-dimensional vector bundle V'_Y over a 3-dimensional manifold Y' constructed in Section 2.2 as a restriction of the continuous function $F : \mathbb{R}^3 \to \mathbb{R}^{m_N}$ to Y' (or any connected compact subspace of $\mathbb{R}^3$ homeomorphic to Y') that is equivariant under the action of $\mathrm{SE}(3)$. Here, the quantity m_N may not necessarily be equal to N, but can be defined as the minimal dimension $m_N \geq N$ of a trivial vector bundle $\mathcal{T}'_Y$ over Y' which contains V'_Y as a vector sub-bundle. We note that this quantity is guaranteed to exist using Proposition A.2, whose statement appears in the next subsection. More concretely, the relation between f and F is given as follows. Given an inclusion $j : Y' \to \mathbb{R}^3$ and a projection $\pi : \mathcal{T}'_Y \to V'_Y$, we have

$$f = \pi \circ F \circ j : Y' \to \mathbb{R}^3 \to \mathbb{R}^{m_N} \to \mathbb{R}^N. \tag{1}$$

Using the analogous argument, we note that the restriction of a global section $f \circ i$ over a 3D point cloud $\mathcal{X}$ can be obtained from a continuous function $F : \mathbb{R}^3 \to \mathbb{R}^{m_N}$ via the composition $\pi \circ F \circ j \circ i$.

Based on this identification, we define what it means for an $\mathrm{SE}(3)$-equivariant neural network to effectively approximate a global section. Similar definitions can be found in Theorem 11 of [27] and Theorem 4.1 of [35].

Definition 2.5. Let Ψ be an $\mathrm{SE}(3)$-equivariant neural network whose activation functions are non-constant and Lipschitz continuous. We say that Ψ effectively approximates a global section $f : Y' \to V'_Y$ if for any $\epsilon > 0$ the neural network Ψ satisfies the following inequality:

$$\sup_{z \in Y' \subset \mathbb{R}^3} |\Psi(z) - F(z)| < \epsilon. \tag{2}$$

Given a discrete point cloud $\mathcal{X}$ on Y' endowed with an injection $i : \mathcal{X} \to Y'$, we say that Ψ effectively approximates a restriction of a global section $f \circ i : \mathcal{X} \to V'_Y$ if for any $\epsilon > 0$ the neural network Ψ satisfies the following inequality:

$$\sup_{z \in \mathcal{X} \subset \mathbb{R}^3} |\Psi(z) - F(z)| < \epsilon. \tag{3}$$

By formulating the notion of effective approximation using continuous functions over manifolds, instead of discrete point clouds, we can apply Definition 2.5 to any 3D point cloud $\mathcal{X}$ on Y' once we specify an injection $i : \mathcal{X} \to Y'$. By doing so, one can hope to ensure robustness in approximating the desired restriction $f \circ i$ given any arbitrary 3D point cloud $\mathcal{X}$ over Y'.

With the theoretic framework in place, we shall analyze potential topological obstructions imposed on certain masking functions. Given an integer N, let $\mathfrak{m} : X \to \mathbb{R}^N$ be a masking function that does not lift to a section of a non-trivial vector bundle $V_X \to X$. As an example, we may consider constant and randomized masking functions, which are default choices for reconstruction based SSL in both images and point clouds. We note that the image $\mathfrak{m}(\mathbf{x})$ is chosen randomly independent of the geometric features of $\mathbf{x}$. We can hence interpret the application of the masking function $\mathfrak{m}$ as replacing V_X with $\mathbb{R}^N \times X$. Because the trivial vector bundle over X is the pullback of the trivial vector bundle over S^2, the masking operator by $\mathfrak{m}$ corresponds to replacing V_X with $f^*(\mathbb{R}^N \times S^2)$. There is a wide class of non-trivial vector bundles over X which may not admit a smooth embedding inside $f^*(\mathbb{R}^N \times S^2)$. A summary of our aforementioned findings, focusing on constant masking functions, leads to the following result, whose proof is in Appendix A.3.

Theorem 2.6. *Let Y' be the underlying 3-dimensional smooth orientable manifold constructed from Section 2.2 which represents the underlying space for a given 3D point cloud. Let $\mathfrak{m} : X \to \mathbb{R}$ be any constant masking function. Then any $\mathrm{SE}(3)$-equivariant neural network with the masking function $\mathfrak{m}$ cannot effectively approximate global sections of any non-trivial vector bundles $V'_Y \to Y'$ (as in Definition 2.5).*

Any $\mathrm{SE}(3)$-equivariant neural network with the randomized masking function $\mathfrak{m}$ cannot effectively approximate global sections of any non-trivial vector bundle of dimension at least N. Even for lower dimensional non-trivial vector bundles, there are many challenges that would need to be addressed to effectively approximate such global sections, see Figure 1 and Appendix A.3 for more details.

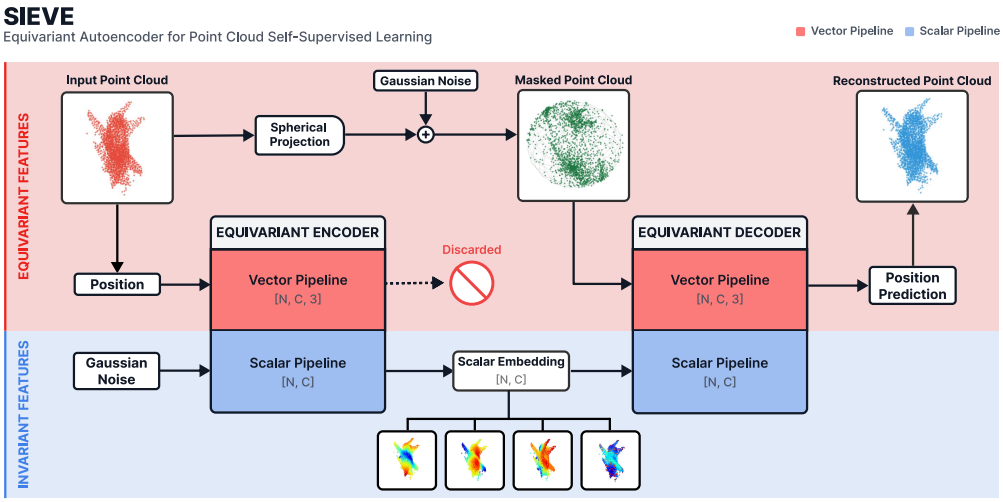

Figure 2: **SIEVE Framework.** SIEVE extracts SE(3)-invariant embeddings using self-supervised reconstruction on an equivariant bottleneck. The equivariant encoder compresses raw point coordinates into an invariant scalar encoding, while the vector pipeline is blocked. The spherical projection mask is applied to the input point cloud and supplied as the new vector embedding. It and the semantic details from the scalar embedding are used by the equivariant decoder to reconstruct the point cloud on a node-level.

3 SIEVE: Scalar Invariant Extraction Via Equivariance

To circumvent the topological obstruction stated in Theorem 2.6, we propose **SIEVE**, an SSL framework that utilizes the *spherical projection mask* to maintain a continuous section in the vector bundle, thereby ensuring the representational completeness of the equivariant pipeline. SIEVE adopts a framework inspired by the Masked Autoencoder (MAE), comprising an encoder, a geometric masking operator, and a decoder [24]. The framework is based on two core design principles.

Equivariant Feature Streams. We leverage SE(3)-equivariant neural networks to process the data through two streams: a vector pipeline and a scalar pipeline (Figure 2). The encoder maps the input point cloud $\mathcal{X}$ to an SE(3)-invariant scalar embedding s_{code}, capturing semantic information independent of the point cloud's global pose.

Semantic-Geometric Decomposition. Unlike standard MAE, which masks a subset of nodes entirely, SIEVE employs a *spherical projection mask*. This operator maps all the points in $\mathcal{X}$ to a unit sphere, obfuscating local semantic features (the "what") while preserving global spatial properties such as orientation and position (the "where"). This sidesteps the topological obstructions of constant or randomized vector masks, admitting a wider range of geometric entanglements for reconstruction.

3.1 Spherical Projection Mask

Standard masking in SSL typically replaces masked node features with a learnable constant token $\mathbf{z} \in \mathbb{R}^C$. However, applying such an isotropic mask to the vector pipeline of an SE(3)-equivariant neural network leads to a collapse of the vector output, as stated in Theorem 2.6.

To circumvent this collapse, we propose the *spherical projection mask*. Given an input point cloud $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N_p} \subset \mathbb{R}^3$, let $\overline{\mathbf{x}} = \frac{1}{N_p}\sum_{i=1}^{N_p} \mathbf{x}_i$ denote the mean of the point cloud. The spherical projection operator projects each point $\mathbf{x}_i$ onto a unit sphere S^2 centered on the mean $\overline{\mathbf{x}}$:

$$\text{SphericalMask}(\mathbf{x}_i) = \overline{\mathbf{x}} + \frac{\mathbf{x}_i - \overline{\mathbf{x}}}{\|\mathbf{x}_i - \overline{\mathbf{x}}\|_2}. \tag{4}$$

Using the mean $\overline{\mathbf{x}}$ as the center guarantees that projected nodes translate identically to the input nodes. Thus, the spherical projection mask is an SE(3)-equivariant operator, ensuring that SIEVE's vector pipeline is equivariant.

A naive projection leaves the scalar radius $r = \|\mathbf{x}_i - \overline{\mathbf{x}}\|$ as a trivial shortcut, leading to a representational collapse of the embeddings as demonstrated in Appendix C.3. To mitigate this, we inject i.i.d. Gaussian noise into the projected coordinates. Further mitigation strategies are discussed in Appendix B.1.

3.2 SE(3)-Equivariant Encoder and Decoder

The encoder and decoder in SIEVE are SE(3)-equivariant neural networks. Each layer has two feature streams: a scalar stream representing invariant features and a vector stream representing steerable features, with cross-stream interaction at every layer. We denote these feature pairs as $[s, \mathbf{v}]$, where $s \in \mathbb{R}^{N_p \times h_1}$ and $\mathbf{v} \in \mathbb{R}^{N_p \times h_2 \times 3}$.

Encoder: The SIEVE encoder extracts the SE(3)-invariant semantic information from the input geometry. The point cloud $\mathcal{X}$ is the vector input. The choice of scalar input becomes critical: we demonstrate in Appendix C.2 that uniform scalar inputs prevent SIEVE from reconstructing the point cloud, so we provide i.i.d. Gaussian noise $s_{in} \sim \mathcal{N}(0, I) \in \mathbb{R}^{N_p}$ as the scalar feature. The encoding process is defined as

$$[s_{code}, \mathbf{v}_{code}] = \text{Encoder}([s_{in}, \mathcal{X}]) \tag{5}$$

Following the forward pass, we impose an equivariant bottleneck by discarding the vector output $\mathbf{v}_{code}$. The invariant scalar embedding s_{code} is retained as the compressed semantic representation of the point cloud.

Decoder: The decoder aims to reconstruct the original point cloud $\mathcal{X}$ using the semantic encoding s_{code} and the masked geometry $\mathbf{x}_{masked}$ as defined in section 3.1. The decoder's vector output $\hat{\mathcal{X}}$ serves as the reconstructed point cloud, while the scalar output s_{out} is discarded.

$$[s_{out}, \hat{\mathcal{X}}] = \text{Decoder}([s_{code}, \mathbf{x}_{masked}]) \tag{6}$$

Backbone Selection: While SIEVE is agnostic to specific SE(3)-equivariant architectures, we employ the Geometric Algebra Transformer (GATr) as our backbone [5]. GATr's global attention captures long-range dependencies. It also uses memory efficient attention [28] and avoids expensive spherical harmonics, allowing for better scalability compared to other equivariant neural networks.

3.3 Learning Objective

The objective of SIEVE is to perform node level reconstruction, wherein each individual point $\mathbf{x}_i \in \mathcal{X}$ is mapped back to its original position. We accomplish this by using a standard $\mathcal{L}1$ loss between the original point cloud $\mathcal{X}$ and the reconstructed point cloud $\hat{\mathcal{X}}$:

$$\mathcal{L}1 = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_1. \tag{7}$$

3.4 A Mathematical Reformulation of SIEVE

Using the theoretical motivation for SIEVE, we deduce theoretical strengths of our proposed method in comparison to other masking schemes, whose proof is in Appendix A.4.

Theorem 3.1. *Let Y' be the underlying 3-dimensional smooth orientable manifold constructed from Section 2.2. Then SIEVE can approximate global sections of any vector bundles which are isomorphic to pullbacks of any vector bundles of S^2.*

The idea of the proof originates from an observation that application of $\mathrm{SphericalMask}$ corresponds to utilizing pullback of vector bundle $V \to S^2$ which contains the tautological bundle $T \to S^2$ as a vector sub-bundle. We thus deduce that SIEVE, unlike constant or random masking techniques, has theoretical capabilities in approximating all possible vector bundles over a wide range of classes of 2-dimensional or 3-dimensional closed oriented manifolds which arise from pullback of both trivial

and non-trivial vector bundles over S^2. To support this, we present experimental demonstrations of SIEVE's theoretical capabilities in Section 4.2.

4 Experiments

4.1 Pre-training

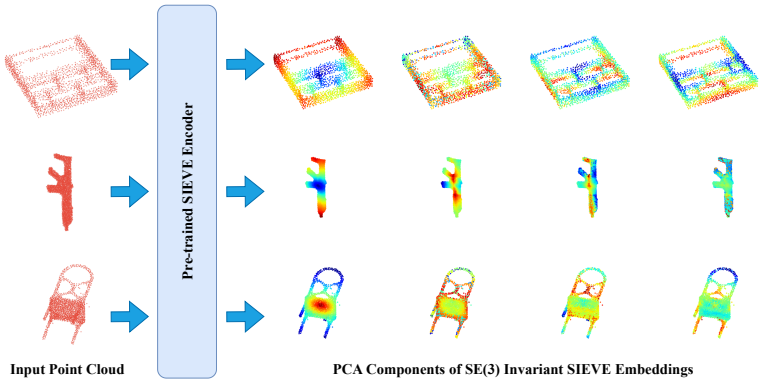

Figure 3: **Pre-trained SIEVE Encoder Embeddings.** We visualize the PCA components of the pre-trained SIEVE embeddings. These embedding are invariant to point cloud rotation and translation.

We pre-train SIEVE on the OpenShape dataset, which consists of approximately 876k 3D models [30]. Each point cloud is generated by sampling 2048 points via furthest point sampling. We utilize the entire dataset for pre-training without a train-validation split to maximize geometric diversity. Full pre-training details are in Appendix B.1.

4.2 Empirical Verification of Collapse

In Section 2, we theoretically established that SE(3) equivariant networks cannot output two point clouds of different topology with a constant or noise vector token as input. To verify this, we perform an experiment using a synthetic dataset of two objects with differing topologies: a solid cuboid (homeomorphic to the 3-disk D^3) and a square toroid (homeomorphic to $S^1 \times S^1$). The dataset consists of 120 instances per class, partitioned into a 100/20 train-test split. Failure on a dataset this simple indicates a fundamental representational issue rather than a capacity or data limitation.

To demonstrate that this limitation is inherent to the equivariance constraint rather than a specific architecture, we evaluate two distinct backbones: GATr [5] and SE(3)-Transformer [16]. We compare three masking strategies: a constant vector mask utilizing a learnable isotropic token, an ellipsoid noise mask sampled from an ellipsoid spectrally aligned with the point cloud's principal components (and thus is SE(3)-equivariant), and our proposed spherical projection mask. All three models use identical initialization and hyperparameter settings to ensure a fair comparison.

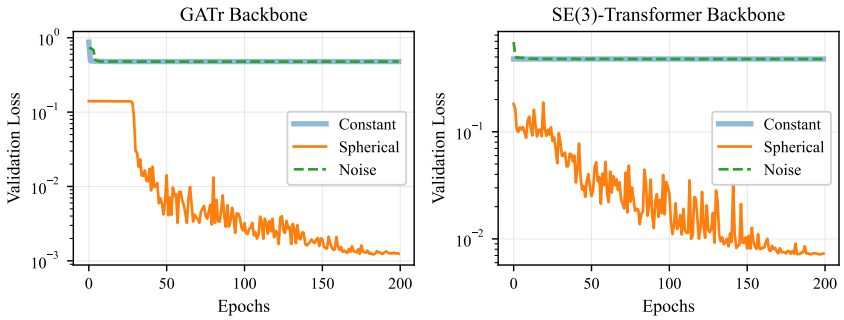

Figure 4: Validation Loss of SIEVE on two baselines: GATr and SE(3)-Transformer. Trained on the synthetic dataset.

As shown in Figure 4, models that utilize the spherical projection mask successfully converge, achieving a validation $\mathcal{L}1$ loss below 0.01. Conversely, models employing the constant vector mask or the ellipsoid noise mask fail to converge, with validation and training losses stagnating around 0.5. This failure to minimize even the training loss demonstrates a fundamental representational collapse rather than a deficit in model capacity or data volume.

A visual inspection of the reconstructed point clouds (Figure 5) further demonstrates the predicted representational collapse of Theorem 2.6. While SIEVE with the spherical projection mask achieves near-perfect reconstruction, SIEVE with the constant vector mask or the ellipsoid noise mask outputs a near zero-dimensional 'blob'. As the proof of Theorem 2.6 suggests, these two masking schemes can only represent global sections of vector bundles of dimension at most 1. Hence, they cannot effectively represent local coordinate functions for 3D point clouds.

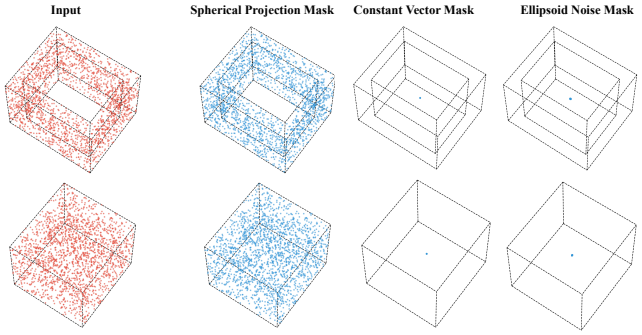

Figure 5: **Reconstruction Comparison and Representational Collapse.** We evaluate the reconstruction *(blue)* of the input point cloud *(red)* under different masking strategies. Using our spherical projection mask, the model achieves near-perfect reconstruction. Reconstructing with a constant vector mask or the ellipsoid noise mask results in a catastrophic representational collapse. We refer to Appendix A.3 for further discussions.

Reconstruction experiments on a dataset with identical topology confirms that it is the topological variety that causes the collapse; constant and noise mask SIEVE have no issues learning to reconstruct such datasets. (Appendix C.1)

4.3 Downstream Tasks

This section evaluates whether s_{code} contains useful, transferable geometric information. We do not target SOTA on any specific benchmark, as SOTA architectures are heavily tuned to their target tasks. Instead we test whether SIEVE consistently improves generalist point cloud architectures (PointNet++, DGCNN [37, 45]). Integration is minimal: the SIEVE encoder is frozen, and s_{code} is concatenated with the raw coordinates at the input layer. This drop-in usage is enabled by SIEVE's node-level granularity, which aligns one-to-one with input points, unlike patch-based SSL. We therefore evaluate on downstream tasks where such granularity is critical.

4.3.1 Molecular Segmentation

We first evaluate the learned representations on the molecular segmentation task which consists of 640 RNA molecules segmented into 120 categories [36]. Following the protocol in [21] we resample each mesh to 2048 points and utilize the standard 80-20 train test split.

To demonstrate that SIEVE embeddings transfer across architectures, we evaluate with three distinct architectures: PointNet++ [37], DeltaNet [46], and DiffusionNet [40]. For each model, the frozen SIEVE scalar embedding s_{code} is concatenated with the raw node coordinates $\mathbf{x}_i$ as input. We compare our learned representation against two SE(3)-invariant geometric representations: the distance from mean r and curvature κ_1, κ_2. As shown in Table 1, SIEVE embeddings consistently outperform the "position-only" baselines and the geometry-enhanced models across all three architectures.

Table 1: Test accuracies(%) on the RNA Segmentation Task

MODEL	xyz	r	κ_1, κ_2	**SIEVE**
POINTNET++	71.55	75.54	72.90	**86.28**
DELTANET	87.21	88.07	86.86	**89.58**
DIFFUSIONNET	81.20	85.89	85.97	**89.30**

4.3.2 3D Shape Completion

3D-EPN is a point cloud completion dataset consisting of 16 categories: 8 categories are included in both the training and test sets ("Seen"), while the remaining 8 appear only in the test set ("Unseen") [14]. We utilize PCN as our baseline and compare the performance with and without the addition of SIEVE encodings [51].

As shown in Table 2, SIEVE consistently reduces the Chamfer Distance (CD) loss across both seen and unseen categories. The performance gain is most pronounced on unseen categories, with a 12.5% reduction in the average loss compared to the 1.62% reduction on seen categories. This suggests that the geometric rigor distilled into SIEVE embeddings provides a more robust structural prior than coordinate-based features alone, facilitating superior zero-shot generalization to novel topological structures.

Table 2: Reconstruction $\mathcal{L}2$ CD Loss $(\times 10^4)$ on the 3D-EPN Dataset

SEEN CATEGORIES	PCN	PCN + SIEVE	UNSEEN CATEGORIES	PCN	PCN + SIEVE
AIRPLANE	1.6508	1.8674	BUS	4.2137	4.2908
CABINET	4.8287	4.3237	BED	33.1502	29.4725
CAR	2.6543	2.7663	BOOKSHELF	13.6558	12.5375
CHAIR	5.7248	5.6236	BENCH	7.4325	6.3300
LAMP	7.4924	7.2977	GUITAR	7.4999	5.2607
SOFA	5.7775	5.6728	MOTORBIKE	9.4908	8.9940
TABLE	4.4875	4.6435	SKATEBOARD	7.7232	6.4723
VESSEL	4.4766	4.2968	PISTOL	12.7125	10.2674
SEEN AVERAGE	4.6366	4.5615	UNSEEN AVERAGE	11.9848	10.4869

4.3.3 Geometric Registration

Geometric registration is the prediction of a rigid body transformation to align one shape to another. We follow the task and architecture in Deep Closest Point (DCP) [44]. A point cloud is sampled from the ModelNet40 dataset [48], which is then cloned, point order shuffled, then transformed with a random rigid body transformation. We embed the SIEVE embeddings to DCP baseline and test with two train-test splits: one where all 40 categories are in both sets, and the other where 20 are used in train and the other 20 are reserved for testing.

The results show that SIEVE embeddings improve both the shared and separate category registration, with more dramatic gains in the separate category. The geometric registration performance with SIEVE is much more robust to unseen objects. This pattern is consistent with the 3D-EPN results: SIEVE embeddings encode intrinsic geometric structure that transfers across categories.

5 Related Work

Point Cloud Self-Supervised Learning. Inspired by the success of Masked Autoencoders in images [24], reconstruction based point cloud SSL has largely adopted the 3D patch reconstruction framework established by PointBERT [50]. These methods typically employ a binary masking strategy on local patches, where specific regions of the point cloud are entirely removed from the input [9, 29, 32]. However, these frameworks lack inherent SE(3)-symmetry and their embeddings remain sensitive to global rotations and translations. RI-MAE pursues rotation invariance without equivariant architectures by PCA-aligning patches and reconstructing in latent space [41]. RIVQ-

Table 3: Point Cloud Registration Results on the ModelNet40 Dataset

MODEL	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
SHARED CATEGORIES						
DCP	1.475760	1.214808	0.826500	0.000004	0.002040	0.001239
DCP + SIEVE	0.703465	0.838728	0.479402	0.000002	0.001539	0.000774
SEPARATE CATEGORIES						
DCP	6.948212	2.635946	1.782947	0.000025	0.005048	0.003669
DCP + SIEVE	0.874955	0.935390	0.511852	0.000002	0.001509	0.000813

VAE learns a discretized, SE(3)-invariant patch representation using TFN [31]. All of these methods produce patch embeddings, requiring fine tuning with a task-specific head for downstream tasks. SIEVE instead uses equivariant networks and a spherical projection mask, producing node-level SE(3)-invariant embeddings that concatenate cleanly with raw coordinates as input to any downstream architecture, without fine-tuning the encoder.

Equivariant Neural Networks. To date, numerous equivariant architectures have been proposed [2, 7, 10, 15, 34]. However, many of these models have focused on molecular biology applications [20]; QM9 and MD17 molecular prediction datasets are some of the most popular benchmarks for equivariant neural networks [11, 38]. Molecules possess a significant SE(3)-invariant node feature, the atomic number, which distinguishes the domain from purely geometric data like point clouds. Consequently, the only existing work that combines SSL with SE(3)-equivariant networks is not directly applicable to point clouds due to this difference in domain [1]. To our knowledge, this is the first work to combine point cloud SSL with SE(3)-equivariant neural networks for general geometric learning.

6 Conclusion

In this work, we theoretically established that conventional masking strategies in geometric self-supervised learning induce a representational collapse when applied to SE(3)-equivariant architectures. Using the theory of vector bundles, we showed that isotropic or randomized masking functions which ignore innate geometric features of point clouds do not have enough capabilities to reconstruct such point clouds, because they cannot effectively reconstruct a wide range of classes of non-isomorphic vector bundles.

Based on these theoretical observations, we proposed SIEVE, a pioneering framework that bridges the gap between equivariance and SSL for point clouds. By utilizing a novel spherical projection masking strategy, SIEVE circumvented the topological obstructions inherent in isotropic masking, enabling robust geometric reconstruction. Our experiments empirically confirmed the representational collapse and demonstrated that SIEVE learns highly generalizable embeddings that outperform traditional descriptors in downstream tasks. To our knowledge, SIEVE is the first framework to combine self-supervised learning with SE(3)-equivariant networks for general point cloud geometry, and we hope it spurs further work at this intersection.

Acknowledgments and Disclosure of Funding

This research was supported by the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT).

This work was initiated while Jongann Lee was a paid intern at SolverX in the Republic of Korea and continued after the internship concluded. Following the internship, Jongann Lee was no longer employed by SolverX and received no monetary compensation from SolverX for his subsequent contributions. SolverX continued to provide in-kind research support through access to GPU computing resources and LLM-based tools.

Sun Woo Park thanks the Max Planck Institute for Mathematics for its generous support.

Competing interests. Jongann Lee was previously a paid intern at SolverX in the Republic of Korea. SolverX subsequently continued to provide the in-kind research resources described above. The authors declare no other competing interests.

References

[1] J. An, C. Qu, Y.-F. Shi, X. Liu, Q. Tang, F. Cao, and Y. Qi. Equivariant masked position prediction for efficient molecular representation. In *International Conference on Learning Representations (ICLR)*, 2025.

[2] S. Aykent and T. Xia. GotenNet: Rethinking efficient 3D equivariant graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2025.

[3] E. J. Bekkers, S. Vadgama, R. D. Hesselink, P. A. V. der Linden, and D. W. Romero. Fast, expressive SE(n) equivariant networks through weight-sharing in position-orientation space. In *International Conference on Learning Representations (ICLR)*, 2023.

[4] J. Brandstetter, R. Hesselink, E. van der Pol, E. Bekkers, and M. Welling. Geometric and physical quantities improve E(3) equivariant message passing. In *International Conference on Learning Representations (ICLR)*, 2022.

[5] J. Brehmer, P. de Haan, S. Behrends, and T. Cohen. Geometric algebra transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[6] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020.

[7] J. Cen, A. Li, N. Lin, Y. Ren, Z. Wang, and W. Huang. Are high-degree representations really unnecessary in equivariant graph neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] G. Chen, M. Wang, Y. Yang, K. Yu, L. Yuan, and Y. Yue. PointGPT: Auto-regressively generative pre-training from point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[10] H. Chen, S. Liu, W. Chen, H. Li, and R. Hill. Equivariant point network for 3D point cloud analysis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

[12] Y. Y. Choi, S. W. Park, Y. Woo, and U. J. Choi. Cycle to clique (cy2c) graph neural network: A sight to see beyond neighborhood aggregation. In *The eleventh international conference on learning representations*, 2022.

[13] Y. Y. Choi, S. W. Park, M. Lee, and Y. Woo. Topology-informed graph transformer. *arXiv preprint arXiv:2402.02005*, 2024.

[14] A. Dai, C. Ruizhongtai Qi, and M. Nießner. Shape completion using 3D-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas. Vector Neurons: A general framework for SO(3)-equivariant networks. *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2021.

[16] F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[17] A. Ghanwat and D. M. Pancholi. Embeddings of 4–manifolds in $\mathbb{CP}^3$. *arXiv preprint arXiv:2002.11299*, 2015.

[18] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations (ICLR)*, 2023.

[19] A. Haefliger and M. W. Hirsch. On the existence and classification of differentiable embeddings. *Topology*, 2(1-2):129–135, 1963.

[20] J. Han, J. Cen, L. Wu, Z. Li, X. Kong, R. Jiao, Z. Yu, T. Xu, F. Wu, Z. Wang, et al. A survey of geometric graph neural networks: Data structures, models and applications. *Frontiers of Computer Science*, 19(11):1911375, 2025.

[21] J. Harrison, J. Benn, and M. Sermesant. Improving neural network surface processing with principal curvatures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[22] A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, England, 2002.

[23] A. Hatcher. Vector bundles and K-theory, 2017. URL `https://pi.math.cornell.edu/~hatcher/VBKT/VBpage.html`.

[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022.

[25] S. Hong, Y. Kwon, D. Shin, J. Park, and N. Kang. DeepJEB: 3d deep learning-based synthetic jet engine bracket dataset. *Journal of Mechanical Design*, 147(4):041703, 2025.

[26] A. Kratsios and L. Papon. Universal approximation theorems for differentiable geometric deep learning. *Journal of Machine Learning Research*, 23:1–73, 2022.

[27] W. Kumagai and A. Sannai. Universal Approximation Theorem for Equivariant Maps by Group CNNs. *arXiv preprint arXiv:2012.13882*, 2020.

[28] B. Lefaudeux, F. Massa, D. Liskovich, W. Xiong, V. Caggiano, S. Naren, M. Xu, J. Hu, M. Tintore, S. Zhang, P. Labatut, D. Haziza, L. Wehrstedt, J. Reizenstein, and G. Sizov. xFormers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`, 2022.

[29] D. Liang, X. Zhou, W. Xu, X. Zhu, Z. Zou, X. Ye, X. Tan, and X. Bai. PointMAMBA: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[30] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su. OpenShape: Scaling up 3D shape representation towards open-world understanding. *Advances in neural information processing systems (NeurIPS)*, 2023.

[31] M. Mezghanni, M. Boulkenafed, and M. Ovsjanikov. RIVQ-VAE: Discrete rotation-invariant 3D representation learning. *2024 International Conference on 3D Vision (3DV)*, 2024.

[32] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision (ECCV)*, 2022.

[33] S. W. Park, Y. Y. Choi, U. J. Choi, and Y. Woo. A functorial formulation of neighborhood aggregating deep learning. *arXiv preprint arXiv:2604.24672*, 2026.

[34] R. P. Pelaez, G. Simeon, R. Galvelis, A. Mirarchi, P. Eastman, S. Doerr, P. Thölke, T. E. Markland, and G. D. Fabritiis. TorchMD-net 2.0: Fast neural network potentials for molecular simulations. *arXiv preprint arXiv:2402.17660*, 2024.

[35] P. Petersen and F. Voigtlaender. Equivalence of approximation by Convolutional Neural Networks and Fully-connected Networks. *Proceedings of the American Mathematical Society*, 148 (4):1567–1581, 2020.

[36] A. Poulenard, M. Rakotosaona, Y. Ponty, and M. Ovsjanikov. Effective rotation-invariant point CNN with spherical harmonics kernels. In *International Conference on 3D Vision (3DV)*, 2019.

[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[38] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[39] V. G. Satorras, E. Hoogeboom, and M. Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

[40] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.

[41] K. Su, Q. Wu, P. Cai, X. Zhu, X. Lu, Z. Wang, and K. Hu. RI-MAE: Rotation-invariant masked autoencoders for self-supervised point cloud representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[42] C. Wall. All 3-manifolds imbed in 5-space. *Bulletin of the American Mathematical Society*, 71 (3):564–567, 1965.

[43] D. Wang, X. Zhu, J. Park, M. Jia, G. Su, R. Platt, and R. Walters. A general theory of correct, incorrect, and extrinsic equivariance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[44] Y. Wang and J. M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2019.

[45] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic Graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[46] R. Wiersma, A. Nasikun, and E. E. K. Hildebrandt. DeltaConv: Anisotropic operators for geometric deep learning on point clouds. *Transactions on Graphics*, 41(4), 2022.

[47] X. Wu, D. DeTone, D. Frost, T. Shen, C. Xie, N. Yang, J. Engel, R. Newcombe, H. Zhao, and J. Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[49] Y. Xie and T. Smidt. A tale of two symmetries: Exploring the loss landscape of equivariant models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[50] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[51] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert. PCN: Point completion network. In *International Conference on 3D Vision (3DV)*, 2018.

A Mathematical Background

We recommend interested readers to review [22] for an introduction to algebraic topology (where definitions of homotopy, homotopy equivalence, homotopy groups, closed orientable manifolds, and fiber bundles can be found), and Section 1 of [23] for an introduction to the theory of vector bundles over topological spaces.

A.1 Motivation for using vector bundles

We motivate the formulation of deep learning techniques as an approximator of global section of vector bundles by utilizing insights from the main results of Park et al [33].

Universal approximation theorem: The classical formulations of universal approximation theorem states that a feedforward neural network of arbitrary depth with non-polynomial activation function can approximate any given continuous function $f : \mathbb{R}^m \to \mathbb{R}^n$. Such an approximation theorem ensures a neural network with arbitrary depth (or arbitrarily many weights and parameters) to find any auxiliary relation among collections of datasets, as long as they embed in Euclidean space $\mathbb{R}^m$.

However, such a formulation of the universal approximation theorem does not necessarily imply that a neural network with arbitrary depth (or arbitrarily many weights and parameters) can find auxiliary relations among collections of datasets defined over manifolds M which are not homeomorphic to $\mathbb{R}^m$. For example, given a manifold M with open affine charts $\{U_\alpha\}_{\alpha\in\mathcal{A}}$, a neural network can approximate a continuous function $f_\alpha : U_\alpha \to \mathbb{R}^m$ for each $\alpha \in \mathcal{A}$, but one cannot guarantee that such a neural network will learn how to coherently glue such functions $f_\alpha : U_\alpha \to \mathbb{R}^m$. In terms of datasets associated to 3D point clouds, the functions $f_\alpha : U_\alpha \to \mathbb{R}^m$ correspond to measurements taken over 3D point clouds which constitute a part of (but not all) underlying geometric space.

Using vector bundles: By using the notion of vector bundles (see Definition 2.1 for its definition), we can reformulate the problem of gluing functions f_α as follows. Given a vector bundle $V \to M$, each function $f_\alpha : U_\alpha \to \mathbb{R}^m$ corresponds to local sections of the vector bundle restricted to the open affine chart U_α. Here, the term "local" means that the function f_α is only well defined over the neighborhood U_α, and it is not necessarily defined over M. One can construct a global section $f : M \to V$ by ensuring that the collection of functions $\{f_\alpha\}_{\alpha\in\mathcal{A}}$ satisfy the condition that given any two indices $\alpha, \beta \in \mathcal{A}$, the restrictions of functions f_α and f_β over $U_\alpha \cap U_\beta$ are equal. By allowing such a condition to hold, one extends the function from U_α to $U_\alpha \cup U_\beta$. Iterating the procedure for all indices $\alpha \in \mathcal{A}$ allows us to construct the global section $f : M \to V$.

Non-triviality of vector bundles: We note that the global section need not be a function $f : M \to \mathbb{R}^m$. If a global section is given by a function $f : M \to \mathbb{R}^m$, then the vector bundle $V \to M$ is a trivial bundle, i.e. $V \cong M \times \mathbb{R}^m$. Otherwise, we say that the vector bundle $V \to M$ is non-trivial. Such non-triviality of vector bundle implies that there are obstructions in naively gluing local sections $f_\alpha : U_\alpha \to \mathbb{R}^m$. These obstructions can be detected by computing topological invariants of vector bundles, such as their homology or cohomology groups. Loosely speaking, a global section of a non-trivial vector bundle can be thought of as of datasets whose measurements depend crucially on their underlying geometric space (or an underlying manifold M). Meanwhile, a global section of a trivial vector bundle can be considered as of datasets whose measurements can be well approximated by classical deep neural networks without making significant reference to underlying geometric space of the datasets. In particular, this observation hints that even if the measurements of 3D point clouds defined over an underlying manifold M take values in $\mathbb{R}^m$, we cannot guarantee that such measurements can be continuously extended to a function $f : M \to \mathbb{R}^m$.

Strength of deep learning algorithms: We note that there are variants of universal approximation theorem which regards approximating functions between differentiable manifolds with neural networks of arbitrary width and depth, see for example the work of Kratsios and Papon [26]. On the other hand, deep learning algorithms of finite depth and width may not have enough capabilities to approximate global sections of any vector bundle $V \to M$ of a manifold. In light of this vein, the main result of Park et al [33] (see for example Theorem 1.2, Sections 2 and 3) gives a presheaf/copresheaf theoretic characterization of capabilities of message passing neural networks: they approximate global sections of presheaf (and copresheaf) of continuous functions induced from direct sums of skyscraper cosheaves (and respectively sheaves). The obstructions of these presheaves (and copresheaves) from being sheaves (and cosheaves, respectively) can be utilized to explain limitations of a wide range of classes of deep learning algorithms, such as non-unique gluing of local sections of such presheaves (and copresheaves), and adversarial attacks.

Deep learning algorithms and vector bundles: Given that limitations of such deep learning algorithms originate from obstructions of a presheaf (and copresheaf) from being a sheaf (and respectively cosheaf), the paper hence calls for a need to design neural networks which are capable of approximating global sections of sheaves (or cosheaves) other than the aformentioned classes of presheaves and copresheaves. In particular, vector bundles over manifolds are examples of sheaves

over manifolds. A neural network which has capabilities to approximate global sections of vector bundles over manifolds hence has potential to overcome limitations that other previously studied deep learning algorithms may have experienced. We can ask a valid question on what isomorphism classes of vector bundles $V \to M$ of a manifold a given neural network can approximate their global sections. The capability that a neural network can effectively analyze dataset dependent on its intrinsic geometry can be rephrased as the capability to approximate global sections of non-trivial isomorphism classes of vector bundles over M. In turn, it implies that the neural network has capabilities in detecting exotic underlying geometric features of a given dataset, and can perform on par with or beyond other neural network algorithms.

A.2 Proof of Theorem 2.4

We first recall the definition of tautological bundle.

Definition A.1. Recall the homeomorphism of complex manifolds $S^2 \cong \mathbb{CP}^1$, which is the Grassmannian of complex projective lines in $\mathbb{C}^2$. The tautological bundle $\pi_t : T \to S^2$ is the vector bundle defined as follows: for each $x \in S^2$, the homeomorphism $S^2 \cong \mathbb{CP}^1$ identifies x with a point that represents a complex projective line $\ell \subset \mathbb{C}^2$. The fiber $\pi_t^{-1}(x)$ is given by the 1-dimensional complex vector space ℓ itself. We note that $\pi_t : T \to S^2$ can also be interpreted as a 2-dimensional real vector bundle by using the fact that $\mathbb{C}$ is a 2-dimensional $\mathbb{R}$-vector space.

The proof follows from applying Proposition 1.14 of Hatcher[23]. The isomorphism classes of n-dimensional real oriented vector bundles over S^2 is isomorphic to the fundamental group $\pi_1(SO(n))$. This group is isomorphic to $\pi_1(SO(3))$. To see this, we consider the short exact sequence of topological spaces obtained from fiber bundle of S^n,

$$SO(n) \to SO(n+1) \to S^n, \tag{8}$$

from which the long exact sequence of homotopy groups can be obtained using Theorem 4.41 of [22]:

$$\cdots \to \pi_2(S^n) \to \pi_1(SO(n)) \to \pi_1(SO(n+1)) \to \pi_1(S^n) \to \cdots. \tag{9}$$

The triviality of higher homotopy groups $\pi_2(S^n) = \pi_1(S^n) = 0$ for $n \geq 3$, see in particular Corollary 4.9 of [22], implies that we have $\pi_1(SO(n)) \cong \pi_1(SO(n+1))$ for all $n \geq 3$. By induction we have $\pi_1(SO(n)) \cong \pi_1(SO(3))$ for all $n \geq 3$. But using the homeomorphism $SO(3) \simeq \mathbb{RP}^3$, we obtain

$$\pi_1(SO(3)) \cong \pi_1(\mathbb{RP}^3) \cong \mathbb{Z}/2\mathbb{Z}. \tag{10}$$

Hence, every vector bundle over S^2 is either isomorphic to the trivial vector bundle $\mathbb{R}^n \times S^2 \to S^2$, or a non-trivial vector bundle of dimension n. We note that because $\pi_t : T \to S^2$ is a non-trivial real vector bundle of dimension 2. (As a complex line bundle, $L \otimes L$ gives the holomorphic tangent bundle of S^2.) Because $n \geq 3$, every non-trivial vector bundle of dimension n is isomorphic to the vector bundle obtained by taking direct sum of fibers of L with those of the trivial vector bundle $\mathbb{R}^{n-2} \times S^2 \to S^2$. This finishes the proof of the theorem.

A.3 Additional elaboration and proof of Theorem 2.6

We recall the following application of Proposition 1.4 to our setup from [23].

Proposition A.2. *Given a vector bundle $\pi : V'_Y \to Y'$ over a closed smooth manifold $Y' \subset \mathbb{R}^3$, there exists a vector bundle $E \to Y'$ such that $V'_Y \oplus E$ is the trivial vector bundle over Y'.*

The construction of the vector bundle E is obtained by finding a finite open cover $\{U_i\}_{i=1}^J$ of Y', over which the restriction of the vector bundle $V'_Y \to U_i$ is the trivial vector bundle $U_i \times \mathbb{R}^m \to U_i$ (often denoted as local trivialization of V'_Y). Using Urysohn's lemma, there exists a function $g_i : V'_Y \to \mathbb{R}^m$ which factors through the local trivialization $V'_Y \to U_i$. One can then define a map $g : V'_Y \to \mathbb{R}^{mJ} = \prod_{i=1}^J \mathbb{R}^m$ by defining $g := (g_1, g_2, \cdots, g_J)$. Then the map $(g, \pi) : V'_Y \to \mathbb{R}^{mJ} \times Y'$ defines the isomorphism of vector bundles. We note that the vector bundle $E \to Y'$ is of dimension $(J-1)m$. Because Y' is homotopy equivalent to a smooth orientable closed surface of genus g, there must be at least 2 open covers U_1, U_2, over each of which one obtains a trivialization of vector bundle $V'_Y \to U_i$.

In the case of constant masking function, we may consider the subspace $\mathbb{R} \subset \mathbb{R}^N$ spanned by the fixed choice of a vector $v \in \mathbb{R}^N$. Hence, constant masking function corresponds to replacing the

vector bundle V_X with $\mathbb{R} \times X$, because the range of the masking function lies in $\mathbb{R} \times X$ instead of $\mathbb{R}^N \times X$. This implies that if $\mathfrak{m} : X \to \mathbb{R}$ is a constant masking function, then the only possible way to embed a non-trivial vector bundle $V'_Y \to Y'$ into $Y' \times \mathbb{R} \to Y'$ is by establishing an isomorphism of vector bundles between $V'_Y \cong Y' \times \mathbb{R}$, a contradiction. This finishes the proof of Theorem 2.6.

We note that in the case where $\mathfrak{m} : X \to \mathbb{R}^N$ is a masking function that is not a section of a non-trivial vector bundle, the proof of the above proposition requires that the dimension of E has to be at least m. Therefore, unless $N \geq 2m$, the above method cannot be utilized to ensure that the non-trivial vector bundle $V'_Y \to Y'$ can be regarded as a vector sub-bundle of a trivial vector bundle of dimension N. Even if one enlarges the parameter N to be greater than $2m$, finding exactly what vector sub-bundle of the trivial bundle is the non-trivial bundle V'_Y is a challenging task. This is because the construction of the map (g, π) depends crucially on the choice of the open cover $\{U_i\}_{i=1}^J$ over which the non-trivial vector bundle V'_Y admits local trivialization. It would be a very difficult task for the neural network to predict such geometric data regarding open covers U_i, especially in cases where even the auxiliary space Y' parametrizing the point clouds is not understood correctly.

We also elaborate more on the relation between Theorem 2.6 and Figure 5. We thank anonymous reviewer for pointing out the need to clarify the relation between these two results. Let $f : Y' \to V'_Y$ be a non-trivial global section. We note that f shall be regarded as a restriction of a continuous function $F : \mathbb{R}^3 \to \mathbb{R}^{m_N}$ to Y'. Because Y' is compact, Heine-Cantor theorem implies that f is uniformly continuous. Hence, given any $\epsilon > 0$, there exists $\delta > 0$ such that for any $x, y \in Y'$ with $|x - y| < \delta$, we have $|f(x) - f(y)| < \epsilon$. Given that the learning objective of SIEVE is to optimize L_1 norm, we will denote by $|\cdot|$ the L_1 norm over $\mathbb{R}^3$ and $\mathbb{R}^{m_N}$, respectively.

Let Φ be the $\mathrm{SE}(3)$-equivariant neural network with constant masking function and scalar pipeline perturbed by Gaussian noise. Suppose Φ effectively approximates f. Then for every $x \in Y'$ we have $|\Phi(x) - f(x)| < \epsilon$. Fix a point $x \in Y'$. Using triangle inequality and uniform continuity of f, for every such $y \in Y'$ with $|x - y| < \delta$ we have

$$\epsilon > |\Phi(x) - f(x)| \geq |\Phi(x) - \Phi(y)| - |\Phi(y) - f(y)| - |f(y) - f(x)| \geq |\Phi(x) - \Phi(y)| - 2\epsilon. \quad (11)$$

Hence, we require that for every $y \in Y'$ such that $|x - y| < \delta$, we have $|\Phi(x) - \Phi(y)| < 3\epsilon$, i.e. Φ is also uniformly continuous. Since the masking function is constant, there exists a vector $v \in \mathbb{R}^m$ for some $m > 0$ such that

$$\Phi(x) = \mathrm{Decoder}(v, \mathrm{Encoder}([\mu_x, x])), \quad (12)$$

where μ_x denotes perturbation by Gaussian noise $\mathcal{N}(0, I)$, and the appropriate projection maps for encoder and decoder functions are abbreviated. Here, given $a \in \mathbb{R}^{n_1}$ and $b \in \mathbb{R}^{n_2}$, the notation $[a, b] \in \mathbb{R}^{n_1+n_2}$ denotes concatenation of vectors a and b. Both encoder and decoder functions from GATr are continuous. Hence, there exists some $\delta^* > 0$ such that

$$|[\mu_x, x] - [\mu_y, y]| < \delta^* \Rightarrow |\Phi(x) - \Phi(y)| < 3\epsilon. \quad (13)$$

From our assumption, we had $|x - y| < \delta$. Hence, by the definition of L_1 norm and triangle inequality, we have

$$\delta^* > |[\mu_x, x] - [\mu_y, y]| = |[\mu_x - \mu_y, x - y]| = |\mu_x - \mu_y| + |x - y|. \quad (14)$$

This implies that for any $y \in Y'$ such that $|x - y| < \delta$, we have $|\mu_x - \mu_y| < \delta^* - \delta$. However, because μ_x and μ_y are identically independently distributed, and there are arbitrarily many points lying in an open neighborhood of x of radius δ, the inequality $|\mu_x - \mu_y| < \delta^* - \delta$ cannot hold uniformly for all such $y \in Y'$ almost surely, unless δ^* is arbitrarily large. Hence, with such δ^* to be chosen to be arbitrarily large, in order for the inequality $|\Phi(x) - \Phi(y)| < 3\epsilon$ to be satisfied, we require that there exists some arbitrarily small $\epsilon'' > 0$ such that either $\sup_{x \in Y'} |\mathrm{Encoder}([\mu_x, x])| < \epsilon''$, or $\sup_{x \in Y'} |\Phi(x)| < \epsilon''$. Suppose the former holds. (We note that the latter case implies that f is a trivial function, contradicting our assumption on f). This implies that the domain of the decoder function (the last component of Φ) is contained in a compact subset corresponding to a ball of radius $\epsilon'' > 0$ from the constant vector $v \in \mathbb{R}^m$ that we chose earlier. Within this domain, the decoder function is uniformly continuous, hence one can choose $\epsilon > 0$ to be arbitrarily small so that the image of Φ is mapped to a ball of arbitrarily small radius centered at $\Phi(v)$. This explain the empirical collapse in Figure 5.

In relation to Theorem 2.6, the above argument shows that if Φ were to approximate the global section $f : Y' \to V'_Y$ effectively, then f would have been a constant function $\Phi(v)$ for some fixed vector v with arbitrarily small perturbations. We can then proceed as in the proof of Theorem 2.6 stated above to conclude that the only global sections of vector bundles such Φ can approximate (almost surely) are rank 1 trivial bundle of form $\mathbb{R} \times Y' \to Y'$.

A.4 Proof of Theorem 3.1

The theorem follows from considering the reformulation of the architectural components of SIEVE into the following steps as follows.

1. Using 3D point clouds, consider an auxiliary (possibly unknown) n-dimensional vector bundle $V_Y \to Y$' of some 3-dimensional closed orientable manifold Y' homeomorphic to $X \times I$ for some closed orientable surface $X \subset \mathbb{R}^3$ of genus g. We regard the n-dimensional measurements over 3D point clouds correspond as a global section $f : Y' \to V_Y$ of the n-dimensional vector bundle over Y'.
2. **(Encoder):** Find an m-dimensional vector bundle V_Y' $\to Y$' with $m \geq 2n + 5$ such that the fiber of V_Y injects into a vector subspace of the fiber of V_Y' at every $y \in Y$'.
3. Consider an auxiliary m-dimensional vector bundle $V_X \to X$ such that V_X and V'_Y are equivalent vector bundles with respect to homotopy equivalence $X \times I \to X$.
4. **(Spherical Masking):** Using the spherical projection $\pi_2 : X \to S^2$ which is the continuous map, replace the m-dimensional vector bundle $V_X \to X$ with the pullback of some m-dimensional vector bundle $V \to S^2$. This replacement operator is well defined if V_X is isomorphic to the pullback bundle $\pi_2^* V$ as vector bundles over X, which is why the statement of the theorem makes technical assumption on isomorphism class of vector bundles that SIEVE can approximate their global sections. By definition of spherical masking, V contains the dimension 2 tautological bundle $\pi_t : T \to S^2$ as its sub-vector bundle. Using classification of vector bundles over S^2, the vector bundle $V \to S^2$ must be isomorphic to either the trivial vector bundle or the unique non-trivial vector bundle of dimension m, whose fibers given by a summand of form $\mathbb{R}^{m-2} \oplus \pi_t^{-1}(x)$ for every $x \in S^2$.
5. The spherical projection transforms the n-dimensional measurements over 3D point clouds via the composition of maps

$$f^* : Y' \to V_Y \hookrightarrow V'_Y \cong V_X \cong \pi_2^* V \to \pi_2^* T. \tag{15}$$

   Here, the first map is our global section $f : Y' \to V_Y$. The second map is the inclusion of fibers of V_Y into V'_Y (i.e. the encoder). The third map originates from equivalence of vector bundles between X and Y'. The fourth and fifth map originates from replacing the vector bundle V_X with the pullback bundle of $V \to S^2$ using the spherical masking function. Note that the fifth map originates from the projecting the vector bundle $\pi_2^* V$ to its vector sub-bundle $\pi_2^* T$. (The previous step implies that $\pi_2^* V \cong \mathbb{R}^{m-2} \oplus \pi_2^* T$, so the fifth projection map can be explicitly define by mapping $\mathbb{R}^{m-2}$ to $\{0\}$).
6. **(Decoder):** Reconstruct the n-dimensional vector bundle $V_Y \to Y$' as a closed orientable $n + 2$ dimensional submanifold of $\mathbb{R}^{m+3}$, and approximate the global section $f : Y' \to V_Y$ using the spherical projection of measurements of 3D point cloud $f^* : Y' \to \pi_2^* T$.

The fact that spherical projection masking corresponds to replacing the vector bundle with the pullback of the non-trivial vector bundle over S^2 can be shown from the following reasoning. Using the definition of pullback of vector bundles and tautological bundles, the feature vectors obtained from the spherical masking function $\mathrm{SphericalMask}$ lie in the span of a linear subspace of $\mathbb{C}^2$ that passes through the mean $\overline{\mathbf{x}}$. Up to linear change of coordinates that sends $\overline{\mathbf{x}}$ to the point of origin $0 \in \mathbb{C}^1$, we obtain that the spherical masking function $\mathrm{SphericalMask}$ corresponds to defining a global section $g : X \to \pi_2^* T$, where for each $x \in X$, we have $g(x) = \mathrm{SphericalMask}(x)$. For general vector bundles $V_X \to X$, we can hence use $\mathrm{SphericalMask}$ to replace them with pullback of vector bundle $V \to S^2$ which contains $T \to S^2$ as a vector sub-bundle.

The requirement for $m \geq 2n + 5$ need not be satisfied, but we explain the motivation behind such a choice. By construction, we may assume that the vector bundle $V_Y \to Y$' is a smooth compact orientable manifold of dimension $n + 2$. By Haefliger-Hirsch, C.T.C. Wall, and Ghanwat-Pancholi's results, every such manifold embed into $\mathbb{R}^{2(n+2)-1} = \mathbb{R}^{2n+3}$. [19] [42] [17] Using the fact that V contains fibers isomorphic to the Euclidean space $\mathbb{R}^h$ of dimension at least $m - 2$, the condition on m ensures to identify the desired vector bundle V_Y as a smoothly embedded submanifold of the vector bundle $V \to S^2$. We note that the reconstruction of the desired vector bundle $V_Y \to Y$' can be obtained from applying the universal approximation theorem for approximating the desired continuous function $F : \mathbb{R}^{m+3} \to \mathbb{R}^{m+3}$ whose image is homeomorphic to the vector bundle V_Y.

The approximation of the global section $f : Y' \to V_Y$ using $f^* : Y' \to \pi_2^* T$ follows from using universal approximation theorem for $\mathbb{R}^m$. Without loss of generality, we assume V_Y is a non-trivial vector bundle which are isomorphic to pullback of a non-trivial vector bundle of S^2. The decoder satisfying conditions of universal approximation theorem for approximating continuous functions $f : \mathbb{R}^m \to \mathbb{R}^n$ can be used to approximate global sections of trivial vector bundle $g : Y' \to \mathbb{R}^{n-2}$ by approximating local restrictions of g to an open affine chart $\{U_\alpha\}_{\alpha \in \mathcal{A}}$ of Y' and gluing such local restrictions of g without any obstructions. Because the vector bundle V_Y is isomorphic to $\mathbb{R}^{n-2} \oplus \pi_2^* T$, we can use the product of functions $(g, f^*) : Y' \to \mathbb{R}^{n-2} \oplus \pi_2^* T \cong V_Y$ to approximate any global section of V_Y.

A.5 Theoretical Limitations

Our theoretical results suggest that SIEVE is capable of approximating global sections of a wide range of classes of non-trivial vector bundles over manifolds. Nevertheless, as similar to other state-of-the-art techniques, SIEVE has some theoretical limitations, in that it does not guarantee outperformance in analyzing datasets which are global sections of vector bundles which do not contain pullback of tautological bundles as vector sub-bundle. For instance, there may be many non-trivial isomorphism classes of vector bundles over manifolds which do not originate from pullback of non-trivial vector bundles over S^2. Further studies would be required to devise new neural networks which are capable of approximating such global sections of nontrivial vector bundles.

B Implementation Details

B.1 Pre-Training

Model Parameters: The backbone of pre-trained SIEVE is GATr, a model based on multivector representations [5]. We embed the position of each node as a trivector inside the multivector.

The encoder comprises a 9-layer GATr with 16 multivector and 64 scalar channels. The input is 1 multivector and 1 scalar channel and the output is 1 multivector and 32 scalar channel. The 32-dimensional scalar output constitutes the SIEVE embedding s_{code}. The encoder has 949,825 parameters.

The decoder is a 3-layer GATr with 16 multivector and 64 scalar channels. The input is 1 multivector and 1 scalar channel and the output is a single-channel multivector and scalar output for coordinate reconstruction. The decoder has 318,690 parameters.

Note that for variants of SIEVE trained with ShapeNet [8], the model parameters from Table 4 are used.

Training Method: The model is trained for 10 epochs using the AdamW optimizer with a batch size of 200. To accelarate training, bf16 quantization is used. We employ a cosine learning rate schedule with a warm-up phase, reaching a maximum learning rate of 3×10^{-4}. To stabilize training, we apply gradient clipping with a maximum norm of 10. We train on a single Nvidia B200 GPU over 4 days. Memory consumption with a batch size of 16 is 154 GB.

Additional Techniques: Three distinct pre-training strategies have been tested on the smaller ShapeNet dataset [8].

The first strategy utilizes simple position reconstruction as outlined in the main text, with pre-training performed on a volumetric ShapeNet dataset. The standard deviation of the spherical projection mask noise is $\sigma = 0.02$.

The second strategy is to reconstruct the normal vector and the curvature in addition to the position of the node. The normal vector is the output multivector's vector, and the curvature is the output scalar. The reconstruction loss for this strategy is

$$\mathcal{L} = L1(\text{PredPos}, \text{Pos}) + 10 \cdot (1 - \cos(\text{PredNorm} \cdot \text{Norm}) + 5 \cdot L1(\text{PredCurv}, \text{Curv}). \quad (16)$$

We use the surface ShapeNet dataset for this strategy, as neither curvature and normal vectors can be defined in the interior.

The third strategy involves the application of a nearest-neighbor attention mask within the decoder. To ensure the encoder extracts global embeddings, the decoder is forced to reconstruct the nodes

without local information by masking 70% of the nearest nodes during self-attention. This strategy is implemented on a volumetric ShapeNet dataset.

As discussed in Appendix C.3 and Table 6, the third strategy of using a nearest-neighbor attention mask yielded the best results. Therefore, the final OpenShape [30] pretrained SIEVE utilizes a nearest neighbor attention mask with ratio 0.7 along with a spherical projection mask.

B.2 Synthetic Data Experiment

Model Parameters: The mask ablation study (Section 4.2) evaluates two different equivariant backbone architectures: GATr [5] and SE(3)-Transformer [16]. As detailed in Table 4, both models utilize matching hidden channels and layer numbers to ensure a fair comparison. The positions are embedded as trivectors for GATr and as type-1 vectors for the SE(3)-Transformer. For the SE(3)-Transformer, we generate edges between the nodes using k-nearest neighbor with $k = 50$.

Table 4: Model Hyperparameters for the Mask Ablation Study

Module	Hyperparameter	GATr	SE(3)-Transformer
Encoder	Input Channel (s,v)	(1,1)	(1,1)
	Hidden Channel (s,v)	(32,8)	(32,8)
	Output Channel (s,v)	(16,1)	(16,1)
	Number of Attention Heads	4	1
	Number of Layers	6	6
Decoder	Hidden Channel (s,v)	(32,8)	(32,8)
	Output Channel (s,v)	(1,1)	(1,1)
	Number of Attention Heads	4	1
	Number of Layers	2	2

Mask Details: The constant mask is a single vector or multivector token that is applied uniformly over all of the nodes. This token is trainable. The noise mask is a uniform sample of points on an ellipsoid surface with semi-axes $(2, 1, 0.5)$. The ellipsoid is centered at each point cloud's centroid, and the orientation is determined by PCA of the original point cloud: x-axis for the largest variance direction, y-axis for the middle, and z-axis for the smallest. This ensures that the noise mask is also an equivariant operator. The spherical projection mask is as described in Section 3.1, but no noise is added to the projected position.

Training Method: Both models are trained with a simple $\mathcal{L}1$ loss on predicted node positions. The models are trained over 200 epochs, with a batch size of 4. We employ a cosine learning rate decay that reaches a maximum learning rate of 3×10^{-4} after a 10% warm up period. To ensure numerical stability, the gradients are clipped at a threshold of 10.0. No weight decay is applied.

B.3 Downstream Tasks

Each downstream task utilizes a domain specific architecture. For the 3D-EPN point cloud completion task, we utilize PCN [51]; for the point cloud registration task on ModelNet40, we utilize DCP [44]. All downstream models use the hyperparameters specified in their original publications and official implementations. The only modification made is to the input channel to accommodate the additional 16-dimension s_{code} embeddings. The SIEVE embeddings are concatenated along with the position as the input. The SIEVE encoder is frozen when used in a downstream task; we do not fine-tune it. The RNA segmentation task and the geometric registration task uses the OpenShape pretrained encoder, and the geometric registration task uses the ShapeNet pretrained encoder with the "simple noise" setting.

The added downstream task training time is a single forward pass through the frozen SIEVE encoder. On the RNA segmentation task with PointNet++, memory increases from 2134 MiB to 2166 MiB, and per-epoch time from 92s to 123s (+31s). On 3D-EPN with PCN, per-epoch time increases from 4 min 56s to 5 min 43 sec (+47s). The absolute overhead is comparable in both cases, so its relative share falls as the downstream backbone becomes more expensive: 34% for PointNet++ against 16% for PCN.

C Ablation Study

C.1 Reconstruction of Point Clouds with Identical Topology

To demonstrate that the topological obstruction is the cause of the representational collapse, we train SIEVE to reconstruct a point cloud dataset with identical topology. We use DeepJEB, which contains 2138 meshes of jet engine brackets [25]. These engine brackets all have two holes at the top and 4 holes at the bottom; therefore all of the meshes have the same topology.



(a) Reconstruction Validation Loss

(b) Reconstruction Result

Figure 6: **SIEVE on DeepJEB.** DeepJEB is a point cloud dataset that contains objects with identical topology. Therefore, both the constant mask SIEVE and the spherical mask SIEVE is able to able to do near-perfect reconstruction.

The results (Figure 6) show that both constant mask SIEVE and spherical mask SIEVE are able to reconstruct point clouds in the DeepJEB dataset. Hence we confirm that it is the topological obstruction that is causing the representational collapse in Section 4.2, not an inherent architectural limitation.

C.2 Scalar Input of Encoder

Our choice to use Gaussian noise as the scalar input of the equivariant encoder is based on its capability and ease of use. This subsection is an ablation study on the scalar input of the encoder to justify this selection.

We compare noise to two other scalar encoder inputs: a constant array of ones $\mathbf{1} = [1, 1, \cdots, 1]^T$ and a handpicked invariant scalar feature of the point cloud, for which we chose surface variance (SV). Given a node $\mathbf{x}_i$, and its 16 closest neighbors $\mathcal{N}(\mathbf{x}_i)$, the covariance matrix of the 16 neighbors' relative position $\mathbf{p}_{ij} = \mathbf{x}_j - \mathbf{x}_i, \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$ is defined as K. Denoting the three eigenvalue of K as $\lambda_1 \geq \lambda_2 \geq \lambda_3$, sv is defined as $\text{sv} = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3)$.

Three distinct SIEVE models were trained with different encoder inputs. The ModelNet40 dataset was used to pre-train all of the models, and the backbone model for SIEVE was GATr. Training was conducted over 100 epochs with a batch size of 32, using a cosine learning rate decay with a maximum learning rate of 3×10^{-4}.

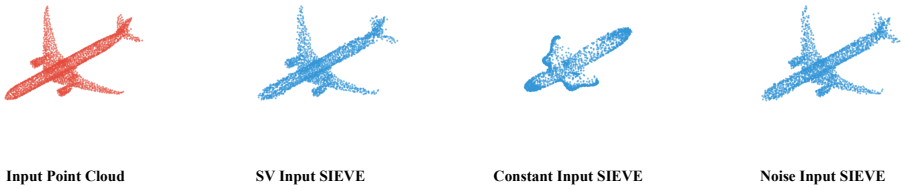

Figure 7: **Input Noise Ablation.** We train SIEVE on ModelNet40 with three different input: local point cloud feature sv, a constant array of ones, and noise. Either noise or sv input enables SIEVE to reconstruct the input point cloud. SIEVE with constant input shows significantly worse performance when reconstructing the input point cloud.

Figure 7 showcases the reconstruction results of SIEVE with different encoder inputs. Both sv and noise are shown to be valid inputs; SIEVE is capable of reconstructing the input point cloud nearly perfectly with either. The constant array input is shown to be suboptimal. While it does not yield a complete collapse, it demonstrates difficulties in reconstructing the geometric details of the point cloud.

We choose noise over sv because it does not require pre-processing of the input point cloud. Relying on handpicked SE(3)-invariant point cloud features increases the difficulty of implementing SIEVE across a wide range of datasets.

It is interesting to note that the use of noise as input does not lead to the scalar input being completely ignored. Figure 8 showcases the result when SIEVE trained with noise as input receives sv as input instead. The reconstruction is distorted, which would not be the case if the scalar input was simply ignored by the model. We therefore conclude that the encoder utilizes the specific distribution of the scalar input to construct the embedding, further highlighting the importance of the input signal even when it is stochastic.



Figure 8: **Noise Input Trained SIEVE with SV as Input.** The above figure demonstrates the result *(blue, right)* of reconstructing the input point cloud *(red, left)* using a noise-input pre-trained SIEVE with SV as scalar input. The distorted reconstruction implies that the encoder does not simply ignore the scalar noise input.

The use of noise as an encoder input does not lead to SIEVE simply memorizing a specific noise realization for a given point cloud. Table 5 provides the results when using different random seeds to generate the noise input for the encoder. While each seed yields a unique input noise for the encoder, these differences have almost no effect on SIEVE's ability to reconstruct the point cloud. The three different seeds yield nearly identical validation losses, demonstrating that the model can consistently generate embeddings over various samples of noise.

Table 5: Reconstruction Validation Loss ($\mathcal{L}1$) of SIEVE with different random seeds on the ModelNet40 dataset

	SEED 42	SEED 67	SEED 721
VALIDATION LOSS	1.004×10^{-2}	1.002×10^{-2}	0.9981×10^{-2}

C.3 SIEVE Embeddings do not Collapse to the Radius

In this subsection, we rigorously analyze the embeddings to demonstrate that they do not collapse into trivial representations of the radius $r = \|\mathbf{x}_i - \overline{\mathbf{x}}\|$. We compare SIEVE using a simple spherical projection mask without any noise to a variant with noise added to the spherical projection. We refer to this variant as 'simple noise'.

Additionally, we compare two further variations of SIEVE. The first reconstructs the normal vector and the curvature in addition to the position. We refer to this variant as 'curv and norm'. The second applies an additional self-attention mask to the decoder, where for every node the nearest 70% of nodes are masked during self-attention. We refer to this variant as 'nearest neighbor attention mask'. The models are pre-trained with the settings outlined in Appendix B.1.

We first analyze the embeddings using principal component analysis (PCA). Given an input point cloud and its SIEVE embedding $s_{code} \in \mathbb{R}^{N_p \times 16}$, the covariance of the embedding is $K = \frac{1}{N_p - 1}[s_{code} - \text{mean}(s_{code})]^T[s_{code} - \text{mean}(s_{code})] \in \mathbb{R}^{16 \times 16}$. Since K is a positive semi-

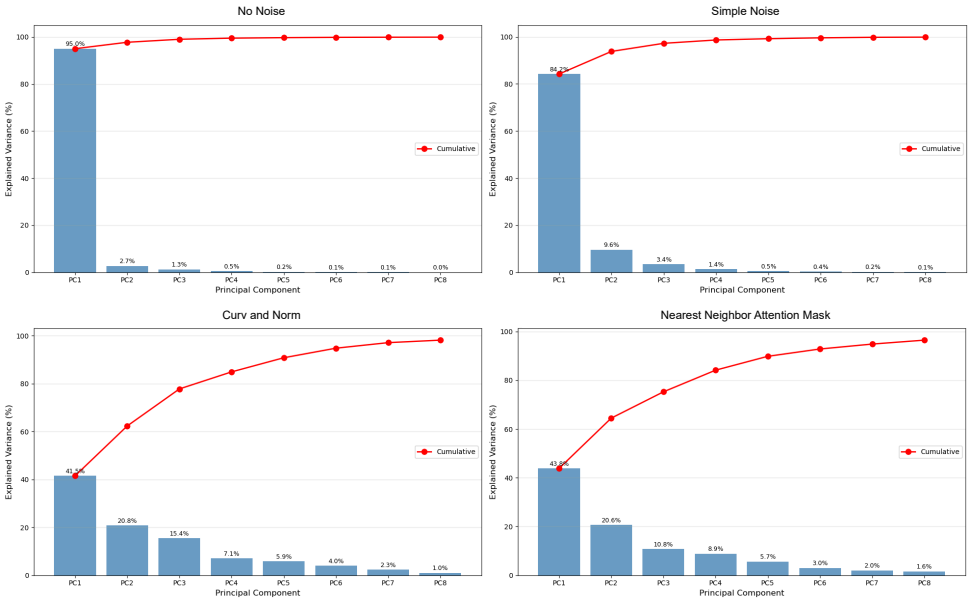

Figure 9: **SIEVE Embedding PCA Analysis.** We perform PCA on the SIEVE embedding s_{code} and plot the percentage of each eigenvalue λ_i over the sum of all the eigenvalues. In addition, the cumulative percentage is also plotted. From the top left, we plot the analysis results for SIEVE with no noise, 'simple noise' strategy, 'curv and norm' strategy, and 'nearest neighbor attention mask' strategy.

definite matrix, all the eigenvalues are non-negative. We sort the eigenvalues in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{16}$, and calculate the ratio of each eigenvalue relative to the total variance $\frac{\lambda_i}{\sum_{i=1}^{16} \lambda_i}$.

PCA analysis (Figure 9) showcases that not adding any noise to the spherical projection mask yields a dimensional collapse, with 95% of the variation contained in the first principal component. The addition of a simple Gaussian noise prevents this collapse, decreasing the first principal component's variation percentage to 84.2%, and increasing the second principal component's variation percentage to 9.6%. Furthermore, using more difficult reconstruction strategies such as 'curv and norm' and 'nearest neighbor attention mask' mitigates the dimensional collapse even further, with both strategies showing variation being spread out over multiple principal components.

Next, we directly compare the learned embeddings to the radial distance $r = \|\mathbf{x}_i - \overline{\mathbf{x}}\|$. We calculate the Pearson correlation coefficient between each of the 16 channels of s_{code} and the radius r for a representative point cloud sample.

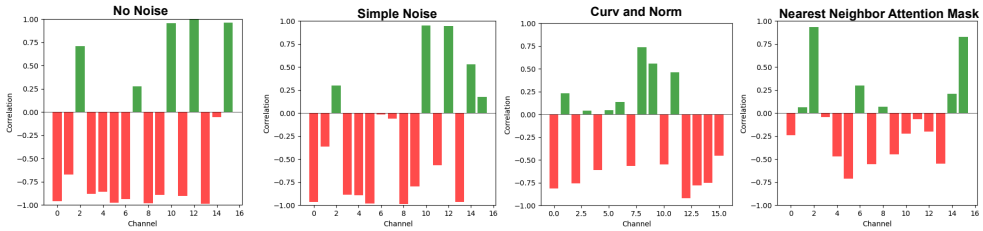

Figure 10: **Embedding Radius Correlation.** This plot illustrates the correlation between each channel of the SIEVE embedding and the radius $r = \|\mathbf{x}_i - \overline{\mathbf{x}}\|$.

The correlation analysis (Figure 10) demonstrates that when no noise is added to the masking process, the embeddings collapse and correlate heavily with the radius r. In this configuration, 12 out of 16 channels exhibit an absolute correlation value above 0.75. The addition of simple noise reduces this

count to 9, indicating that even basic noise helps prevent a complete dimensional collapse to the radial component. The more sophisticated reconstruction strategies further decouple the embeddings from the radius; notably, the 'nearest neighbor attention mask' variant results in only 2 channels possessing an absolute correlation value above 0.75.

Finally, we apply each strategy's embedding to the RNA segmentation downstream task with PointNet++ to evaluate the marginal utility of each variant compared to a raw radial feature. The results are in table 6.

Table 6: Test accuracies(%) on the RNA segmentation task with different SIEVE strategies

xyz ONLY	RADIUS r	NO NOISE	SIMPLE NOISE	CURV AND NORM	NN ATTENTION MASK
71.55	75.54	76.08	75.94	78.05	84.99

The SIEVE embedding pre-trained without noise yields results nearly identical to those using the radius r as a feature, confirming the dimensional collapse identified in our PCA analysis. While the addition of simple noise disrupts this radial dependency, it initially results in a marginal performance decrease. In contrast, the more sophisticated strategies yield significantly higher accuracies. This indicates that these strategies have learned richer features beyond the simple radius r.

C.4 SIEVE Encoder Needs to be Trained

Since SIEVE uses SE(3)-equivariant neural networks, even a randomly initialized SIEVE encoder provides an SE(3)-invariant embedding to the downstream task model. To isolate the contribution of the pretraining, we compare the downstream task performance of a randomly initialized SIEVE encoder against a pre-trained one. The setup reproduces Table 1 exactly: same encoder architecture, s_{code} integration and hyperparameters. The only difference are the weights loaded into the SIEVE encoder. In both arms, s_{code} is passed to the downstream network exactly as the encoder emits it, with no rescaling or normalization.The results are below.

Table 7: Comparison of untrained SIEVE and pre-trained SIEVE on the RNA segmentation task (%)

	PointNet++	DeltaNet	DiffusionNet
xyz as feature	71.55	87.21	81.20
random init SIEVE	67.47	77.59	82.52
pre-trained SIEVE	86.28	89.58	89.30

Adding a randomly initialized SIEVE encoder to the downstream task does not meaningfully improve the performance. DiffusionNet's accuracy increases marginally (+1.32), but both DeltaNet (-9.62) and PointNet++ (-4.08) suffer performance loss with a randomly initialized SIEVE encoder. Across all three backbones, neither the equivariant architecture nor the additional input width accounts for the gain in Table 1; those gains are attributable to the pretraining itself.

C.5 SIEVE Embedding Injection Location

The SIEVE encoder is frozen downstream, and every downstream task uses the original published hyperparameters; the only modification is widening the input layer. The choice that does matter is where the embedding is injected. We have done an ablation study on the geometric registration task where the ShapeNet pre-trained SIEVE embeddings are injected at different locations.

We have found that concatenating at the start yields the best results. Therefore, all of the downstream tasks concatenate s_{code} at the downstream network input.

Table 8: s_{code} injection location comparison on the geometric registration task

s_{code} injection location	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
Before DGCNN (chosen)	7.05	2.65	1.39	0.000004	0.0020	0.00092
DGCNN Intermediate	7.34	2.71	1.55	0.000006	0.0025	0.0015
Before Transformer	12.43	3.53	2.01	0.000029	0.0054	0.0036
Before SVD	24.59	4.96	3.12	0.000452	0.021	0.015

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have claimed to have found a theoretical limitation (Section 2), proposed a solution (Section 3), and have experimentally demonstrated the theoretical predictions (Section 4.2) and the claimed utility of the embeddings (Section 4.3).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have thoroughly analyzed the properties of our embeddings in Appendix. C, and have discussed the limitations of our theoretical analysis in A.5. Furthermore, we honestly discuss the mixed results for point cloud registration, acknowledging that for in distribution test samples, the gain in performance is not substantial.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We give motivation for why vector bundles are relevant mathematical concepts to analyze strenghts of deep learning algorithms (as proposed in Section 2) in Appendix A. The theoretical results (Theorem 2.4, 2.6, and 3.1) are proven in Appendix A.2, A.3, and A.4. We also give a brief discussion of theoretical limitations of SIEVE in Appendix A.5.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We briefly mention the training details in Section 4, and provide further details in Appendix B.3.

   Guidelines:

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

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We provide the code via anonymous github for submission, and have attached a README for easy execution.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: These details are provided in Appendix B.3.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: We do not report error bars. Pretraining a single SIEVE encoder on ShapeNet is the dominant cost in our pipeline, and repeating pretraining across multiple seeds for every ablation and downstream evaluation was not feasible within our compute budget. We partially address seed sensitivity in Table 5, where reconstruction with three different random seeds yields similar validation losses. Downstream architectures (PointNet++, DGCNN, PCN, DCP) use their authors' original hyperparameters without modification, so the remaining unmeasured variance is downstream-training variance characteristic of those well-studied baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: These details are provided in Appendix B.1

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research is on point cloud self-supervised learning, and all the datasets were obtained legally and with consent from the authors.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This research showcases a way to create point cloud embeddings. While point cloud deep learning has numerous applications, our research is not specialed to a specific application, and hence we cannot discuss the direct social impact of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We used verified, open source datasets such as ShapeNet and ModelNet40. Furthermore, our research has no specific application; it is intended to benefit general point cloud learning.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All of the reference methods and datasets have been cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide the code with a thorough README via anonymous github for submission.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research has not used crowdsourcing, nor used human subjects in our research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: This research has not used human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development did not involve LLMs as an important or non-standard component.

    Guidelines:

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