VBA: Vector Bundle Attention for Intrinsically Geometric Representation Learning

Shenglei Fang [1] **Xianfang Sun** [2] **You Zhou** [1]

Abstract

Learning from geometrically structured data is central to applications in biology, physics, and computer vision. In many tasks, meaningful comparisons depend on how features are aligned in space. Graph Neural Networks capture local structure but are constrained by message passing. Transformers model long-range dependencies but largely ignore geometry. We introduce the Vector Bundle Attention Transformer (VBA-Transformer), a framework that redefines attention as an intrinsic geometric operator. Each token couples a base manifold coordinate with a fiber feature vector, following vector bundle theory. A principled parallel transport mechanism aligns fiber features across local coordinate systems before similarity is computed. This embeds geometry directly into the attention operator. Unlike prior methods that inject geometry as an external bias or positional encoding, VBA integrates geometry natively inside attention. On challenging single-cell RNA sequencing benchmarks, VBA achieves state-of-the-art accuracy, outperforming Transformer baselines by over 3–5%. On spatial transcriptomics, it demonstrates superior clustering performance. On 3D point clouds, it achieves competitive accuracy, validating broad generalization across domains. Beyond empirical gains, we provide theoretical analysis of invariance and perturbation stability. We also demonstrate robust transport behavior empirically. Together, these results establish intrinsic geometric alignment as a powerful principle for scalable representation learning. Our code is available at: https://github.com/yzlab1/Vector-Bundle-Attention

[1]Department of Infection and Immunity, School of Medicine, Cardiff University, Cardiff, UK [2]School of Computer Science and Informatics, Cardiff University, Cardiff, UK. Correspondence to: You Zhou <zhouy58@cardiff.ac.uk>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

Many real-world datasets are geometric in nature, where meaningful comparisons depend not only on feature similarity but also on how features are aligned in space. Intuitively, two features should be considered similar only after they are expressed in a common local coordinate system defined by the underlying geometry. Learning representations from such non-Euclidean geometric data is therefore a central challenge in machine learning (Bronstein et al., 2017; 2021). Such data is ubiquitous, ranging from molecular graphs and 3D point clouds to social networks and biological tissues. However, existing paradigms often struggle to capture these intrinsic geometries effectively. While Graph Neural Networks (GNNs) and Transformers represent the dominant paradigms for learning on structured data, both are constrained by inherent limitations. GNNs (Scarselli et al., 2008; Defferrard et al., 2016; Velickovic et al., 2017) capture local structure via message passing, their reliance on fixed adjacency matrices limits modeling long-range dependencies and latent manifolds (Xu et al., 2018; Maron et al., 2018; Zhou et al., 2020; Zheng et al., 2024). In contrast, Transformers (Vaswani et al., 2017) excel at modeling complex dependencies but remain fundamentally "geometry-blind," treating data as fully-connected graphs without intrinsic structural awareness (Ying et al., 2021; Yuan et al., 2025).

Recent studies have attempted to bridge this gap. Some inject geometric information as additive biases in attention, while others employ Riemannian or Hyperbolic geometry to embed data into curved spaces (Ying et al., 2021; Chen et al., 2022; 2024; Yang et al., 2024; Nickel & Kiela, 2017). While effective, these approaches only weakly integrate content and geometry (Yuan et al., 2025). Geometry is treated as an external constraint or fixed prior, rather than being integrated directly into the similarity operator itself. Consequently, attention weights remain largely geometry-agnostic, and geometric consistency is not enforced during similarity computation, limiting robustness and interpretability in complex geometric domains. This raises a central question: **Can we redefine attention itself as an intrinsic geometric operator within a learned manifold?**

We answer this challenge with the **Vector Bundle Attention Transformer (VBA-Transformer)**. Vector bundles

naturally model situations where each data point has both a global position (where it lies on a manifold) and a local feature space (how information is represented at that point). This matches many learning problems in which geometry describes relationships between points, while features encode rich local content. Our core insight is that a truly powerful geometric model should not passively accept a predefined structure, but should instead simultaneously learn the underlying manifold of the data and a metric for measuring affinity within that manifold. To achieve this, we turn to the mathematical framework of vector bundles from differential geometry(Nakahara, 2018; Bamberger et al., 2024; Liu & Su, 2015). This framework models data points as residing in a vector bundle. Each point is associated with a low-dimensional base coordinate capturing global spatial arrangement, together with a high-dimensional fiber space encoding its local feature representation. Crucially, within this learned vector bundle, we define attention by first projecting features into local fiber spaces and then aligning them across points using learned parallel transport induced by the geometric coordinates. This alignment ensures that similarity is computed in a geometry-consistent space rather than in an arbitrary ambient embedding. This ensures that features are compared in a common local reference frame rather than in incompatible coordinate systems. Consequently, the attention score is no longer a measure of feature similarity in an arbitrary ambient space, but a learnable metric of affinity within the learned vector bundle space itself. This fundamentally redefines attention as an intrinsic geometric operator, tightly coupling representation learning with manifold discovery.

We instantiate our approach as an end-to-end autoencoder and validate the VBA-Transformer on diverse benchmarks. Specifically, we focus on biological data with complex intrinsic geometries: spatial transcriptomics (ST) datasets, which capture gene expression while preserving spatial location (Heydari & Sindi, 2023), and single-cell RNA sequencing (scRNA-seq) data, where cells form an implicit manifold in high-dimensional gene expression space (Moon et al., 2018; Xu et al., 2023). Beyond biological domains, we validate VBA's universality through evaluation on ModelNet40, the canonical 3D point cloud classification task.

Our contributions are summarized as follows:

1.Vector Bundle Attention (VBA): We introduce a theoretically grounded attention mechanism that operates intrinsically on a learned geometric manifold.

2.VBA-Transformer: We present a new architecture that effectively disentangles geometry from features, endowing Transformers with native geometric awareness.

3.Theory: We establish invariance properties and perturbation bounds that formally explain the stability and robustness of the proposed geometric attention mechanism.

4.Empirical Validation: We achieve state-of-the-art performance on scRNA-seq, highly competitive results on spatial transcriptomics, and demonstrate broad generality on 3D point-cloud classification.

2. Related Works

Graph Neural Networks (GNNs). Graph Neural Networks (GNNs) form the classical approach for learning on structured data (Scarselli et al., 2008; Defferrard et al., 2016; Kipf & Welling, 2017; Velickovic et al., 2017). And most GNNs fall into Message Passing Neural Networks (MPNNs) (Gilmer et al., 2017), where nodes exchange messages with their immediate neighbors and update their representations by aggregating these local features. While inherently geometric, they are tightly coupled to a predefined adjacency structure, which often limits their ability to capture long-range dependencies or recover the underlying manifold (Xu et al., 2018; Zhou et al., 2020) and leading to issues like over-smoothing and limited expressivity (Li et al., 2018; Oono & Suzuki, 2019; Xu et al., 2018; Morris et al., 2019). To address this issue, some research has been conducted into graph structure learning, where the adjacency is adaptively inferred from data(Zheng et al., 2024; Zhao et al., 2023; Franceschi et al., 2019). While these methods improve flexibility, they remain within the message-passing paradigm. This makes them less suitable for disentangling geometry from content.

Transformers on Graphs and Geometric Data. Transformers (Vaswani et al., 2017) have recently been adapted to structured domains, thanks to their ability to model long-range interactions. Early works such as Graph-BERT treated graphs as sequences, losing much of the structural information (Zhang et al., 2020). Subsequent models such as Graphormer, GPS and GBT(Ying et al., 2021; Rampášek et al., 2022; Venkat et al., 2023) incorporate centrality and spatial encodings into the attention mechanism, while approaches for 3D point clouds(Guo et al., 2021; Chen et al., 2025) directly encode geometric relationships as attention biases. Although these methods incorporate geometric priors, geometry is typically introduced only as an auxiliary bias on top of standard Euclidean dot products, leaving the core similarity computation insensitive to coordinate changes and local frames. In contrast, we explicitly align features across local coordinate systems before interaction, turning attention from a bias-corrected heuristic into an intrinsic geometric operator, where geometry defines similarity itself rather than serving as a post-hoc correction.

Riemannian and Manifold Representation Learning. A parallel direction seeks to endow models with stronger geo-

metric foundations by embedding data into non-Euclidean spaces. Notable approaches, such as Hyperbolic and Riemannian neural networks (Nickel & Kiela, 2017; Chami et al., 2019; Sala et al., 2018; Feng et al., 2019; Liu et al., 2019; de Ocáriz Borde et al., 2023), redefine representation learning in spaces with constant negative or positive curvature to better model hierarchies or cycles. Recent advancements have extended these principles to Transformer architectures, for instance, Hypformer (Yang et al., 2024) explores efficient attention mechanisms directly within hyperbolic space. Beyond constant curvature settings, other works focus on enforcing geometric symmetries through invariant and equivariant networks (Maron et al., 2018). Bundle-based methods further propose to model data using fiber structures over a base manifold (Bamberger et al., 2024). However, they generally treat geometry as an embedding space or external bias, failing to integrate it as an intrinsic operator within the attention mechanism itself.

Our Position. Unlike prior approaches that introduce geometry through external positional encodings or apply geometric corrections after attention, VBA-Transformer embeds geometry inside the attention mechanism itself. Each token is represented as a point on a learned base manifold with an attached fiber vector, and key/value features are first aligned by an isometric transport before any similarity is computed in the fiber space. This "transport-then-attend" formulation differs fundamentally from positional-bias Transformers and from message-passing or gauge-based models because geometry defines the similarity operator itself rather than modulating it post-hoc.

3. Method

We propose **Vector Bundle Attention (VBA)**, a Transformer attention operator defined *intrinsically* on a learned vector bundle over a learned base manifold. Rather than treating geometry as an auxiliary bias, VBA rewrites the elementary attention operations (projection, similarity, aggregation) to operate directly on fiber-valued features. A learnable *connection* performs parallel transport to align features across local coordinate systems before similarity is computed. This yields an attention operator that couples *content* and *geometry* at the operator level, enabling the model to learn non-flat geometries (via a curvature proxy) and to flexibly trade off efficiency and expressivity via bundle-mixing and low-rank/residual parameterizations. The overall architecture of the VBA is shown in Figure 1.

As illustrated in Figure 2, unlike standard Transformers that treat features as residing in a flat global space, VBA acknowledges the manifold structure of data. We introduce an **Align-then-Compare** paradigm: features are parallel transported to a common fiber before any interaction, ensuring that all similarity computations are geometrically consistent. Intuitively, VBA treats attention as comparing features only after they are expressed in a common local coordinate system, just as tangent vectors must be aligned before being compared on a manifold.

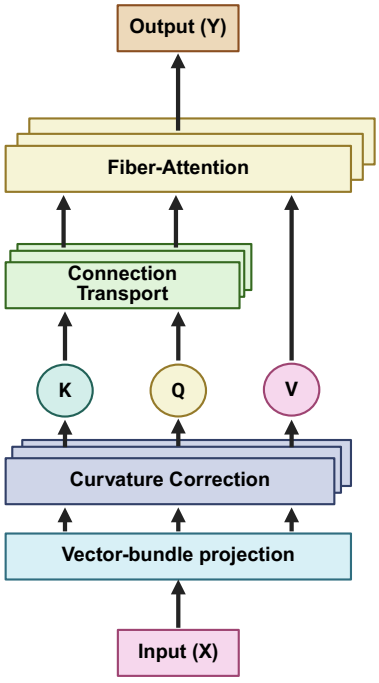

Figure 1. Overview of Vector Bundle Attention (VBA).

3.1. Vector Bundle Attention

The key idea is simple: before comparing two features, we first transport one into the local coordinate system of the other so that similarity is geometrically meaningful. Given an input sequence of ambient features $X = \{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^D$, a VBA layer performs the following conceptual steps, which are detailed below and summarized in Algorithm 1. Conceptually, this mirrors comparing tangent vectors on a manifold: features must first be transported into a common local frame before meaningful similarity can be evaluated.

1. Projection: Each input x_i is projected into a disentangled representation consisting of a coordinate on a latent base manifold, $b_i \in \mathbb{R}^{d_b}$, and a feature vector in a local fiber space, $f_i \in \mathbb{R}^{d_f}$.

2. Curvature Correction: The fiber vector is optionally modulated by a learnable curvature term derived from the connection field, allowing the model to capture non-flat geometries.

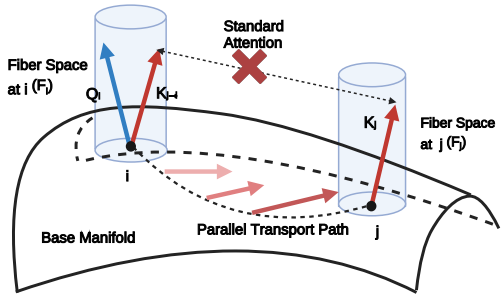

Figure 2. Conceptual illustration of the attention mechanism in VBA.

3. Parallel Transport: For any pair of points (i, j), a learnable connection defines a parallel transport operator $T_{j\to i}$ that moves the fiber vector at point j to the fiber space at point i.

4. Fiber Attention: Attention is then computed in the fiber space, using the transported vectors to calculate similarity and aggregate values.

Projection to the Vector Bundle. We first realize the vector bundle decomposition via linear projections. Each input token x_i is mapped to a base coordinate b_i and a set of M candidate fiber vectors:

$$b_i = W_b x_i \in \mathbb{R}^{d_b}, \qquad f_i^{(m)} = W_f^{(m)} x_i \in \mathbb{R}^{d_f}. \tag{1}$$

where $m = 1, \ldots, M$

To allow for flexible representations, we use a data-dependent gating mechanism to dynamically combine these candidate fibers. A small bundle-selector network $s : \mathbb{R}^D \to \mathbb{R}^M$ produces mixing weights $\alpha_i = softmax(s(x_i))$. These are used to produce the final fiber representation $\bar{f}_i$, which is passed to the subsequent stages:

$$\bar{f}_i = \text{LayerNorm}\left(\sigma(g(x_i)) \odot \phi\left(\sum_{m=1}^{M} \alpha_i^{(m)} A^{(m)} f_i^{(m)}\right)\right), \tag{2}$$

where $g(\cdot)$ is a linear projection, σ is a sigmoid gate, ϕ is a GELU non-linearity. Crucially, the matrices $\{A^{(m)}\}_{m=1}^{M}$ function as learnable, bundle-specific feature transformations. This multi-bundle design allows the model to process input features along M parallel pathways, learning specialized refinements for different aspects of the data before they are adaptively combined. This design choice ensures that the fiber refinement is not an arbitrary feature transformation, but a geometrically-informed process that is directly coupled to the parameters of the underlying vector bundle structure.

Curvature Correction. To provide the model with the most principled geometric information, we compute a high-fidelity approximation of the formal curvature 2-form, $\Omega = d\Gamma + \Gamma \wedge \Gamma$. This is made possible by our learnable connection field, implemented as a neural network `ConnectionNet` which maps any base coordinate b to the connection coefficient matrices $\{\Gamma_k(b)\}$.

The two components of the curvature tensor are computed as follows:

- The derivative term, $d\Gamma$, is calculated by taking the Jacobian of the `ConnectionNet`'s output with respect to the input coordinates b, which is computed efficiently via automatic differentiation.
- The algebraic term, $\Gamma \wedge \Gamma$, is calculated using the commutator (Lie bracket) of the output connection matrices $[\Gamma_1(b), \Gamma_2(b)]$.

These two terms are then combined to form a position-dependent curvature tensor $\Omega(b)$. This tensor provides a rich, local description of the learned manifold's geometry, which can then be used to modulate the fiber representations. This approach moves beyond simple proxies and directly integrates a core component of differential geometry into the network's architecture.

The Learnable Connection and Parallel Transport. A central innovation of our work is an endpoint-conditioned isometric transport mechanism which is orthogonal, length-preserving, serving as a practical surrogate for formal path-dependent parallel transport. We design a `TransportNet` module that learns an orthogonal transport operator $T_{j\to i} \in SO(d_f)$ for each pair of points. This enforces isometry, ensuring that the transport operation is a pure rotation that preserves the length of the feature vectors.

To achieve this, the `TransportNet` uses a lightweight MLP to predict a generator matrix $S \in \mathbb{R}^{d_f \times d_f}$ from the base coordinates (b_i, b_j). This matrix is then forced to be skew-symmetric:

$$S_{\text{skew}} = \frac{1}{2}(S - S^\top). \tag{3}$$

The final transport operator is the matrix exponential of this skew-symmetric matrix, which is guaranteed by construction to be a special orthogonal matrix (a rotation):

$$T_{j\to i} = \exp(S_{\text{skew}}). \tag{4}$$

This formulation enables the model to learn an endpoint-dependent and geometrically principled transport mechanism directly from data.

Fiber Attention. With the transport operator defined, we first project the (optionally curvature-corrected) fiber vectors

$\bar{f}_i$ into queries, keys, and values: $Q_i = W_q \bar{f}_i$, $K_j = W_k \bar{f}_j$, $V_j = W_v \bar{f}_j$. To compute the attention score from query i to key j, we first transport K_j and V_j to the fiber space at i:

$$\tilde{K}_{j\to i} = T_{j\to i}\, K_j, \qquad \tilde{V}_{j\to i} = T_{j\to i}\, V_j. \tag{5}$$

The attention logits and weights are then computed using the transported key:

$$e_{ij} = \frac{\langle Q_i,\, \tilde{K}_{j\to i}\rangle}{\sqrt{d_f}}, \qquad \alpha_{ij} = softmax_j(e_{ij}). \tag{6}$$

The final output fiber is a weighted sum of the transported value vectors:

$$y_i^{\text{fiber}} = \sum_j \alpha_{ij}\, \tilde{V}_{j\to i}. \tag{7}$$

This output is then projected back to the ambient space $\mathbb{R}^D$ with a final linear layer W_o: $y_i = x_i + W_o y_i^{\text{fiber}}$

3.2. Model Architecture and Implementation

The VBA block is designed as a drop-in replacement for standard self-attention within a Pre-LayerNorm (Pre-LN) Transformer architecture:

$$X' = X + \mathrm{VBA}(\mathrm{LN}(X)), \quad X'' = X' + \mathrm{FFN}(\mathrm{LN}(X')). \tag{8}$$

This modular design is made practical, stable, and efficient through several key choices:

Stable Initialization: The transport operator T is initialized near the identity matrix to ensure stable training dynamics from the start.

Expressivity: Soft bundle mixing and data-dependent gating are employed to efficiently enhance the model's representational power.

Efficiency: For large-scale inputs, computational cost is managed via low-rank parameterizations of the transport matrices or by restricting attention to local windows.

These features ensure that VBA is a robust operator that embeds a powerful geometric inductive bias at the core of the Transformer architecture.

4. Experiment

In this work, our proposed VBA-Transformer is designed to capture complex intrinsic geometric structures by disentangling geometry from content, making it especially suitable for data characterized by rich and often non-Euclidean relationships. We test this hypothesis by benchmarking our model against state-of-the-art baselines across three distinct domains: 1. Spatial Transcriptomics (ST), where data possesses an explicit spatial, near-Euclidean geometry. 2. Single-Cell RNA Sequencing (scRNA-seq), where the geometric relationships are implicit and must be learned from a high-dimensional manifold. 3. 3D Point Cloud Classification, a canonical benchmark for geometric deep learning that tests generalization to non-biological, explicit 3D structures.

4.1. Experiments on spatial transcriptomics

The explicit geometric structure of spatial transcriptomics (ST) data presents a unique challenge and opportunity for representation learning (Heydari & Sindi, 2023). We first apply our model to identify spatial domains, a fundamental unsupervised learning task in the field. Success in this task demonstrates a model's ability to effectively integrate gene expression data with spatial information to uncover meaningful biological patterns in tissue.

Data. We evaluate the performance of our model on the widely-used human dorsolateral prefrontal cortex (DLPFC) dataset from the Lieber Institute for Brain Development (LIBD) (Maynard et al., 2021). This benchmark dataset contains 12 spatially-resolved transcriptomics slices, each with expert-annotated ground-truth labels corresponding to the four or six distinct cortical layers and white matter (WM). Its well-organized, multi-layered tissue architecture provides an ideal testbed for evaluating a model's capacity to learn representations that preserve spatially coherent biological structures. In line with standard practice, we log-normalize the raw gene expression counts and use the top 3,000 highly variable genes (HVGs) as input features. To demonstrate our model is not tailored to DLPFC only, we additionally evaluate on the 10x Genomics human breast cancer Visium dataset (Cui et al., 2025).

Experimental Settings. The task is to perform unsupervised spatial clustering to recover the annotated anatomical layers. We benchmark our VBA-Transformer model for spatial transcriptomics (VBA-ST) against six state-of-the-art baselines: BayesSpace(Zhao et al., 2021a), SpaGCN(Hu et al., 2021), DeepST(Xu et al., 2022), GraphST(Long et al., 2023), BASS(Li & Zhou, 2022), and DiffusionST(Cui et al., 2025). Baselines were executed using their officially suggested hyperparameters for a fair comparison. We quantify clustering accuracy using three standard metrics: the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) are reported in the main paper as the primary performance metric; for a more comprehensive analysis,Completeness scores is provided in the Appendix B.3.

Result. As summarized in Table 1, our comprehensive evaluation using both Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) demonstrates the robustness of VBA-ST. Across the 12 DLPFC samples, VBA-ST achieves the best or tied-for-best ARI on 4 samples (151507,

Table 1. Comparison of ARI and NMI metrics on 12 human DLPFC samples and the Breast Cancer dataset. The best performance per sample (row) is shown in **bold**.

Sample	BayesSpace		SpaGCN		DeepST		GraphST		BASS		DiffusionST		VBA-ST	
	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
151507	0.47	0.59	0.46	0.54	0.50	0.62	**0.51**	**0.65**	0.50	0.63	0.43	0.64	**0.51**	0.64
151508	0.44	0.56	0.43	0.52	0.39	0.54	0.35	0.57	0.46	0.57	0.46	0.59	**0.47**	**0.60**
151509	0.40	0.55	0.47	0.62	0.36	0.57	0.41	0.59	0.51	0.62	0.54	0.62	**0.61**	**0.64**
151510	0.38	0.52	0.46	0.54	0.38	0.55	0.41	**0.61**	0.50	0.58	**0.51**	**0.61**	0.50	0.60
151669	**0.47**	0.59	0.37	0.52	0.35	0.51	0.27	0.50	0.35	0.52	0.38	0.58	0.42	**0.60**
151670	0.43	0.52	0.38	0.49	0.34	0.48	0.25	0.46	0.38	0.51	**0.46**	**0.57**	0.41	0.56
151671	0.49	0.62	0.56	0.68	0.52	0.69	0.50	0.67	0.56	0.71	0.58	0.72	**0.65**	**0.73**
151672	0.44	0.58	0.57	0.68	0.50	0.67	**0.58**	**0.69**	0.56	0.65	0.55	0.66	0.57	0.67
151673	**0.55**	0.69	0.52	0.51	**0.55**	0.68	0.51	0.66	0.53	0.72	0.54	**0.73**	0.53	0.66
151674	0.32	0.48	0.39	0.53	0.49	0.63	0.48	0.61	0.51	0.62	**0.52**	**0.69**	0.46	0.58
151675	0.53	0.69	0.47	0.58	**0.57**	0.72	0.52	0.64	0.55	**0.74**	0.46	0.61	0.47	0.59
151676	0.37	0.53	0.33	0.48	0.52	0.59	0.42	0.56	**0.53**	**0.70**	0.51	0.68	0.37	0.53
Average	0.44	0.58	0.45	0.56	0.46	0.60	0.43	0.60	0.50	0.63	**0.50**	**0.64**	**0.50**	0.62
Breast Cancer	0.49	0.53	0.57	0.61	0.51	0.52	0.53	0.54	0.55	0.54	0.57	0.54	**0.59**	**0.69**

151508, 151509, 151671). Notably, on samples 151509 and 151671, VBA-ST shows a substantial margin over the next best model (+0.07 in ARI), highlighting its strong ability to capture spatially coherent biological structures. On average, VBA-ST achieves a top-tier ARI of 0.50, consistently outperforming GCN-based approaches such as SpaGCN (0.45) and GraphST (0.43).

Furthermore, on the heterogeneous Breast Cancer dataset, VBA-ST achieves state-of-the-art performance with an ARI of **0.59** and an NMI of **0.69**. Most strikingly, in terms of NMI, VBA-ST outperforms the strongest baseline (DiffusionST) by a significant margin of **0.15**. This result reinforces our hypothesis: while iterative local aggregation in methods like SpaGCN and GraphST can be prone to over-smoothing, VBA-ST's ability to directly model global context and intricate geometric relationships allows it to learn more expressive representations, yielding superior clustering consistency across diverse tissue types

4.2. Experiments on scRNA-seq

Our next set of experiments aims to explore VBA-Transformer's generalization capabilities in a biological context where the geometric structure is more abstract and latent. To this end, we focus on the task of cell type annotation in single-cell RNA sequencing (scRNA-seq). It is widely hypothesized that distinct cell types form a complex, low-dimensional manifold within the high-dimensional transcriptomic space (Moon et al., 2018). Accurate cell type annotation is therefore not merely a classification problem, but fundamentally a task of geometric structure discovery. This is precisely the challenge VBA-Transformer is designed to address. Through its vector bundle attention mechanism, our model explicitly learns an underlying base manifold to capture the intrinsic, non-Euclidean relationships between cells, rather than operating on them as disconnected points in a high-dimensional space. We therefore use this task to validate VBA-Transformer's core capabilities in implicit geometry discovery and effective representation learning.

Data. **Peripheral blood mononuclear cell (PBMC)**: We used the Zheng68K dataset, sequenced using the 10X Chromium platform, to train and evaluate the model for cell type annotation (Zheng et al., 2017). This large-scale scRNA-seq dataset contains 65,943 cells spanning 11 cell populations and includes expression profiles for 20,387 genes. The dataset was randomly split into 70% for training, 15% for validation, and 15% for testing. **Pancreatic datasets**: Pancreatic datasets were obtained from five studies: Baron (GSE84133)(Baron et al., 2016), Muraro (GSE85241)(Muraro et al., 2016), Xin (GSE81608)(Xin et al., 2016), Segerstolpe (E-MTAB-5061)(Segerstolpe et al., 2016), and Lawlor (GSE86473) (Lawlor et al., 2017). Among these, the Baron and Muraro datasets were used for training, while the remaining three datasets were used for testing. The training set contains 10,600 cells spanning 15 cell types, and the test set includes 4,218 cells representing 11 cell populations. **The human cell landscape (HCL) dataset** (Han et al., 2020) comprises 599,926 cells across 59 tissues. This data is used for pretraining.

Experimental Settings. We benchmarked the VBA-Transformer model for single cell RNA sequencing (VBA-SC) against seven widely adopted single-cell models. Among them scGPT (Cui et al., 2024), scBERT (Yang et al., 2022), Geneformer(Theodoris et al., 2023) and

TOSICA (Chen et al., 2023b) are transformer-based single-cell models. Seurat (Hao et al., 2021) and singleR (Aran et al., 2019) are correlation-based models. scNym (Kimmel & Kelley, 2021) is a recently proposed semi-supervised learning method. We evaluate annotation performance using two primary metrics: overall accuracy (OA) and the macro F1-score (F1). Accuracy measures the proportion of correctly labeled cells, providing a general assessment of performance. Crucially, the macro F1-score is included to evaluate the model's ability to identify all cell types equally, giving more weight to the correct classification of rare cell populations, which is a key challenge in scRNA-seq analysis.

Result. The quantitative results for cell type annotation, presented in Table 2, provide a two-fold validation of our model's effectiveness. First, we evaluated the standard VBA-SC model trained from scratch. Even without the benefit of large-scale pre-training, VBA-SC demonstrates clear architectural superiority over other Transformer-based models. This is particularly evident on the challenging PBMC dataset, where it outperforms pre-trained methods such as scBERT and scGPT by a notable margin of over 3% in accuracy. This result highlights the power and data-efficiency of our model's geometric inductive bias. Next, to create a direct comparison with self-supervised methods, we introduced VBA-SC (SSL), which incorporates a pre-training stage. With the ability to first learn a general-purpose representation of the cellular manifold, our model's performance was substantially boosted, achieving new state-of-the-art results on both datasets. On the Pancreas dataset, VBA-SC (SSL) surpasses the strong Seurat baseline, reaching 97.64% accuracy. On the PBMC dataset, it further extends its lead over all competitors, achieving 80.68% accuracy.

This two-tiered success is particularly significant. The strong performance of the base model validates the power of our architecture, while the state-of-the-art results of the pre-trained version show that this superior architecture also effectively leverages self-supervised learning. This confirms that the core mechanism of VBA, identifying and leveraging the intrinsic geometric manifold of the data, is the main driver of its performance, enabling more accurate discrimination of fine-grained cell types and setting a new benchmark for this task.

4.3. Experiments on point cloud

The core thesis of our work is that VBA-Transformer is a generalizable model for geometric learning, whose principles are not confined to any single domain. To substantiate this claim, we must benchmark it on a canonical geometric task outside of bioinformatics to demonstrate its true universality. To this end, we apply our model to the task of 3D point cloud classification, a canonical benchmark in the field of geometric deep learning and a direct test of a model's capacity for geometric representation learning. Unlike the bioinformatics tasks, the goal here is to infer a global object category from the 3D coordinates of the points. This experiment evaluates whether VBA's core mechanism, vector bundle attention, can effectively learn a global representation to distinguish complex 3D shapes.

Table 2. Performance comparison of VBA-SC with baseline methods for cell type annotation

	PBMC		Pancreas	
Model	**OA**	**F1**	**OA**	**F1**
scBERT	75.52%	0.61	93.59%	0.85
TOSICA	73.65%	0.59	92.37%	0.84
scGPT	75.47%	0.61	93.10%	0.85
Geneformer	74.67%	0.60	92.72%	0.84
scNym	69.50%	0.53	89.72%	0.80
singleR	68.40%	0.50	91.82%	0.84
Seurat	54.50%	0.37	96.37%	0.89
VBA-SC	**78.71%**	**0.63**	**93.89%**	**0.86**
VBA-SC (SSL)	**80.68%**	**0.65**	**97.64%**	**0.89**

Table 3. Performance and efficiency comparison on the ModelNet40 benchmark.

Model	**Param.**	**FLOPs**	**OA**	**mAcc**
PointNet	3.5M	0.4G	89.2%	86.2%
PointNet++	1.5M	1.7G	92.5%	89.7%
DGCNN	1.8M	2.4G	92.7%	90.4%
PCT	2.9M	2.3G	93.2%	90.0%
GTNet	2.1M	4.3G	93.2%	92.6%
PointTransformer	17.1M	9.1G	93.7%	90.6%
DTNet	6.4M	4.4G	93.4%	90.9%
PointGA	1.7M	2.0G	93.8%	90.9%
VBA-P	7.0M	12.2G	92.9%	90.3%
VBA-P-Tiny	1.7M	2.9G	90.1%	87.2%

Data. We evaluated our model on the canonical 3D point cloud classification task using the ModelNet40 benchmark (Wu et al., 2015). This dataset contains 12,311 clean 3D CAD models across 40 categories, such as airplane, sofa, plant, and desk. We follow the official split, with 9,843 models for training and 2,468 for testing. For each model, we uniformly sample 1,024 points from the object surface and use only their 3D coordinates as input, providing a direct test of geometric feature learning.

Experimental Settings. This experiment serves as a crucial test of VBA-Transformer's universality as a foundational geometric learning model. Our aim is not to outperform highly-specialized, fine-tuned architectures, but to demonstrate that our model's core principles can achieve

Table 4. Ablation study of VBA-Transformer across three distinct datasets.

	Point cloud		PBMC		Pancreas	
Model	**OA**	**mAcc**	**OA**	**F1**	**OA**	**F1**
VBA-T	**92.9%**	**90.3%**	**78.71%**	**0.63**	**93.89%**	**0.86**
Transformer	82.6%	78.2%	66.76%	0.44	88.92%	0.71
GBT	85.8%	81.7%	70.32%	0.56	90.33%	0.83
MQA-T	83.1%	78.9%	67.73%	0.48	87.33%	0.67
GQA-T	83.6%	79.1%	66.57%	0.43	87.19%	0.66

strong, competitive results on a canonical task far outside its primary application domains. We compare VBA-Transformer against a suite of representative point cloud models, from the pioneering PointNet(Qi et al., 2017a) and PointNet++(Qi et al., 2017b) to modern graph-based (DGCNN)(Wang et al., 2019) and attention-based methods(Guo et al., 2021; Zhao et al., 2021b; Han et al., 2022; Zhou et al., 2024; Chen et al., 2025). For a fair comparison, we adopt standard data augmentation and training procedures for ModelNet40. We report two key metrics: Overall Accuracy (OA) for a top-level performance summary, and mean-class Accuracy (mAcc) to ensure performance is robust and not biased towards majority object classes.

Result. As presented in Table 3, the results strongly support our central thesis on the universality of the VBA-Transformer. The goal was not to establish a new state-of-the-art, but to validate that our foundational architecture could generalize to a completely different domain. Our VBA-Transformer for point cloud (VBA-P) model's robust 92.9% OA confirms its effectiveness. More strikingly, the compact VBA-P-Tiny remains competitive with classic methods while operating at a scale comparable to highly efficient models, such as PointNet++. The significance of these findings lies not in breaking records but in the demonstration of true architectural versatility. The ability of VBA-Transformer, without domain-specific fine-tuning, to achieve such competent performance on this canonical benchmark validates its principles as a genuinely general-purpose framework for geometric learning.

Additionally, we also evaluated the rotation robustness of VBA-P, following SO(3) evaluation protocols, and evaluate VBA-P with ScanObjectNN, as shown in Appendix B.5

4.4. Ablation Study

To quantify the contribution of the core components in VBA-Transformer, we conducted a comprehensive ablation study on the Point Cloud, PBMC, and Pancreas datasets. We systematically replaced the proposed Vector Bundle Attention with alternative attention mechanisms and report the results in Table 4.

First, we confirm the importance of incorporating geometric information. The geometry-agnostic Transformer baseline exhibits consistent performance degradation across all datasets compared to geometry-aware models. For instance, on the Point Cloud benchmark, its overall accuracy is nearly 3.2% lower than that of the geometry-biased Transformer (GBT)(Venkat et al., 2023).

More importantly, VBA-T consistently outperforms GBT, which injects geometry only through additive bias terms. On the Point Cloud benchmark, VBA-T achieves a 7.1% higher absolute accuracy improvement over GBT, and on PBMC the gap exceeds 8%. This indicates that explicitly modeling geometry through vector bundle structure and feature alignment provides substantially stronger representations than bias-based geometric corrections.

We further compare VBA-T with efficient attention variants, including Multi-Query Attention (MQA-T)(Shazeer, 2019) and Grouped-Query Attention (GQA-T)(Ainslie et al., 2023). Although these variants reduce computational cost, their accuracy is consistently lower than VBA-T across all datasets. This suggests that the observed performance gains do not arise from attention scaling or architectural efficiency tricks.

Overall, these ablations confirm that performance improvements stem specifically from intrinsic geometric alignment and transport within the vector bundle formulation, rather than from parameter scaling, attention variants, or additive geometric bias mechanisms.

Additional ablation details are shown in Appendix B.7.

5. Conclusion

We introduce the VBA Transformer, a principled yet practical framework that treats attention as an operator on a learned geometric space. Its core, Vector Bundle Attention, disentangles base manifold geometry from fiber features. We enforce endpoint conditioned isometric transport to align fibers before similarity, embedding this geometric property directly in attention. A curvature informed correction from a learnable connection field captures nonflat manifold structure. Across domains, it is effective: state of the art on single cell RNA sequencing, competitive in spatial tran-

scriptomics, and strong on 3D point clouds. Overall, the design pairs a hard isometry constraint with flexible, data driven parameterization, gaining geometric bias without requiring expensive path-dependent transport integration. These results support geometric disentanglement and further narrow the gap between deep learning and differential geometry. More broadly, VBA demonstrates that treating attention as a geometric operator enables models to reason more faithfully over structured data without relying on handcrafted priors, providing a principled foundation for future geometric Transformers and scalable geometric foundation models.

Acknowledgements

We thank the reviewers and area chair for their constructive feedback. This work was supported by the Cardiff University Digital Transformation Innovation Institute Seedcorn Funding Scheme. We acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR). Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023] (McIntosh-Smith et al., 2024). We also acknowledge Advanced Research Computing at Cardiff (ARCCA) for providing supercomputing resources and technical support that contributed to this work.

Impact Statement

This paper presents a geometric deep learning framework for representation learning on structured and non-Euclidean data. Potential positive impacts include improved analysis and integration of complex biological, biomedical, and multimodal scientific datasets, including applications in single-cell genomics, spatial transcriptomics, and other data-intensive areas relevant to precision medicine. By enabling more effective incorporation of geometric structure into deep learning, the proposed framework may support advances in AI-driven biomedical discovery, patient stratification, and computational approaches for translational research. As with other machine learning approaches, careful validation, interpretability, and responsible deployment are important to mitigate risks related to bias, robustness, and overreliance on automated predictions in real-world biomedical settings.

References

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebron, F., and Sanghai, S. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023.

Aran, D., Looney, A. P., Liu, L., Wu, E., Fong, V., Hsu, A., Chak, S., Naikawadi, R. P., Wolters, P. J., Abate, A. R., et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology*, 20(2):163–172, 2019.

Bamberger, J., Barbero, F., Dong, X., and Bronstein, M. M. Bundle neural networks for message diffusion on graphs. *arXiv preprint arXiv:2405.15540*, 2024.

Baron, M., Veres, A., Wolock, S. L., Faust, A. L., Gaujoux, R., Vetere, A., Ryu, J. H., Wagner, B. K., Shen-Orr, S. S., Klein, A. M., et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

Brehmer, J., De Haan, P., Behrends, S., and Cohen, T. S. Geometric algebra transformer. *Advances in Neural Information Processing Systems*, 36:35472–35496, 2023.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, 2017.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Chami, I., Ying, Z., Ré, C., and Leskovec, J. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

Chen, C., Wu, Y., Dai, Q., Zhou, H.-Y., Xu, M., Yang, S., Han, X., and Yu, Y. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Chen, D., O'Bray, L., and Borgwardt, K. Structure-aware transformer for graph representation learning. In *International conference on machine learning*, pp. 3469–3489. PMLR, 2022.

Chen, G., Wang, M., Yang, Y., Yu, K., Yuan, L., and Yue, Y. Pointgpt: Auto-regressively generative pre-training from point clouds. *Advances in Neural Information Processing Systems*, 36:29667–29679, 2023a.

Chen, J., Xu, H., Tao, W., Chen, Z., Zhao, Y., and Han, J.-D. J. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1):223, 2023b.

Chen, S., Fang, Z., Wan, S., Zhou, T., Chen, C., Wang, M., and Li, Q. Geometrically aware transformer for point cloud analysis. *Scientific Reports*, 15(1):16545, 2025.

Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., and Wang, B. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, 2024.

Cui, Y., Cui, Y., Wang, R., Zhu, Z., Zeng, X., Nakai, K., Cui, F., Zhang, Z., Shi, H., Chen, Y., et al. Diffusionst: a deep generative diffusion model-based framework for enhancing spatial transcriptomics data quality and identifying spatial domains. *Briefings in Bioinformatics*, 26(4): bbaf390, 2025.

de Ocáriz Borde, H. S., Kazi, A., Barbero, F., and Lio, P. Latent graph inference using product manifolds. In *The eleventh international conference on learning representations*, 2023.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Dong, R., Qi, Z., Zhang, L., Zhang, J., Sun, J., Ge, Z., Yi, L., and Ma, K. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022.

Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019.

Franceschi, L., Niepert, M., Pontil, M., and He, X. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pp. 1972–1982. PMLR, 2019.

Fuchs, F., Worrall, D., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., and Hu, S.-M. Pct: Point cloud transformer. *Computational visual media*, 7(2):187–199, 2021.

Han, X., Zhou, Z., Fei, L., Sun, H., Wang, R., Chen, Y., Chen, H., Wang, J., Tang, H., Ge, W., et al. Construction of a human cell landscape at single-cell level. *Nature*, 581(7808):303–309, 2020.

Han, X.-F., Jin, Y.-F., Cheng, H.-X., and Xiao, G.-Q. Dual transformer for point cloud analysis. *IEEE Transactions on Multimedia*, 25:5638–5648, 2022.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zager, M., et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

Heydari, A. A. and Sindi, S. S. Deep learning in spatial transcriptomics: Learning from the next next-generation sequencing. *Biophysics Reviews*, 4(1), 2023.

Hu, J., Li, X., Coleman, K., Schroeder, A., Ma, N., Irwin, D. J., Lee, E. B., Shinohara, R. T., and Li, M. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods*, 18(11): 1342–1351, 2021.

Kimmel, J. C. and Kelley, D. R. Semisupervised adversarial neural networks for single-cell classification. *Genome research*, 31(10):1781–1793, 2021.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lawlor, N., George, J., Bolisetty, M., Kursawe, R., Sun, L., Sivakamasundari, V., Kycia, I., Robson, P., and Stitzel, M. L. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type–specific expression changes in type 2 diabetes. *Genome research*, 27(2): 208–222, 2017.

Lehmann, D., Spanholtz, J., Osl, M., Tordoir, M., Lipnik, K., Bilban, M., Schlechta, B., Dolstra, H., and Hofer, E. Ex vivo generated natural killer cells acquire typical natural killer receptors and display a cytotoxic gene expression profile similar to peripheral blood natural killer cells. *Stem Cells and Development*, 21(16):2926–2938, 2012. doi: 10.1089/scd.2011.0659. URL https://journals.sagepub.com/doi/abs/10.1089/scd.2011.0659.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Li, Z. and Zhou, X. Bass: multi-scale and multi-sample analysis enables accurate cell type clustering and spatial domain detection in spatial transcriptomic studies. *Genome biology*, 23(1):168, 2022.

Liang, D., Zhou, X., Xu, W., Zhu, X., Zou, Z., Ye, X., Tan, X., and Bai, X. Pointmamba: A simple state space model for point cloud analysis. *Advances in neural information processing systems*, 37:32653–32677, 2024.

Liu, Q., Nickel, M., and Kiela, D. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Liu, R. and Su, Z. Feature extraction via vector bundle learning. In *Mathematical Problems in Data Science: Theoretical and Practical Methods*, pp. 143–157. Springer, 2015.

Long, Y., Ang, K. S., Li, M., Chong, K. L. K., Sethi, R., Zhong, C., Xu, H., Ong, Z., Sachaphibulkij, K., Chen, A., et al. Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with graphst. *Nature Communications*, 14(1):1155, 2023.

Lu, C., Liu, Q., Wang, C., Huang, Z., Lin, P., and He, L. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1052–1060, 2019.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Maynard, K. R., Collado-Torres, L., Weber, L. M., Uytingco, C., Barry, B. K., Williams, S. R., Catallini, J. L., Tran, M. N., Besich, Z., Tippani, M., et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature neuroscience*, 24(3):425–436, 2021.

McIntosh-Smith, S., Alam, S., and Woods, C. Isambard-ai: a leadership-class supercomputer optimised specifically for artificial intelligence. In *Proceedings of the Cray User Group*, pp. 44–54. 2024.

Moon, K. R., Stanley III, J. S., Burkhardt, D., van Dijk, D., Wolf, G., and Krishnaswamy, S. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Muraro, M. J., Dharmadhikari, G., Grün, D., Groen, N., Dielen, T., Jansen, E., Van Gurp, L., Engelse, M. A., Carlotti, F., De Koning, E. J., et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394, 2016.

Nakahara, M. *Geometry, topology and physics*. CRC press, 2018.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Sala, F., De Sa, C., Gu, A., and Ré, C. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pp. 4460–4469. PMLR, 2018.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Segerstolpe, Å., Palasantza, A., Eliasson, P., Andersson, E.-M., Andréasson, A.-C., Sun, X., Picelli, S., Sabirsh, A., Clausen, M., Bjursell, M. K., et al. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell metabolism*, 24(4):593–607, 2016.

Shazeer, N. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Su, Z., Welling, M., Pietikäinen, M., and Liu, L. Svnet: Where so (3) equivariance meets binarization on point cloud representation. In *2022 International Conference on 3D Vision (3DV)*, pp. 547–556. IEEE, 2022.

Theodoris, C. V., Xiao, L., Chopra, A., Chaffin, M. D., Al Sayed, Z. R., Hill, M. C., Mantineo, H., Brydon, E. M., Zeng, Z., Liu, X. S., et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T., and Yeung, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. Graph attention networks. *stat*, 1050 (20):10–48550, 2017.

Venkat, N., Agarwal, M., Singh, M., and Tulsiani, S. Geometry-biased transformers for novel view synthesis. *arXiv preprint arXiv:2301.04650*, 2023.

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

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Xin, Y., Kim, J., Okamoto, H., Ni, M., Wei, Y., Adler, C., Murphy, A. J., Yancopoulos, G. D., Lin, C., and Gromada, J. Rna sequencing of single human islet cells reveals type 2 diabetes genes. *Cell metabolism*, 24(4):608–615, 2016.

Xu, C., Jin, X., Wei, S., Wang, P., Luo, M., Xu, Z., Yang, W., Cai, Y., Xiao, L., Lin, X., et al. Deepst: identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Research*, 50(22):e131–e131, 2022.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Xu, Y., Zang, Z., Xia, J., Tan, C., Geng, Y., and Li, S. Z. Structure-preserving visualization for single-cell rna-seq profiles using deep manifold transformation with batch-correction. *Communications Biology*, 6(1):369, 2023.

Yang, F., Wang, W., Wang, F., Fang, Y., Tang, D., Huang, J., Lu, H., and Yao, J. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

Yang, M., Verma, H., Zhang, D. C., Liu, J., King, I., and Ying, R. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3770–3781, 2024.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Yuan, C., Zhao, K., Kuruoglu, E. E., Wang, L., Xu, T., Huang, W., Zhao, D., Cheng, H., and Rong, Y. A survey of graph transformers: Architectures, theories and applications. *arXiv preprint arXiv:2502.16533*, 2025.

Zhang, J., Zhang, H., Xia, C., and Sun, L. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

Zhao, E., Stone, M. R., Ren, X., Guenthoer, J., Smythe, K. S., Pulliam, T., Williams, S. R., Uytingco, C. R., Taylor, S. E., Nghiem, P., et al. Spatial transcriptomics at subspot resolution with bayesspace. *Nature biotechnology*, 39(11):1375–1384, 2021a.

Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16259–16268, October 2021b.

Zhao, W., Wu, Q., Yang, C., and Yan, J. Graphglow: Universal and generalizable structure learning for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3525–3536, 2023.

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

Zheng, Y., Zhang, Z., Wang, Z., Li, X., Luan, S., Peng, X., and Chen, L. Rethinking structure learning for graph neural networks. *arXiv preprint arXiv:2411.07672*, 2024.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

Zhou, W., Wang, Q., Jin, W., Shi, X., and He, Y. Graph transformer for 3d point clouds classification and semantic segmentation. *Computers & Graphics*, 124:104050, 2024.

A. Appendix I: detail of model

A.1. Additional Theoretical Clarifications

This section presents the formal geometric foundations of Vector Bundle Attention (VBA). We provide structured statements—Lemma, Theorem, and Claim—to clarify how each theoretical result directly supports a specific component of the mechanism. All proofs rely on the properties of skew-symmetric generators, the definition of curvature on vector bundles, and the structure of the proposed attention operator.

A.1.1. ISOMETRIC TRANSPORT

Lemma A.1 (Isometric Parallel Transport)**.** *Let $S_{\text{skew}}(b_i, b_j)$ be the skew-symmetric generator predicted from base coordinates. The transport operator*

$$T_{j\to i} = \exp(S_{\text{skew}}(b_i, b_j)) \tag{9}$$

belongs to the special orthogonal group $SO(d_f)$ for all token pairs (i, j). Thus, for any fiber vectors $u, v \in \mathbb{R}^{d_f}$,

$$\langle T_{j\to i}u,\ T_{j\to i}v\rangle = \langle u, v\rangle, \qquad \|T_{j\to i}u\| = \|u\|. \tag{10}$$

Connection to the mechanism. This result justifies the use of the exponential map: because $T_{j\to i}$ is a pure rotation, aligning keys and values prior to computing attention ensures that similarity is evaluated in a geometry-consistent fiber coordinate system.

A.1.2. CURVATURE-INDUCED FIBER MODULATION

Lemma A.2 (Curvature-Based Feature Adjustment)**.** *Let $\Gamma(b)$ denote the learned connection field over the base manifold, and let*

$$\Omega(b) = d\Gamma(b) + \Gamma(b) \wedge \Gamma(b) \tag{11}$$

be the induced curvature 2-form. Define the effective curvature operator $\mathrm{R}_{\text{eff}}(b)$ as a contraction of $\Omega(b)$. Then the curvature-aware fiber update

$$f_i \leftarrow f_i + \mathrm{R}_{\text{eff}}(b_i)\, f_i \tag{12}$$

provides a first-order correction capturing local deviation from flat geometry.

Connection to the mechanism. This establishes why curvature appears as a multiplicative modulation of fiber features: it compensates for non-flat geometric structure and enables the model to adaptively adjust representations based on intrinsic manifold shape.

A.1.3. GEOMETRY-CONSISTENT ATTENTION

Theorem A.3 (Invariance of Transport-Then-Attend)**.** *For queries Q_i, keys K_j, and the transported keys*

$$\widetilde{K}_{j\to i} = T_{j\to i}K_j, \tag{13}$$

the VBA attention score

$$e_{ij} = \frac{\langle Q_i,\ \widetilde{K}_{j\to i}\rangle}{\sqrt{d_f}} \tag{14}$$

is invariant to arbitrary orthogonal frame rotations applied at token j. That is, for any $R \in SO(d_f)$,

$$K'_j = RK_j \quad \Longrightarrow \quad e'_{ij} = e_{ij}. \tag{15}$$

Connection to the mechanism. This theorem shows that attention in VBA is intrinsically geometric: similarity is computed in the aligned fiber at i, guaranteeing that local coordinate choices at j do not affect attention computation. This differentiates VBA from positional biases or post-hoc geometric corrections used in prior Transformers.

A.1.4. Stability Under Perturbations

[Stability of Learned Transport] Let δb denote a perturbation in base coordinates. Because the exponential map is 1-Lipschitz in the neighborhood of the identity for skew-symmetric generators, the induced transport perturbation satisfies

$$\|T_{j\to i}(b+\delta b) - T_{j\to i}(b)\| \le C\|\delta b\|, \tag{16}$$

for some constant C depending on the generator network.

Connection to the mechanism. This claim explains the empirical robustness of VBA's transport operator: small geometric perturbations (e.g., noise or resolution changes) cannot cause large deviations in the transport alignment, stabilizing the entire attention computation.

A.1.5. Summary

Together, these structured results demonstrate that VBA embeds geometry directly into the attention operator through (1) isometric transport, (2) curvature-informed modulation, and (3) geometry-invariant similarity evaluation. This establishes a principled connection between differential geometric concepts and the computational mechanism used in the model.

A.1.6. Theory–Mechanism Correspondence

To make the geometric foundations of VBA clearer, Table 5 summarizes how each differential-geometric concept maps directly to a neural operator in our attention module.

Table 5. Correspondence between core geometric concepts and their neural realizations in VBA.

Theoretical Concept	Neural Implementation (VBA Module)	Role in Architecture
Bundle section (p, v): point $p \in M$ on the base manifold and vector $v \in F_p$ in the fiber	Geometric token $\text{Token}_i = (b_i, f_i)$, where b_i is the base coordinate (projected from input) and f_i is the fiber feature	**Disentanglement**: separates global geometry (b_i) from local semantic representation (f_i) so geometry can guide attention
Parallel transport $P_\gamma : F_q \to F_p$: moves a vector along a path while keeping it parallel	Transport operator $T_{j\to i} = \exp(S_{\text{skew}})$, an orthogonal matrix predicted from (b_i, b_j). Keys/values transported as $K_{j\to i} = T_{j\to i}K_j$, $V_{j\to i} = T_{j\to i}V_j$	**Alignment before attention**: aligns features into the fiber at i so attention is computed in a consistent tangent space
Metric compatibility: inner products preserved by the connection	Isometric constraint $T_{j\to i} \in \mathrm{SO}(d_f)$ via skew-symmetric S_{skew}	**Stable scores**: rotation-only transport preserves norms, making $A_{ij} \propto Q_i^\top K_{j\to i}$ geometrically meaningful
Curvature 2-form Ω: deviation from flatness / path dependence	Curvature correction: $f_i \leftarrow f_i + R_{\text{eff}}(b_i) f_i$, where R_{eff} is computed from learned curvature Ω	**Inductive bias**: curvature modulates features to adapt to non-flat manifold structure

A.1.7. Notation Summary

For clarity, Table 6 summarizes all core symbols used in the method section.

A.2. From continuous bundle objects to discrete parameterization

Vector bundle decomposition. We regard our representation as a (learned) trivializing atlas: each token i has a base coordinate $b_i \in \mathbb{R}^{d_b}$ and a fiber $F_{b_i} \cong \mathbb{R}^{d_f}$. The projection maps are linear:

$$b_i = W_b x_i, \qquad f_i^{(m)} = W_f^{(m)} x_i. \tag{17}$$

The bundle selector outputs $\alpha_i = (s(x_i))$ and we form the aggregated fiber $f_i = \sum_m \alpha_i^{(m)} f_i^{(m)}$. LayerNorm/gating are applied:

$$\bar{f}_i = \mathrm{LN}\big(\sigma(g(x_i)) \odot \phi(\sum_m \alpha_i^{(m)} A^{(m)} f_i^{(m)})\big). \tag{18}$$

Table 6. Summary of notation used in the proposed VBA model.

Symbol	Meaning
x_i	Input feature of token i
(b_i, f_i)	Base coordinate b_i and fiber feature f_i (geometric token)
d_b, d_f	Dimensions of base and fiber spaces
Q_i, K_i, V_i	Query, key, and value vectors derived from f_i
$T_{j\to i}$	Transport operator from fiber at j to fiber at i
$T_{j\to i} = \exp(S_{\text{skew}}(b_i, b_j))$	Learnable isometric transport parameterization
$T_{j\to i} \in \mathrm{SO}(d_f)$	Rotation constraint ensuring metric compatibility
$K_{j\to i}, V_{j\to i}$	Transported key and value at location i
$A_{ij} \propto Q_i^\top K_{j\to i}$	Attention score after geometric alignment
Ω	Curvature 2-form derived from learned connection
$R_{\text{eff}}(b_i)$	Effective curvature operator at base point b_i
$f_i \leftarrow f_i + R_{\text{eff}}(b_i) f_i$	Curvature-modulated fiber update

A Geometrically Constrained Transport Operator. In differential geometry, a connection defines the rules for parallel transport, which describes how a vector is transported along a path on a base manifold. A key property in many geometric spaces is isometry, meaning that the vector's length is preserved. To create a learnable operator that is both powerful and geometrically faithful, we design our `TransportNet` to structurally enforce this constraint. Specifically, we require the learned operator $T_{j\to i}$ to be a special orthogonal matrix ($T^\top T = I$, $\det(T) = 1$), ensuring that it performs a pure rotation in the fiber space.

This is achieved by parameterizing the operator via the matrix exponential. Given base coordinates (b_i, b_j), `TransportNet`, a lightweight MLP, outputs a generator matrix $S \in \mathbb{R}^{d_f \times d_f}$, which is projected to the skew-symmetric space:

$$S_{\text{skew}} = \tfrac{1}{2}(S - S^\top). \tag{19}$$

The final operator is then defined as

$$T_{j\to i} = \exp(S_{\text{skew}}), \tag{20}$$

which guarantees that $T_{j\to i} \in SO(d_f)$ by construction.

We initialize the final layer of `TransportNet` with near-zero weights, so that $S_{\text{skew}} \approx 0$ at the start of training. This makes $T_{j\to i} \approx I$, stabilizing early optimization.

Curvature Proxy. Our model moves beyond simple algebraic proxies to compute a high-fidelity, position-dependent approximation of the formal curvature 2-form, $\Omega = d\Gamma + \Gamma \wedge \Gamma$. This section provides a detailed breakdown of this calculation. The entire process is enabled by our learnable connection field, implemented as a neural network, `ConnectionNet`, which maps any base coordinate b to the corresponding connection coefficient matrices for each basis direction, $\{\Gamma_k(b)\}$.

The Algebraic Term ($\Gamma \wedge \Gamma$): Non-Commutativity of Transport. The $\Gamma \wedge \Gamma$ term captures the quintessential feature of curvature: the path-dependence of parallel transport. Transporting a vector around an infinitesimal parallelogram by first moving in direction i then j is not the same as moving in direction j then i. This failure to commute is the essence of curvature. In our discrete setting, this is captured by the commutator of the connection matrices for pairs of directions. For a 2D base manifold, the component is:

$$(\Gamma \wedge \Gamma)_{12} = [\Gamma_1(b), \Gamma_2(b)] = \Gamma_1(b)\Gamma_2(b) - \Gamma_2(b)\Gamma_1(b). \tag{21}$$

This term is computed directly from the outputs of the `ConnectionNet` at a given point b.

The Derivative Term ($d\Gamma$): Inhomogeneity of the Geometric Structure. The $d\Gamma$ term, or the exterior derivative of the connection, captures how the rules of transport themselves change from point to point. A non-zero $d\Gamma$ implies that the geometry is not homogeneous; the way vectors are transported at point b is different from the way they are at a nearby point $b + \epsilon$. This is calculated from the partial derivatives of the connection coefficient matrices. For a 2D base manifold, the component is:

$$(d\Gamma)_{12} = \frac{\partial \Gamma_2}{\partial x_1} - \frac{\partial \Gamma_1}{\partial x_2}. \tag{22}$$

In our implementation, these partial derivatives are the elements of the Jacobian of the `ConnectionNet`'s output with respect to its input coordinates b. We compute this Jacobian efficiently using automatic differentiation.

The Full Curvature Tensor and Practical Considerations. The two components are combined to form the full, position-dependent curvature tensor, $\Omega(b)$, whose components are $\Omega_{ij}(b) = (d\Gamma)_{ij} + (\Gamma \wedge \Gamma)_{ij}$. This tensor provides a rich, local description of the learned manifold's geometry that is used to modulate the fiber representations.

This direct computation represents a deliberate trade-off between theoretical rigor and computational cost. Calculating the Jacobian of the `ConnectionNet` for a batch of points is computationally intensive. However, we argue this is a worthwhile investment for tasks where capturing the precise local geometry is critical, and it demonstrates the feasibility of directly integrating core objects of differential geometry into a network's architecture.

A.3. Endpoint-Dependent Transport vs. Formal Path-Dependent Parallel Transport

A frequent source of confusion is the term *path-dependent*. In differential geometry, parallel transport along a connection is intrinsically path-dependent: for points $b_j \to b_k \to b_i$ on a curve γ, the operator satisfies the **composition law** $T_{k\to i} \circ T_{j\to k} = T_{j\to i}$, where each T is the path-ordered exponential of the connection integrated *along the specific path segment*. Nonzero curvature further implies nontrivial holonomy around loops.

Our endpoint-dependent surrogate. For tractability, our `TransportNet` maps endpoints to a transport operator,

$$T_{j\to i} \;=\; \texttt{TransportNet}(b_i, b_j) \in SO(d_f), \tag{23}$$

without explicitly integrating a connection along a path. This design does not enforce the composition law and therefore is *not* a faithful implementation of path-dependent parallel transport. Instead, it should be understood as an *endpoint-dependent surrogate* that summarizes the effect of transporting along an implicit canonical path between b_j and b_i. We do not assume or compute geodesics; any reference to a 'canonical path' is descriptive rather than algorithmic.

Rationale and implications. Relaxing the composition law yields a lightweight, expressive operator that remains geometry-aware ($SO(d_f)$ parameterization and base coordinates) while avoiding the cost of path integration . The trade-off is that properties tied to true path dependence, such as exact compositionality and holonomy derived from a single underlying connection, are not guaranteed. We find that this surrogate suffices for our tasks, but we view efficient approximations to genuine path integrals as promising future work.

A.4. Mathematical Details of the Curvature Calculation

Our model moves beyond simple algebraic proxies to compute a high-fidelity, position-dependent approximation of the formal curvature 2-form, $\Omega = d\Gamma + \Gamma \wedge \Gamma$. This section provides the detailed mathematical and implementation framework.

The Learnable Connection Field. The entire calculation is enabled by a learnable connection field, implemented as a neural network we term `ConnectionNet`. For a base manifold of dimension d_b, this network maps any base coordinate $b \in \mathbb{R}^{d_b}$ to a set of d_b connection coefficient matrices, $\{\Gamma_k(b)\}_{k=1}^{d_b}$. Each matrix $\Gamma_k(b)$ represents the connection along the k-th coordinate direction and is an element of the general linear Lie algebra $\mathfrak{gl}(d_f, \mathbb{R})$, meaning it is a real-valued $d_f \times d_f$ matrix that acts on the fiber space.

Generalization to Higher Dimensions ($d_b > 2$). The full curvature 2-form $\Omega(b)$ is a tensor whose components, for any pair of directions i and j, are given by the well-known formula:

$$\Omega_{ij}(b) = \frac{\partial \Gamma_j}{\partial x_i} - \frac{\partial \Gamma_i}{\partial x_j} + [\Gamma_i(b), \Gamma_j(b)]. \tag{24}$$

In a general d_b-dimensional space, there are $\binom{d_b}{2}$ such unique components. Our framework is fully general and can, in principle, compute this entire tensor. The derivative term involving the partial derivatives is computed from the Jacobian of the `ConnectionNet` via automatic differentiation. The algebraic term is computed from the commutator (Lie bracket) of the corresponding output connection matrices.

However, computing the full Jacobian and all $\binom{d_b}{2}$ commutator components for a high-dimensional base manifold is computationally prohibitive. Therefore, we clarify our practical implementation. For a general d_b, we compute a scalar summary of the total curvature by summing the magnitudes of a subset of the most significant components, or by learning a low-dimensional projection of the full tensor. For the 2D case presented for illustration, we compute the single component $\Omega_{12}(b)$ directly. This represents a pragmatic trade-off that retains the core principles of the formal theory while ensuring computational tractability.

A.5. Theoretical Guarantees

In this section, we provide theoretical insights into the properties of our geometric attention mechanism. We begin by establishing a fundamental principle that an ideal geometric attention operator should satisfy, and then we analyze the stability of our practical implementation.

Theorem A.4 (1. Invariance to Local Orthogonal Frame Changes). *Let a change of local basis in the fiber at each base point b_p be represented by an orthogonal matrix $S_p \in O(d_f)$. An ideal transport operator should transform according to the rules of a geometric connection, i.e., $T'_{j\to i} = S_{b_i}\, T_{j\to i}\, S_{b_j}^{-1}$. If the fiber vectors (Queries, Keys) also transform accordingly ($Q'_i = S_{b_i} Q_i, K'_j = S_{b_j} K_j$), then the pre-softmax attention scores $e_{ij} = \langle Q_i,\, T_{j\to i} K_j\rangle$ are invariant.*

Proof. By direct substitution,

$$\begin{aligned}\langle Q'_i,\, T'_{j\to i} K'_j\rangle &= (S_{b_i} Q_i)^\top \left(S_{b_i} T_{j\to i} S_{b_j}^{-1} \cdot S_{b_j} K_j\right) \\ &= Q_i^\top S_{b_i}^\top S_{b_i}\, T_{j\to i} K_j \\ &= Q_i^\top T_{j\to i} K_j = e_{ij},\end{aligned} \tag{25}$$

where we used the orthogonality condition $S_{b_i}^\top S_{b_i} = I$. □

Remark on Theorem 1's Applicability. Theorem 1 establishes a fundamental desirable property for any robust geometric attention mechanism: its output should not depend on the arbitrary choice of coordinate systems (bases) in each local fiber space. While our `TransportNet`, even when constrained to produce orthogonal transformations (isometries), is not guaranteed to satisfy the required equivariance property by construction, this theorem provides a strong theoretical motivation for its design. It also suggests a clear path for future work: designing `TransportNet` architectures that are explicitly equivariant to such transformations, which could lead to even better generalization.

[2. Pre-softmax Score Perturbation Bound]

Let the transport operator be parameterized residually as

$$T_{j\to i} = I + A_{j\to i}, \tag{26}$$

where $A_{j\to i}$ is initialized near zero. For fixed Q_i and K_j, the change in the attention logit is bounded by

$$|e_{ij} - e_{ij}^{(0)}| \le \|Q_i\|_2\, \|A_{j\to i}\|_2\, \|K_j\|_2, \tag{27}$$

where $e_{ij}^{(0)} = \langle Q_i, K_j\rangle$.

Proof. We have

$$e_{ij} - e_{ij}^{(0)} = \langle Q_i, A_{j\to i} K_j\rangle. \tag{28}$$

By the Cauchy–Schwarz inequality,

$$|\langle Q_i, A_{j\to i} K_j\rangle| \le \|Q_i\|_2\, \|A_{j\to i} K_j\|_2. \tag{29}$$

By the definition of the operator norm,

$$\|A_{j\to i} K_j\|_2 \le \|A_{j\to i}\|_2\, \|K_j\|_2, \tag{30}$$

which gives the stated bound. □

Remark on Consistency with the Exponential Implementation. In practice, our transport operator is parameterized as the *matrix exponential* of a skew-symmetric generator,

$$T_{j\to i} = \exp(S_{\text{skew}}). \tag{31}$$

Near initialization, where S_{skew} is close to zero, we can use the first-order approximation

$$\exp(S_{\text{skew}}) \approx I + S_{\text{skew}}, \tag{32}$$

which exactly matches the residual form assumed in Theorem 2. Therefore, the perturbation bound is directly applicable to the early stages of training, explaining why the initialization near the identity stabilizes optimization. As training progresses, higher-order terms in the exponential expansion enrich the expressivity of the operator, while the isometric property is preserved by construction.

A.6. Contraction and Invariance of the Curvature Tensor

Let M be a d_b-dimensional base manifold with a Riemannian metric g and $F \to M$ a rank-d_f vector bundle modeling the fiber features. Given a learnable connection field (from `ConnectionNet`), we denote its curvature by $\Omega \in \Omega^2(M, \mathrm{End}(F))$ with local components $\{\Omega_{ij}(b)\}_{1\le i<j\le d_b}$, so $\Omega_{ij}(b) \in \mathbb{R}^{d_f\times d_f}$ and the full tensor lives in $\mathbb{R}^{d_b\times d_b\times d_f\times d_f}$. Our goal is to contract the base-space indices of Ω into a single corrective endomorphism $R_{\text{eff}}(b) \in \mathbb{R}^{d_f\times d_f}$ that modulates $\bar{f}_i$ via $\bar{f}_i \leftarrow \bar{f}_i + R_{\text{eff}}(b_i)\,\bar{f}_i$. We require that the construction be *invariant to orthogonal coordinate changes on the base* ($O(d_b)$), and stable numerically.

A canonical $O(d_b)$-invariant contraction. Changing orthonormal coordinates on T_bM induces an $O(d_b)$ action on $\Lambda^2 T_bM$, thus the index pair (i, j) mixes via an orthogonal matrix. A canonical, rotation-invariant contraction that removes the base indices and yields a fiber endomorphism is the *energy operator*

$$S(b) \;:=\; \sum_{1\le i<j\le d_b} \Omega_{ij}(b)^\top\, \Omega_{ij}(b) \;\in\; \mathbb{S}^{d_f}_+. \tag{33}$$

Under any $U \in O(d_b)$, the components $\{\Omega_{ij}\}$ mix orthogonally, hence $S(b)$ is unchanged: $S'(b) = \sum_{i<j} \Omega'_{ij}{}^\top \Omega'_{ij} = \sum_{i<j} \Omega_{ij}^\top \Omega_{ij} = S(b)$. If the connection is metric-compatible so that $\Omega_{ij} \in \mathfrak{so}(d_f)$, then $S(b) = -\sum_{i<j} \Omega_{ij}(b)^2$ is symmetric positive semidefinite (PSD).

Invariant scalar features for learned gating. While S retains anisotropy in the fiber, it is also useful to summarize curvature strength by scalar invariants, which are invariant to both $O(d_b)$ and fiber-basis changes:

$$\kappa_1(b) = \operatorname{tr} S(b), \qquad \kappa_2(b) = \operatorname{tr}\big(S(b)^2\big), \qquad \kappa_3(b) = \operatorname{tr}\big(S(b)^3\big), \;\ldots \tag{34}$$

We additionally include the Frobenius norm of the full curvature tensor as a compact descriptor:

$$s(b) = \|\Omega(b)\|_F^2 = \sum_{i<j} \|\Omega_{ij}(b)\|_F^2 \,. \tag{35}$$

These scalars are fed into a lightweight MLP, dubbed `CurvatureAdapter`, to produce *scalar* gates used below.

Directional contraction by a data-driven 2-form. If a preferred local plane is available (such as, from a structure tensor on M), let $\Sigma(b) \in \Lambda^2 T_bM$ be a unit 2-form constructed in an $O(d_b)$-equivariant way (such as, take the top-2 eigenvectors of a base-space structure tensor, wedge them and L^2-normalize). Define the directional contraction

$$R_{\text{dir}}(b) \;=\; \langle \Omega(b), \Sigma(b)\rangle \;=\; \sum_{i<j} \Sigma^{ij}(b)\, \Omega_{ij}(b) \;\in \mathrm{End}(F_b). \tag{36}$$

When the base coordinates rotate, both Ω and Σ co-transform, and the inner product on Λ^2 is $O(d_b)$-invariant, hence R_{dir} is invariant to base rotations while retaining directional information.

Unified effective curvature operator. We combine the canonical invariant operator S and the optional directional term R_{dir} using *invariant scalar gates* produced by `CurvatureAdapter`. Let $\kappa(b) = \big[s(b), \kappa_1(b), \kappa_2(b), \dots\big]$ be the scalar feature vector. We define

$$\tilde{S}(b) = \frac{S(b)}{\operatorname{tr} S(b) + \varepsilon}, \qquad \big(\alpha, \beta, \gamma, \delta\big) = \texttt{CurvatureAdapter}\big(\kappa(b)\big), \tag{37}$$

$$R_{\text{eff}}(b) \;=\; \alpha\, I \;+\; \beta\, \tilde{S}(b) \;+\; \gamma\, \tilde{S}(b)^2 \;+\; \delta\, R_{\text{dir}}(b)\,. \tag{38}$$

Here $\varepsilon > 0$ stabilizes normalization; I is the identity on the fiber. The use of only *scalar* gates preserves invariance: $\alpha, \beta, \gamma, \delta$ are functions of invariants and thus do not depend on the choice of coordinates.

Special case $d_b = 2$. When $d_b = 2$, there is a single curvature component $\Omega_{12}(b)$ up to sign, and $S(b) = -\Omega_{12}(b)^2$, $s(b) = \|\Omega_{12}(b)\|_F^2$. Then $R_{\text{dir}}(b) = \pm\,\Omega_{12}(b)$ if Σ is chosen as the (oriented) unit area 2-form. Hence (38) reduces to a polynomial in Ω_{12}^2 with an optional linear term in Ω_{12}.

Invariance guarantees.

Proposition A.5 (Base-rotation invariance)**.** *Under any orthonormal change of base coordinates $U \in O(d_b)$: (i) S in (33) is invariant; (ii) the scalars in (34) and (35) are invariant; (iii) if Σ is constructed $O(d_b)$-equivariantly, R_{dir} in (36) is invariant. Consequently, R_{eff} in (38) is invariant to base rotations.*

(i) follows from orthogonal mixing on Λ^2 and the sum-of-squares form; (ii) follows from the cyclicity of trace and invariance of S; (iii) follows since $\langle \cdot, \cdot \rangle$ on Λ^2 is $O(d_b)$-invariant and Ω, Σ co-transform. The scalar gates depend only on invariants, so (38) is invariant.

If $\Omega_{ij} \in \mathfrak{so}(d_f)$ (metric-compatible setting), then S is PSD and diagonalizable with an orthonormal eigenbasis; any polynomial in S is PSD and shares eigenvectors. Hence $I, \tilde{S}, \tilde{S}^2$ commute, yielding a numerically stable modulation. The scalars $\kappa_p = \operatorname{tr}(S^p)$ are invariant to fiber basis changes, and the gates α, β, γ preserve this invariance.

Relation to the scalar-only adapter. A purely scalar-driven approach would map $s(b)$ (or κ) directly to a matrix via an unconstrained MLP, $R_{\text{eff}} = \texttt{MLP}(s)$.

Our unified design keeps the scalar *gating* but lets geometry enter through S and R_{dir}, giving both invariance and controllable anisotropy.

Complexity and implementation notes. Computing S requires $d_b(d_b-1)/2$ matrix products of size $d_f \times d_f$ per location (often shared across heads). In practice we: (i) accumulate $\sum_{i<j} \Omega_{ij}^\top \Omega_{ij}$ on-the-fly in FP16 with loss-scaling; (ii) cache S per layer if curvature is reused by multiple heads; (iii) compute κ_p via power iterations on S to avoid explicit S^p when d_f is large.

Backpropagation. Let L be the loss. Gradients through (33) follow from $\frac{\partial L}{\partial \Omega_{ij}} = \Omega_{ij}\big(\frac{\partial L}{\partial S} + \frac{\partial L}{\partial S}^\top\big)$, and through (38) by standard polynomial rules; for R_{dir}, $\frac{\partial L}{\partial \Omega_{ij}} \mathrel{+}= \delta\, \Sigma^{ij}\, \frac{\partial L}{\partial R_{\text{dir}}}$. The gates' gradients come by the `CurvatureAdapter` chain rule on $\kappa(b)$.

Final update rule. With R_{eff} from (38), the feature update is

$$\bar{f}_i \leftarrow \bar{f}_i + R_{\text{eff}}(b_i)\, \bar{f}_i, \tag{39}$$

which is invariant to base rotations, stable under metric-compatible connections, and retains fiber-directional anisotropy by S and R_{dir} when used.

A.7. Triplet consistency of endpoint-conditioned transport

Our transport operator $T_{j \to i}$ is implemented as an endpoint-conditioned surrogate for true path-dependent parallel transport. To quantify how close it is to a composition-preserving connection, we evaluate a simple triplet consistency metric.

For randomly sampled triplets (i, j, k), we define the Frobenius-norm discrepancy between composed and direct transports as

$$\Delta_{ijk} \;=\; \left\| T_{k\to i}\, T_{j\to k} \;-\; T_{j\to i} \right\|_F. \tag{40}$$

We report the average discrepancy

$$\mathbb{E}_{(i,j,k)}[\Delta_{ijk}] \tag{41}$$

over triplets drawn from a trained model.

On VBA-P (ModelNet40), the trained VBA model yields

$$\mathbb{E}_{(i,j,k)}[\Delta_{ijk}] \;=\; 0.05732, \tag{42}$$

indicating that the learned endpoint-conditioned transport behaves close to a composition-preserving parallel transport in practice. We also measured transport consistency on biological datasets, obtaining 0.06712 on PBMC (VBA-SC) and 0.04958 on Breast Cancer (VBA-ST).

A.8. Complexity Analysis

The computational complexity of our geometrically principled VBA layer is higher than that of standard attention, reflecting the trade-off for a more rigorous model. The primary costs arise from two new components: the constrained transport operator and the direct calculation of the curvature tensor.

Transport Operator Complexity. Our constrained `TransportNet` computes a unique orthogonal matrix for each of the n^2 token pairs. For each pair, this involves a forward pass through an MLP (cost C_{MLP}) to produce a generator matrix, followed by a matrix exponential (cost $C_{\exp}$) to ensure orthogonality. The resulting $d_f \times d_f$ transport matrix is then applied to the key/value vectors (cost $O(d_f^2)$). The total complexity for the transport-attention stage is therefore:

$$O(n^2(C_{\text{MLP}} + C_{\exp} + d_f^2)). \tag{43}$$

Curvature Tensor Complexity. The position-dependent curvature tensor $\Omega(b)$ is computed for each of the N points. This requires evaluating the `ConnectionNet` (cost C_{ConnNet}), computing its Jacobian with respect to the base coordinates (a significant cost we denote as C_{Jacobian}), and calculating the commutator term (cost dominated by matrix multiplication, $O(d_f^3)$). The total per-layer cost for the curvature component is:

$$O(n \cdot (C_{\text{ConnNet}} + C_{\text{Jacobian}} + d_f^3)). \tag{44}$$

Overall Complexity. The overall complexity is dominated by the pairwise transport operator, which is significantly more intensive than the $O(n^2 d)$ complexity of standard self-attention. This highlights the deliberate design choice to prioritize theoretical guarantees over minimal computational cost.

Efficiency–Accuracy Trade-offs We additionally evaluate scalable variants of VBA using (1) local geometric attention and (2) a low-rank approximation of the transport operator. As shown in Table 7, both variants significantly reduce training time while maintaining competitive accuracy.

Table 7. Efficiency–accuracy trade-offs using local geometric attention and low-rank transport on the PBMC dataset.

Method	Acc(%)	F1	Time (s/epoch)
Full VBA	**78.71**	**0.63**	876
Sparse/Local Attention	77.65	0.62	560
Low-rank transport	76.90	0.60	529

(1) Local geometric attention (top-k neighbors). Because VBA learns explicit base coordinates, we can restrict transport and attention to each point's -nearest neighbors on the learned manifold. This reduces complexity from $O(n^2)$ to $O(nk)$.

On the PBMC scRNA-seq dataset, this approach achieves a 36% reduction in epoch time (from 876s to 560s) with only a 1.06-point accuracy drop (from 78.71 to 77.65). This strategy provides the best balance between efficiency and accuracy.

(2) Low-rank transport generators. Instead of predicting a full $d_f \times d_f$ generator matrix for the matrix exponential, we parameterize the generator in a low-rank form. This reduces the cost of both the exponential and the transport application. On PBMC, this yields a 40% speedup (from 876s to 529s) with a moderate accuracy trade-off (from 78.71 to 76.90).

A.9. Algorithm Details

For clarity and reproducibility, we provide the detailed pseudocode for a single layer of our Vector Bundle Attention (VBA) mechanism in Algorithm 1. The algorithm outlines the three core stages discussed in the main paper: (1) the projection of input features into the vector bundle representation (base and fiber), (2) the computation of attention scores using the learnable, geometry-dependent parallel transport operator, and (3) the aggregation of transported value vectors to produce the final output.

A.10. Differences from previous work

Our method represents a paradigm shift from *"Compare-then-Bias"* to *"Align-then-Compare"*.

Geometry inside attention. Prior models (such as Graphormer and GBT) compute dot-product similarity in the ambient feature space and add geometric information as an external bias term. In contrast, VBA aligns key–value features through a learned parallel-transport operator $T_{j\to i}$ before computing similarity. This makes geometry intrinsic to the attention operator rather than a post-hoc correction.

Intrinsic vector-bundle formulation. VBA defines attention on a learned base manifold and fiber space. The comparison $\langle Q_i, T_{j\to i}K_j\rangle$ takes place in a shared, geometry-consistent fiber. To our knowledge, no prior Transformer performs attention as an intrinsic geometric operator.

Pairwise orthogonal transport with curvature correction. VBA learns pairwise orthogonal transports and a curvature-based correction, enabling principled modeling of non-Euclidean biological and spatial manifolds. Earlier Euclidean or message-passing approaches do not provide this level of geometric expressiveness.

B. Appendix II: Additional experiment.

B.1. Common Experimental Settings.

We implement the VBA-Transformer using PyTorch. All experiments are conducted on four NVIDIA GH200 GPUs. For optimization, we use the AdamW optimizer with a learning rate of 10^{-4}. A cosine annealing learning rate schedule with 10% warm-up epochs is applied. Training is performed for 200 epochs with a batch size of 64. A fixed random seed of 42 is used across all experiments for reproducibility. We evaluate performance using standard metrics relevant to each task, detailed in the respective subsections.

B.2. Benchmarker Settings

For all baseline methods, we adhered strictly to the best practices and official implementations from their original publications to ensure a fair and rigorous comparison. Preprocessing followed each method's official pipeline. Where available for any baseline, we used the exact hyperparameters published by the original authors. In cases where optimal settings were not provided for a specific dataset, we performed a search over our predefined parameter ranges. The configuration achieving the best performance on a strictly separate, held-out validation set was then selected for the final evaluation on the test set. At no point was the validation data used for training, or the test data used for model selection. For these pretrained Transformer baselines (scBERT and scGPT), we directly use the officially released pretrained weights provided by the original authors: scBERT: Pretrained weights from [https://drive.weixin.qq.com/s?k=AJEAIQdfAAoUxhXE7r]. scGPT: Pretrained weights from [https://drive.google.com/drive/folders/1oWh_-ZRdhtoGQ2Fw24HP41FgLoomVo-y]. These models are then fine-tuned on our dataset. For the fine-tuning stage, we employed a consistent training setup across all transformer-based models to ensure a fair comparison of their architectural adaptability.

When we pretrain scGPT and scBERT on exactly the same HCL subset as VBA-SC, their performance on PBMC does not improve and even slightly degrades (scGPT decreases from 75.47% to 74.42%, scBERT decreases from 75.52% to 74.16%), whereas VBA-SC with the same HCL data remains several points ahead. This suggests that the performance gap is mainly

Algorithm 1 The Vector Bundle Attention (VBA) Layer

Parameters: Projection matrices $W_b, W_f, W_q, W_k, W_v, W_o$; Learnable fields $\texttt{ConnectionNet}(\cdot)$, $\texttt{TransportNet}(\cdot,\cdot)$; Curvature scale λ.
Input: Sequence of features $X = \{x_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^D$.
Output: Updated sequence $Y = \{y_i\}_{i=1}^N$.

▷ *1. Project inputs to initial base and fiber spaces.*
for $i = 1$ to N **do**
  $b_i \leftarrow \text{LayerNorm}(W_b x_i)$ ▷ Base manifold coordinate
  $f_i \leftarrow \text{LayerNorm}(W_f x_i)$ ▷ Initial fiber feature
end for

▷ *2. Apply Full Curvature Correction ($O(d_b)$-invariant).*
for $i = 1$ to N **do**
  $\Gamma(b_i) \leftarrow \texttt{ConnectionNet}(b_i)$ ▷ Connection 1-form at b_i
  $d\Gamma_i \leftarrow \text{Jacobian}_b(\texttt{ConnectionNet})(b_i)$ ▷ Autodiff w.r.t. base coords
  $\Omega_i \leftarrow d\Gamma_i + \Gamma(b_i) \wedge \Gamma(b_i)$ ▷ Curvature 2-form; $\wedge$ is commutator
  $S_i \leftarrow \sum_{p<q} \Omega_i[p,q]^\top \Omega_i[p,q]$ ▷ Energy operator (PSD), base-rotation invariant
  $\kappa_i \leftarrow \left[\text{tr}(S_i),\ \text{tr}(S_i^2)\right]$ ▷ Scalar invariants for gating
  $(\alpha_i, \beta_i, \gamma_i, \delta_i) \leftarrow \texttt{CurvatureAdapter}(\kappa_i)$ ▷ Scalar (coordinate-free) gates
  $\tilde{S}_i \leftarrow S_i / \left(\text{tr}(S_i) + \varepsilon\right)$ ▷ Stabilized normalization ($\varepsilon > 0$)
  if $\texttt{useDirectional}$ **then**
    $\Sigma(b_i) \leftarrow \text{ComputeEquivariant2Form}(b_i)$ ▷ Equivariant unit 2-form from structure tensor
    $R_{\text{dir},i} \leftarrow \langle \Omega_i, \Sigma(b_i) \rangle$ ▷ Directional contraction in $\text{End}(F)$
  else
    $R_{\text{dir},i} \leftarrow 0$
  end if
  $R_{\text{eff},i} \leftarrow \alpha_i I + \beta_i \tilde{S}_i + \gamma_i \tilde{S}_i^2 + \delta_i R_{\text{dir},i}$ ▷ Final corrective operator
  $f_i \leftarrow f_i + \lambda\, R_{\text{eff},i}\, f_i$ ▷ Fiber modulation; invariant & numerically stable (PSD-based)
end for

▷ *4. Compute attention using geometrically constrained parallel transport.*
for $i = 1$ to N **do** ▷ For each query token
  for $j = 1$ to N **do** ▷ For each key token
    ▷ *— Geometrically Constrained Transport —*
    $S_{j\to i} \leftarrow \text{MLP}_{\text{transport}}(b_i, b_j)$ ▷ Predict a generator matrix
    $S_{\text{skew}} \leftarrow \frac{1}{2}(S_{j\to i} - S_{j\to i}^\top)$ ▷ Enforce skew-symmetry
    $T_{j\to i} \leftarrow \exp(S_{\text{skew}})$ ▷ Compute orthogonal transport via matrix exponential
    ▷ *— Transported Attention Score —*
    $\tilde{K}_{j\to i} \leftarrow T_{j\to i} K_j$ ▷ Apply transport to the key vector
    $e_{ij} \leftarrow (Q_i^\top \tilde{K}_{j\to i})/\sqrt{d_f}$
  end for
  $\{\alpha_{ij}\}_{j=1}^N \leftarrow_j (\{e_{ij}\}_{j=1}^N)$

▷ *5. Aggregate transported values and produce final output.*
  $y_i^{\text{fiber}} \leftarrow \sum_{j=1}^N \alpha_{ij} (T_{j\to i} V_j)$ ▷ Apply transport to values and aggregate
  $y_i \leftarrow x_i + W_o y_i^{\text{fiber}}$ ▷ Project back and add residual
end for
return Y

due to the geometric attention mechanism rather than to differences in pretraining data.

B.3. Additional Result of VBA-ST

To provide a more comprehensive evaluation of clustering performance, we supplement the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) results from the main paper with visualizations of Completeness scores across all 12 DLPFC samples and human breast cancer. These metrics offer alternative perspectives on the quality of the identified spatial

Table 8. Detailed comparison of Completeness scores on 12 human DLPFC samples and the Breast Cancer dataset. The best performance per sample is highlighted in **bold**, and the second best is underlined.

Sample	BayesSpace	SpaGCN	DeepST	GraphST	BASS	DiffusionST	VBA-ST
151507	0.61	0.59	0.64	0.67	0.63	0.62	**0.68**
151508	0.59	0.56	0.59	0.56	0.61	**0.63**	0.59
151509	0.61	0.63	0.64	0.57	0.63	0.66	**0.68**
151510	0.55	0.57	**0.62**	0.58	0.56	0.59	0.59
151669	**0.65**	0.55	0.63	0.45	0.51	0.64	0.55
151670	0.60	0.54	0.53	0.40	0.46	**0.63**	0.50
151671	0.64	0.68	0.69	0.63	0.63	**0.71**	0.70
151672	0.53	0.63	0.60	**0.66**	0.63	0.64	0.65
151673	0.70	0.62	0.69	0.65	0.64	**0.73**	0.66
151674	0.48	0.53	0.65	0.60	0.69	**0.71**	0.57
151675	0.63	0.55	0.66	0.65	**0.69**	0.58	0.58
151676	0.54	0.48	0.63	0.58	0.68	**0.69**	0.53
Average	0.59	0.58	0.63	0.58	0.61	**0.65**	0.61
Breast Cancer	0.52	0.55	0.57	0.64	0.59	0.61	**0.69**

domains.

The full quantitative results are presented in Table 8. On the DLPFC dataset, VBA-ST maintains competitive performance with an average Completeness score of 0.61, outperforming baseline methods such as SpaGCN (0.58) and GraphST (0.58). While DiffusionST performs strongly on the laminar structure of DLPFC, VBA-ST demonstrates superior generalization capabilities on more heterogeneous tissues.

Most notably, on the Breast Cancer dataset, VBA-ST achieves a state-of-the-art Completeness score of **0.69**. This represents a substantial improvement over the strongest baselines, including GraphST (0.64) and DiffusionST (0.61). This result indicates that VBA-ST's geometric attention mechanism is particularly effective at preserving the integrity of complex, irregular tumor regions without fragmenting them, a key advantage over methods relying solely on local message passing.

B.4. Additional Result of VBA-SC

To provide a qualitative and visual assessment of our VBA-SC model's cell type annotation performance, we present visualizations on the PBMC and Pancreas datasets. To ensure a direct comparison, we use the t-SNE coordinates provided by the original source datasets for plotting. Figure 3 shows a side-by-side comparison where cells are colored according to their ground-truth labels versus the labels predicted by our model.

A high degree of visual concordance can be observed between the predicted labels and the ground-truth annotations for both datasets. The spatial distribution of predicted cell types closely matches that of the ground truth, indicating a low misclassification rate. This provides strong visual support for the high Accuracy and F1-scores reported in the main text.

We report wall-clock time per epoch on PBMC for all major baselines using identical hardware and software configurations. As shown in Table 9, VBA is more computationally expensive but remains feasible for biological workloads and achieves substantially higher accuracy and F1 than all alternatives.

Table 9. Wall-clock time per epoch on the PBMC dataset, measured on identical hardware (4×GH200). VBA provides substantially higher accuracy despite a higher computational cost.

Method	Accuracy (%)	F1	Avg epoch time (s)
Transformer	66.76	0.56	273
GBT	70.32	0.56	493
MQA-T	67.73	0.48	267
GQA-T	66.57	0.43	259
VBA (Ours)	**78.71**	**0.63**	876

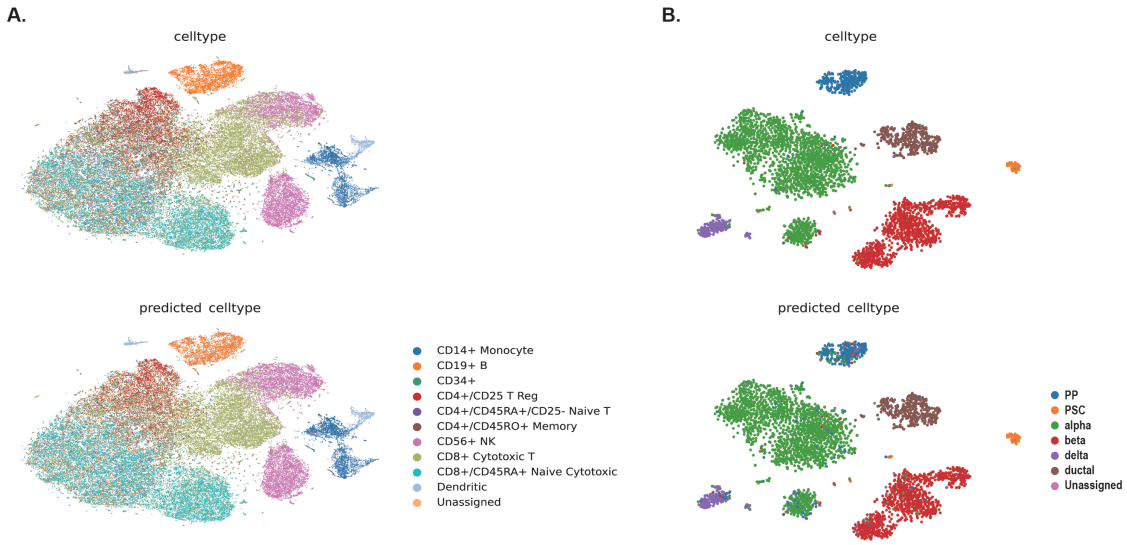

Figure 3. The t-SNE of gene expression of cells from the PBMC data (A) and the Pancreas Dataset (B). Up one is coloured by expert-annotated cell types from the original research. The down panel is colored by the prediction results of VBA-SC Model.

B.5. Additional Result of VBA-P

SO(3) Rotation Robustness To evaluate the rotation robustness of VBA-P, we follow standard SO(3) evaluation protocols and assess the model under three settings:

z/z: trained and tested with upright orientations;

z/SO(3): trained with upright orientations, tested with uniformly sampled SO(3) rotations;

SO(3)/SO(3): trained and tested with randomly sampled SO(3) rotations.

For each test shape, we apply ten independently and uniformly sampled SO(3) rotations and report the average accuracy.

Table 10. SO(3) rotation robustness on ModelNet40.

Method	z/z	z/SO(3)	SO(3)/SO(3)
VBA-T (Ours)	**92.9%**	**77.9%**	83.9%
Transformer	82.6%	17.5%	71.3%
GBT	85.8%	56.9%	73.2%
MQA-T	83.1%	28.4%	71.7%
GQA-T	83.6%	21.3%	70.9%
PointNet	89.2%	17.0%	74.7%
DGCNN	92.7%	28.4%	85.0%
PointNet++	92.5%	33.8%	**88.6%**
SVNet-PointNet	76.3%	76.3%	75.8%

Our goal here is not to outperform all SO(3)-specialized point-cloud networks, but to validate rotation robustness as a stress-test for geometric attention.

Table 10 shows that VBA-T maintains strong upright accuracy (92.9% on z/z) while substantially improving robustness to unseen rotations. In the challenging z/SO(3) setting—training on upright shapes but testing on uniformly rotated shapes, VBA-T achieves 77.9%, vastly outperforming attention-based baselines (Transformer 17.5%, MQA-T 28.4%, GQA-T 21.3%) and exceeding the geometry-biased GBT (56.9%). This indicates that biasing attention scores alone is insufficient; robustness requires aligning features across local frames before similarity computation.

Importantly, we compare against SVNet-PointNet(Su et al., 2022), a point-cloud architecture specifically designed for SO(3) rotation handling. VBA-T attains comparable performance in z/SO(3) (77.9% vs 76.3%), suggesting that our generic transport-then-attend operator can recover rotation robustness typically achieved by specialized SO(3) cloud designs. Meanwhile, VBA-T preserves substantially higher upright accuracy (92.9% vs 76.3%), reflecting a favorable trade-off: enforcing geometric consistency through parallel transport improves robustness without committing to a fully invariant architecture that may sacrifice performance on canonical orientations. When trained with SO(3) augmentation (SO(3)/SO(3)), VBA-T reaches 83.9%, remaining competitive with strong point-cloud backbones (DGCNN 85.0%, PointNet++ 88.6%) while retaining clear gains over standard Transformers.

Overall, these results support our key claim that parallel transport acts as an explicit frame-alignment mechanism, turning attention into a more stable geometric interaction operator under SO(3) perturbations.

ScanObjectNN: Real-world 3D Recognition We additionally evaluate VBA-P on ScanObjectNN (PB-T50-RS split) (Uy et al., 2019), a challenging real-world benchmark with background clutter, occlusion, and viewpoint changes.

Table 11. Performance on ScanObjectNN (PB_T50_RS split). VBA-P is competitive with recent strong 3D point-cloud models.

Model	**Accuracy (%)**
PointNet	68.2%
PointNet++	77.9%
DGCNN	78.1%
PointGPT-S (Chen et al., 2023a)	89.2%
PointMamba (Liang et al., 2024)	89.3%
ACT (Dong et al., 2022)	88.2%
VBA-P (Ours)	81.7%

As shown in Table 11, VBA-P attains **81.7%** accuracy, outperforming classical point-cloud backbones such as PointNet (68.2%), PointNet++ (77.9%), and DGCNN (78.1%). While recent point-cloud–specialized and heavily pretrained models (such as PointGPT-S and PointMamba) achieve higher absolute accuracy, VBA-P remains competitive despite not adopting point-cloud–specific architectural priors or large-scale 3D pretraining. Overall, this supplementary result supports our main claim that the proposed *transport-then-attend* operator provides a stable geometric interaction mechanism that generalizes beyond curated synthetic shapes to cluttered real-world point clouds.

B.6. QM9 Molecular Property Prediction

To assess whether VBA generalizes to molecular geometric learning, we conduct a experiment on the QM9 dataset (Ramakrishnan et al., 2014). Atomic coordinates are treated as base manifold inputs and atom-level embeddings as fiber features. We train a lightweight VBA model to predict the U_0 property using the standard QM9 split.

VBA achieves a validation MAE of **11.49 meV**, which is competitive with widely used geometric baselines such as SchNet (14 meV) (Schütt et al., 2017) and EGCN (11 meV) (Lu et al., 2019). These results indicate that vector-bundle attention transfers effectively to molecular tasks without any architectural modifications.

B.7. more details in Ablation Study.

Ablation of Architectural Components To further investigate the contribution of the specific design choices that enhance the expressivity and geometric awareness of our model, we conducted a series of ablations on its advanced components. The experiments were performed on the ModelNet40 dataset, and the results are summarized in Table 12. The results clearly demonstrate that each of these components makes a valuable and distinct contribution to the model's overall performance. The most significant performance degradation is observed when reducing the Fiber Attention to the Scaled Dot-Product Attention of MHA. Furthermore, disabling the Curvature Correction leads to a noticeable drop in accuracy, providing empirical evidence that this theoretically motivated component successfully helps the model capture the non-Euclidean nature of the data. To determine the relative importance of the curvature components, we conduct an ablation study. We evaluate a model equipped solely with the commutator term ($\Gamma \wedge \Gamma$) and compare its performance to a model equipped only with the exterior derivative ($d\Gamma$). Including the commutator term, $[\Gamma_i, \Gamma_j]$, results in a significant performance increase,

achieving an Overall Accuracy 0.8% higher and a mean-class Accuracy 1.4% higher than the model with only the $d\Gamma$ term. This indicates that the model derives more benefit from capturing the path-dependent, non-abelian nature of the learned connection.

Table 12. Ablation study of key components in the VBA-Transformer on the ModelNet40 dataset.

Model Variant	**Projection**	**Curvature**	**Fiber**	**Connection**	**OA(%)**	**mAcc(%)**	**Avg epoch time (s)**
VBA-T (Full Model)	✓	✓	✓	✓	**92.9**	**90.3**	**96.0**
w/o Projection	X	✓	✓	✓	86.1	81.9	83.4
w/o Curvature	✓	X	✓	✓	87.6	82.3	61.0
w/o $\Gamma\wedge\Gamma$	✓	only $d\Gamma$	✓	✓	88.1	83.9	75.4
w/o $d\Gamma$	✓	only $\Gamma\wedge\Gamma$	✓	✓	88.9	85.3	70.5
w/o Fiber	✓	✓	X	✓	85.7	82.4	87.8
w/o Connection	✓	✓	✓	X	86.6	83.1	86.2

To assess which geometric components matter in noisy biological settings, we perform ablations on PBMC (Table 13). Removing the learned connection or curvature degrades accuracy by 5–7 percentage points, confirming that transport and curvature correction both contribute beyond simple geometry-aware encodings.

Table 13. Ablation study of key components in VBA on the PBMC dataset.

Model Variant	**Projection**	**Curvature**	**Fiber**	**Connection**	**Acc(%)**	**F1**	**Avg epoch time (s)**
Full VBA	✓	✓	✓	✓	**78.71**	**0.63**	**876**
w/o Projection	X	✓	✓	✓	72.94	0.58	690
w/o Curvature	✓	X	✓	✓	73.28	0.60	474
w/o $\Gamma\wedge\Gamma$	✓	only $d\Gamma$	✓	✓	75.32	0.61	552
w/o $d\Gamma$	✓	only $\Gamma\wedge\Gamma$	✓	✓	75.53	0.62	498
w/o Fiber	✓	✓	X	✓	71.87	0.59	774
w/o Connection	✓	✓	✓	X	72.05	0.59	738

Curvature ablation on spatial transcriptomics. To examine whether curvature-aware modulation is useful only when geometry must be inferred latently, or whether it also helps when spatial structure is directly observed, we performed the same ablation on the Breast Cancer spatial transcriptomics dataset. As shown in Table 14, removing the curvature module causes a clear drop from 0.59/0.69 to 0.52/0.55 in ARI/NMI. Ablating its two components also degrades performance: removing the $\Gamma \wedge \Gamma$ term gives 0.52/0.57, while removing the $d\Gamma$ term gives 0.54/0.58. These results suggest that curvature-aware modulation is beneficial not only when geometry is inferred in a latent space, but also when explicit spatial coordinates are available.

Table 14. Ablation study of key components in VBA on the Breast Cancer dataset.

Model Variant	**Projection**	**Curvature**	**Fiber**	**Connection**	**ARI**	**NMI**
Full VBA	✓	✓	✓	✓	**0.59**	**0.69**
w/o Projection	X	✓	✓	✓	0.56	0.63
w/o Curvature	✓	X	✓	✓	0.52	0.55
w/o $\Gamma\wedge\Gamma$	✓	only $d\Gamma$	✓	✓	0.52	0.57
w/o $d\Gamma$	✓	only $\Gamma\wedge\Gamma$	✓	✓	0.54	0.58
w/o Fiber	✓	✓	X	✓	0.57	0.62
w/o Connection	✓	✓	✓	X	0.58	0.63

Beyond accuracy, we care about how expensive a model is to train. We adopt the Average time per epoch as the reference yardstick for training cost. This pattern indicates that the Jacobian-related $d\Gamma$ term is the main contributor to runtime

overhead, while the commutator $\Gamma \wedge \Gamma$ is comparatively cheap.

Hyperparameter Tuning and Analysis This section summarizes our hyperparameter tuning strategy and the final configurations used in all main experiments. We conducted extensive sensitivity analyses over the search space described in Table 15, where each key hyperparameter of the VBA-Transformer was systematically varied while the others were fixed to their optimal values from Table 16. Final hyperparameters for the three main tasks were chosen based on validation performance and are reported in Table 16. An exception was made for the **VBA-ST** model: to ensure fair comparison with baselines, its model dimension was fixed at 3,000, matching the input feature dimension used by competitor methods such as SpaGCN and DiffusionST.

Table 15. Hyperparameter search space explored during model tuning and sensitivity analysis.

Hyperparameter	**Tested Values (Search Space)**
Model Dimension (D)	$\{32, 64, 128, 256, 512\}$
Number of Layers	$\{6, 8, 16\}$
Number of Attention Heads	$\{4, 8, 16\}$
Fiber Dimension (d_f)	$\{8, 16, 32, 64, 128\}$
Number of Bundles (M)	$\{4, 8, 16, 32\}$
Curvature Scale (λ)	$\{0.1, 0.2\}$
Dropout Rate	$\{0.0, 0.1\}$

Table 16. Best hyperparameter configurations for the VBA-Transformer models used in the main experiments across the three tasks.

Hyperparameter	**VBA-ST**	**VBA-SC**	**VBA-P**
Model Dimension (D)	3000*	512	128
Number of Layers	16	6	16
Number of Attention Heads	16	8	16
Fiber Dimension (d_f)	64	128	64
Number of Bundles (M)	32	4	32
Curvature Scale (λ)	0.1	0.1	0.1
Dropout Rate	0.0	0.1	0.1

*Set to match baselines for fair comparison, not tuned.

B.8. Additional Scalability and Empirical Scope Analyses

Runtime and Memory Scaling We provide an additional scalability analysis to clarify the computational trade-off of VBA. VBA introduces geometry-informed pairwise transport before attention comparison, and is therefore more expensive than standard self-attention. Our intended setting is not a single dense full-set attention pass over very large biological datasets, but mini-batched subset training with shuffled subsets, which is the regime used in our large-cell experiments. For example, in PBMC, VBA-SC is trained with shuffled subsets of 2,048 cells rather than one dense pass over all approximately 66K cells.

Table 17 reports runtime and peak memory as a function of sequence length. Compared with a standard Transformer, VBA-T incurs clear additional computational overhead due to the transport module, but remains tractable under the subset-based regime used in our experiments.

Comparison with Specialized Geometric Architectures We also include additional comparisons with specialized geometric architectures to clarify the empirical scope of VBA. SE(3)-Transformer and GATr are designed primarily for rigid 3D Euclidean settings, where the task provides a known symmetry group or a strong predefined geometric/algebraic structure. By contrast, the main target domains of VBA, including single-cell RNA sequencing and spatial transcriptomics, exhibit relational or latent geometry but do not naturally provide a fixed Euclidean symmetry group. Therefore, these comparisons should be interpreted primarily as positioning evidence rather than as comparisons in the canonical target domains of SE(3)-equivariant or geometric-algebra-based models.

Table 17. Runtime and peak memory scaling with sequence length. VBA-T is more expensive than standard self-attention, but remains tractable under the subset-based training regime used in our large-cell experiments.

n	Transformer time (s/epoch)	VBA-T time (s/epoch)	Transformer memory (GB)	VBA-T memory (GB)
1K	303	591	5.72	7.86
2K	379	876	6.97	11.00
4K	618	1055	11.14	16.98
8K	1121	1803	19.03	27.38

Table 18. Comparison with specialized geometric architectures. SE(3)-Transformer (Fuchs et al., 2020) and GATr (Brehmer et al., 2023) are strong in rigid 3D settings, whereas VBA shows stronger performance in high-dimensional biological representation learning tasks.

Model	ModelNet40 OA	PBMC OA	Breast Cancer ARI
SE(3)-Transformer	91.7%	71.27%	0.45
GATr	92.3%	71.36%	0.47
VBA	92.9%	78.71%	0.59

These results suggest that specialized equivariant architectures remain naturally suited to rigid 3D geometric tasks, while VBA is better viewed as a broader geometry-informed attention mechanism for domains with latent or relational geometry. Its strongest empirical advantage is observed in high-dimensional biological representation learning.

Multi-Seed Robustness To assess robustness beyond single-run results, we repeated representative experiments with five random seeds. Table 19 reports matched multi-seed comparisons against strong baselines. The results show that VBA achieves stable performance across seeds, with particularly consistent gains on PBMC and Breast Cancer spatial transcriptomics. On ModelNet40, VBA-P remains competitive but does not outperform the strongest point-cloud baseline, which is consistent with our interpretation that VBA is not primarily optimized as a specialized 3D point-cloud architecture.

Table 19. Five-seed robustness analysis. Results are reported as mean $\pm$ standard deviation.

Task	Models	Metric 1	Metric 2
PBMC	scGPT	OA: 0.7549 ± 0.0045	F1: 0.6144 ± 0.0035
	VBA-SC	OA: 0.7893 ± 0.0039	F1: 0.6432 ± 0.0027
ModelNet40	PointGA	OA: 0.9383 ± 0.0009	mAcc: 0.9087 ± 0.0013
	VBA-P	OA: 0.9293 ± 0.0012	mAcc: 0.9040 ± 0.0026
Breast Cancer ST	DiffusionST	ARI: 0.5693 ± 0.0450	NMI: 0.5877 ± 0.0790
	VBA-ST	ARI: 0.5933 ± 0.0032	NMI: 0.6928 ± 0.0059

These multi-seed results support a more nuanced interpretation of VBA. The evidence is strongest on single-cell RNA sequencing, stable and competitive on spatial transcriptomics, and less optimized for 3D point-cloud classification compared with highly specialized point-cloud methods. Accordingly, our claim is not universal state-of-the-art performance across all geometry-related tasks, but a transferable transport-based attention mechanism whose clearest empirical benefit lies in high-dimensional biological representation learning.

C. Appendix III: Interpretability

A key advantage of the Vector Bundle Attention (VBA) architecture lies in its structured design, which offers a unique window into the model's internal workings. By decomposing each input token into multiple fiber bundle representations, our model can learn specialized feature extractors. To investigate whether the model leverages this capability, we conducted an interpretability analysis by visualizing the learned bundle mixing weights.

C.1. Methodology.

For each input point x_i in a point cloud, the bundle selector network produces normalized weights $\alpha_i = \{\alpha_i^{(m)}\}_{m=1}^{M}$, where $\alpha_i^{(m)}$ quantifies the contribution of the m-th bundle to the final representation of that point. We visualize these weights in two ways:

1. **Spatial Mapping:** We color each point i in the 3D point cloud according to its activation weight $\alpha_i^{(m)}$ for a specific bundle m. This allows us to observe which geometric regions of an object a particular bundle focuses on.
2. **Class-level Usage:** We compute the average bundle usage for an entire object class by averaging the weights $\{\alpha_i\}_i$ across all points of all instances within that class. This reveals which bundles are most important for identifying a particular object category.

C.2. Analysis of Learned Bundle Specialization.

Figure 4 presents our interpretability analysis for three instances across two object classes from the ModelNet40 dataset: one 'table' (A) and two distinct 'airplane' instances (B and C). The results reveal a clear and consistent pattern of learned bundle specialization, providing insight into the model's decision-making process.

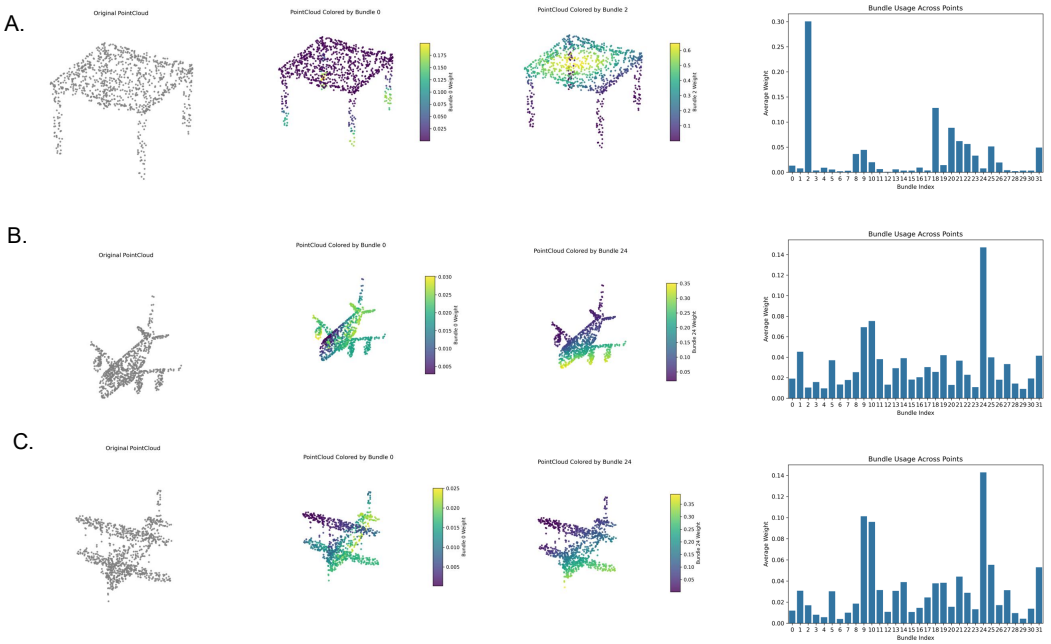

Figure 4. Visualization of learned bundle weights for a 'desk' (A) and two 'airplane' (B, C) instances, including Original point clouds, Spatial mapping of weights for the first bundle, Spatial mapping for the most-used bundle per instance, and Bar charts showing average bundle usage.

First, we observe distinct bundle usage patterns between different classes. The 'desk' instance (A) relies on a different set of primary bundles compared to the 'airplane' instances (B and C), as shown in the bar charts. This indicates that the model successfully learns to activate different feature extractors for different object categories.

More strikingly, the analysis reveals a high degree of consistency within the same class. Both airplane instances (B and C), despite variations in their specific shapes, show the highest activation for the exact same bundle (Bundle 24). The spatial mapping confirms that this specific bundle has learned to consistently focus on core structural elements of an airplane, such as the wings and fuselage. This intra-class consistency is powerful evidence that the bundles are not merely detecting random low-level features, but are learning to function as consistent, semantically meaningful feature extractors. This demonstrates that VBA-Transformer learns generalizable and interpretable representations, a key advantage of our structured geometric approach.

C.3. Interpreting the Role of Curvature

To understand where the learnable curvature component plays a key role in our architecture, we visualized its effect on representative samples from the ModelNet40 test set. Specifically, for each point $\mathbf{p}_i$ on an object, we computed the L2 norm of the final curvature correction vector, $\|\delta_c(\mathbf{p}_i)\|_2$, which is applied to the point's feature representation within each VBA Block. This scalar value directly represents the intensity of the learned non-Euclidean adjustment just before the self-attention step.

Figure 5 shows this intensity mapped as a heatmap onto an 'airplane' sample, illustrating its evolution across six different VBA Blocks. We observe a clear and intuitive pattern: the magnitude of the learned correction is highest in regions of high geometric complexity, such as the wingtips, engines, and tail assembly. This effect becomes more pronounced in deeper layers, where the model has learned to segment the object into its primary structural parts.

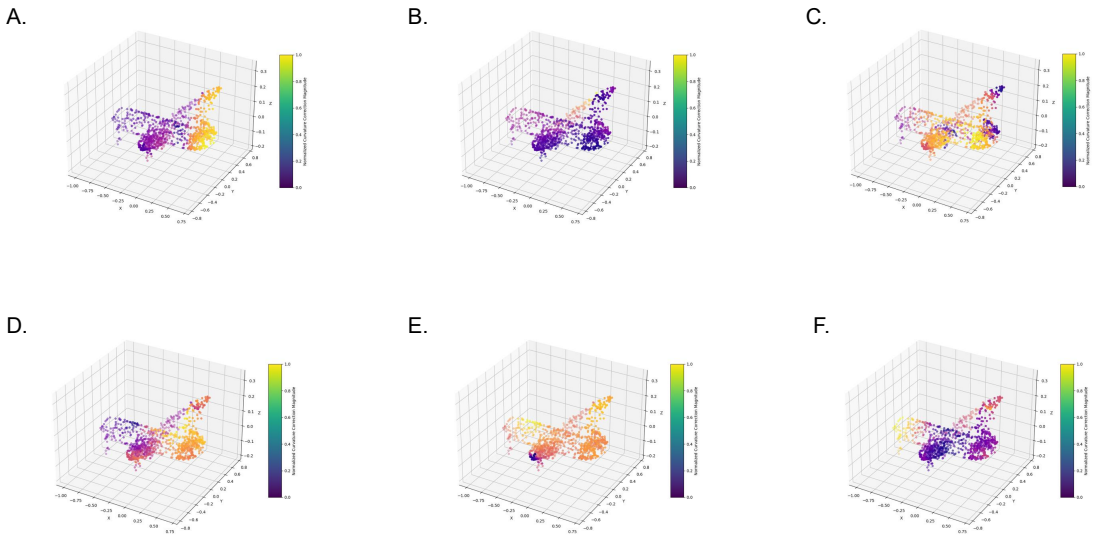

Figure 5. Visualization of the normalized curvature correction magnitude, $\|\delta_c(\mathbf{p}_i)\|_2$, for an 'airplane' sample across the six different `VBA Blocks` of the network. Yellow indicates high magnitude (strong feature correction), while purple indicates low magnitude. The progression from First Block (A) to Last Block (F) illustrates a clear hierarchical learning process.

This provides strong evidence that our model is not applying the curvature correction uniformly, but has instead learned an interpretable and meaningful representation of local geometric complexity. It leverages this understanding to apply stronger non-Euclidean adjustments only where geometrically justified, thereby adapting the feature space to the underlying structure of the data.

C.4. Biological Relevance of Learned Features.

To assess whether our model learns biologically relevant representations, we investigated the feature weights driving the annotation of a specific cell type: CD56+ Natural Killer (NK) cells from the PBMC dataset. We identified the most salient genes contributing to the model's predictions for this population.

In particular, the identified gene set represents a canonical signature of highly activated, cytotoxic NK cells and shows a strong overlap with genes known to be upregulated during NK cell generation and activation (Lehmann et al., 2012). The list is heavily enriched with genes encoding key components of the cytotoxic machinery, including GNLY, GZMA, GZMB, PRF1, and CCL4 (Figure 6).

This strong concordance with established immunological research demonstrates that VBA-SC is not operating as an uninterpretable black box. Instead, our model successfully learns to prioritize the biologically critical gene expression

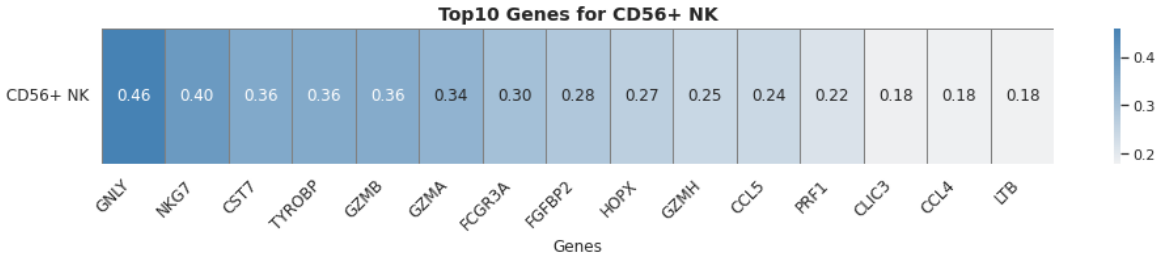

Figure 6. Feature importance scores of the top genes identified by VBA-SC as most predictive for the annotation of CD56+ Natural Killer (NK) cells.

programs that define cellular identity and function. This ability to extract meaningful biological features underscores the effectiveness of our geometric representation learning approach.

D. Limitations and Future Work

While our work demonstrates the significant potential of the VBA-Transformer, we acknowledge several limitations that also highlight exciting avenues for future research.

First, the computational complexity of the VBA layer presents a scalability challenge. The pairwise application of our learnable transport operator results in a complexity of approximately $O(n^2 d_f^2)$, where n is the sequence length and d_f is the fiber dimension (Appendix A.8). This is notably higher than a standard Transformer's $O(n^2 d)$ complexity, as our method requires a matrix vector product for each of the n^2 pairs, rather than a more efficient dot product. This can make the model computationally demanding for extremely large-scale datasets. A promising direction for future work is to integrate principles from the efficient Transformer literature, such as sparse attention (by defining local neighborhoods on the base manifold) or low-rank parameterizations of the transport operator, to mitigate this cost without sacrificing geometric fidelity.

Second, the choice of the base manifold dimension, d_b, is a key hyperparameter that governs the model's capacity to represent the underlying geometry. While our sensitivity analysis indicates that the model is robust within a reasonable range of values, identifying the optimal d_b for a novel dataset currently relies on empirical tuning. Future research could explore methods for automatically adapting or even learning the intrinsic dimensionality directly from the data, which would enhance the model's autonomy and ease of use.