Geometry in Histopathology: Comparative Analysis on Graph Neural Networks and Riemannian Manifold Embeddings in Osteosarcoma Classification

Luka Nedimović
Faculty of Computing
Union University
Belgrade, Republic of Serbia
lnedimovic2723rn@raf.rs

Abstract

Geometric deep learning (GDL) provides a powerful framework for modeling complex relational and hierarchical structures, which is particularly relevant in biomedical image analysis. Accurate osteosarcoma classification requires capturing both local and global cellular arrangements, yet existing methods often overlook these geometric relationships. In this work, we propose and evaluate three novel architectures – OsteoGNN, OSPNet, and OEHNet – that leverage graph-based, manifold-based, and hyperbolic embedding representations to encode histopathological features. We systematically study the effect of hyperparameters, patch-level embeddings, and data imbalance strategies, including weighted losses and weighted sampling, on model performance. Our results demonstrate the effectiveness of incorporating geometric priors and manifold representations in improving classification accuracy and robustness.
The PyTorch code is available at https://github.com/zenwor/osteo_gdl.

1 Introduction

1.1 Previous Work

Osteosarcoma (OS) is recognized as a malignant bone tumor [1], predominantly affecting adolescents. Histopathological analysis remains the standard for diagnosis, but manual examination is labor-intensive and subject to variability among pathologists. Automated image analysis methods can improve consistency and efficiency in diagnosis.

Machine learning (ML), and more recently deep learning, has shown remarkable potential in histopathology by automating tasks such as tumor detection, subtype classification, and prognosis prediction [2]. Convolutional neural networks (CNNs) were one of the first architectures to popularize ML approaches to histopathology tasks. CNNs use (several) convolutional layers to extract features of different locality, detecting edges and textures. For example, Cireşan et al. [3] used a deep CNN for mitosis detection in breast cancer histopathology images, while Cruz-Roa et al. [4] showcased scalability for large whole-slide images (WSIs), on task of invasive ductal carcinoma in breast cancer detection. However, CNNs struggle in learning long-range dependencies or global context in images Litjens et al. [5], hindering potential for capturing global tissue structure in WSIs.

Vision Transformers (ViTs), introduced by Dosovitskiy et al. [6] in their seminal paper, process images as a sequence of patches with global self-attention. Authors contrasted ViTs with CNNs, noting that CNNs struggle with global feature learning due to their localized convolutional operations. Vezakis et al. [7] showcased that ViTs (specifically, ViT-B/16) could lead to lower performance, compared to several CNN-based models, notably MobileNetV2 [8] and EfficientNetB0 [9]. This study also emphasizes future experimentation with respect to model size, i.e. experimenting with heavier regularization techniques, using more creative augmentation techniques, reducing input dimensionality and batch size, and adding dropout [10]. Another popular choice are Swin Transformers [11], which introduce a hierarchical structure with shifted window-based self-attention to efficiently capture local and global features at multiple scales. He et al. [12] experimented with several different architectures, including Swin Transformers, achieving 95.9% performance in classifying high-frequency lesions in PBTs and bone infections in the internal test set.

Notably, Borji et al. [13] concatenate both CNN (ResNet50 [14]) and ViT features, achieving near-perfect accuracy (99.5%).

1.2 Geometric Deep Learning

Geometric Deep Learning (GDL) is an umbrella term for emerging techniques attempting to generalize (structured) deep neural models to non-Euclidean domains such as graphs and manifolds, first introduced in the work of Bronstein et al. [15]. Even though deep learning models have been particularly successful when dealing with images (and similar signals), recently there has been a growing interest in trying to apply learning on non-Euclidean geometric data. For example, in neuroscience, graph models are used to represent anatomical and functional structures of the brain. Building on this foundation, GDL holds particular promise for image classification tasks in histopathology, where the complex spatial relationships and heterogeneous structures of tissue, such as those observed in osteosarcoma, pose significant challenges for traditional CNNs that rely on localized receptive fields.

1.3 Graph Neural Networks

Among the diverse approaches within Geometric Deep Learning, Graph Neural Networks (GNNs) have emerged as a powerful framework for processing data structured as graphs, where entities (nodes) and their relationships (edges) can represent complex systems [15]. In GNNs, information is propagated across nodes through a series of message-passing steps, allowing the model to learn representations that capture both local features of individual nodes and global patterns across the graph [16]. This capability makes GNNs particularly suitable for domains where relational data is prevalent, offering a significant advancement over traditional neural networks that operate on fixed grids, such as images. Unlike the grid-like data of standard images, histopathological whole-slide images (WSIs) can be represented as graphs, with nodes representing cells or tissue regions and edges capturing their interactions, or as manifolds to model continuous deformations reflective of pathological changes.

A graph is formally defined as $G = (V, E)$, where V is the set of nodes (or vertices) and $E \subseteq V \times V$ is the set of edges connecting the nodes. Each node $v \in V$ may have a feature vector $\mathbf{x}_v \in \mathbb{R}^d$, and each edge $(u, v) \in E$ may have an associated feature $\mathbf{e}_{uv} \in \mathbb{R}^{d_e}$.

Graph Neural Networks (GNNs) are designed to learn representations of nodes, edges, or entire graphs by recursively aggregating and transforming information from local neighborhoods. A general message-passing layer can be expressed as:

$$\mathbf{h}_v^{(k)} = \text{UPDATE}^{(k)}\Big(\mathbf{h}_v^{(k-1)}, \text{AGGREGATE}^{(k)}\big(\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}\big)\Big), \tag{1}$$

where $\mathbf{h}_v^{(k)}$ is the hidden state of node v at layer k, $\mathcal{N}(v)$ denotes the neighbors of v, AGGREGATE$^{(k)}$ is a permutation-invariant function (e.g., sum, mean, max), and UPDATE$^{(k)}$ is a learnable transformation (e.g., an MLP).

For graph-level tasks, a readout function pools node embeddings into a global graph representation:

$$\mathbf{h}_G = \text{READOUT}(\{\mathbf{h}_v^{(K)} : v \in V\}), \tag{2}$$

where K is the number of GNN layers.

Graph Convolutional Networks (GCNs) extend the concept of convolution from regular grids (like images) to irregular graph structures by performing neighborhood aggregation through a normalized adjacency matrix. Specifically, a GCN layer updates node features according to

$$\mathbf{H}^{(k)} = \sigma\Big(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}\Big), \tag{3}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\tilde{\mathbf{D}}$ is its degree matrix, $\mathbf{H}^{(k)}$ represents node embeddings at layer k, $\mathbf{W}^{(k)}$ is a learnable weight matrix, and σ is a nonlinear activation function [16]. By contrast, Graph Attention Networks (GATs) use a self-attention mechanism to assign learnable weights to neighbors, allowing the model to focus on the most relevant nodes in the neighborhood. For node i, a single-head GAT layer computes

$$\mathbf{h}_i' = \sigma\Big(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W}\mathbf{h}_j \Big), \quad \alpha_{ij} = \frac{\exp\big(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \,||\, \mathbf{W}\mathbf{h}_j])\big)}{\sum_{k \in \mathcal{N}(i)} \exp\big(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \,||\, \mathbf{W}\mathbf{h}_k])\big)}, \tag{4}$$

where $||$ denotes concatenation, $\mathbf{a}$ is a learnable attention vector, and α_{ij} represents the attention coefficient for neighbor j [17]. Both GCNs and GATs have shown strong performance in tasks where relational structure matters, including molecular property prediction, social network analysis, and histopathology image modeling.

1.4 Manifolds

A manifold is a topological space that locally resembles Euclidean space. Formally, a d-dimensional manifold $\mathcal{M}$ is a set in which every point $p \in \mathcal{M}$ has a neighborhood that is homeomorphic to $\mathbb{R}^d$. This allows us to perform calculus and define smooth functions locally, even if the global structure of the space is curved or complex. Intuitively, manifolds generalize the notion of curves and surfaces to higher dimensions, providing a flexible mathematical framework to represent structured data that does not naturally reside in flat, Euclidean space.

A Riemannian manifold $(\mathcal{M}, g)$ extends the concept of a manifold by equipping it with a smoothly varying inner product g_p on the tangent space $T_p\mathcal{M}$ at each point $p \in \mathcal{M}$. This Riemannian metric enables the measurement of geometric quantities such as distances, angles, and volumes on the manifold. The distance between two points $x, y \in \mathcal{M}$ is defined as the length of the shortest path, or geodesic, connecting them:

$$d_{\mathcal{M}}(x, y) = \min_{\gamma} \int_0^1 \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt, \tag{5}$$

where $\gamma : [0, 1] \to \mathcal{M}$ is a smooth curve such that $\gamma(0) = x$ and $\gamma(1) = y$, and $\dot{\gamma}(t)$ denotes its derivative at t.

Riemannian manifolds provide a natural framework for data that exhibits intrinsic non-Euclidean structure, such as hierarchical relations, covariance matrices, or graph representations. By operating directly on the manifold, rather than flattening the data into Euclidean space, models can respect the inherent geometry of the data, which often leads to better representation learning and improved downstream performance [18, 19].

A simple example of a Riemannian manifold is the 2-dimensional sphere. Let

$$M = S^2 = \left\{ (x, y, z) \in \mathbb{R}^3 \;\middle|\; x^2 + y^2 + z^2 = 1 \right\},$$

that is, the unit sphere embedded in $\mathbb{R}^3$.

The Riemannian metric g on M is defined as the restriction of the Euclidean inner product in $\mathbb{R}^3$ to the tangent spaces of the sphere. Concretely, for any point $p \in S^2$ and tangent vectors $u, v \in T_pS^2$,

$$g_p(u, v) = \langle u, v \rangle_{\mathbb{R}^3},$$

where $\langle \cdot, \cdot \rangle_{\mathbb{R}^3}$ is the standard Euclidean inner product. Thus, the pair $(M, g) = (S^2, g)$ forms a Riemannian manifold.

1.4.1 Hyperbolic Manifolds

Hyperbolic geometry is a non-Euclidean geometry characterized by constant negative curvature. Distances in hyperbolic space grow exponentially with displacement, which makes it particularly effective for representing hierarchical or tree-like structures [20].

The **Poincaré ball model** of hyperbolic space is defined as

$$\mathbb{B}_c^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < \frac{1}{\sqrt{c}}\},$$

where $c > 0$ is the curvature parameter. The hyperbolic distance between points $\mathbf{x}, \mathbf{y} \in \mathbb{B}^n_c$ is given by

$$d_{\mathbb{B}}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \tanh^{-1}\Big(\sqrt{c}\| \ominus \mathbf{x} \oplus \mathbf{y}\|\Big),$$

where $\oplus$ and $\ominus$ denote Möbius addition and subtraction [21]. The **exponential map** $\exp_{\mathbf{x}} : T_{\mathbf{x}}\mathbb{B}^n_c \to \mathbb{B}^n_c$ projects a tangent vector $\mathbf{v}$ at $\mathbf{x}$ to the manifold:

$$\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} \oplus \Big(\tanh(\sqrt{c}\|\mathbf{v}\|/2)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}\Big),$$

enabling gradient-based optimization directly in hyperbolic space.

1.4.2 Symmetric Positive Definite (SPD) Manifolds

SPD matrices are square matrices $X \in \mathbb{R}^{n \times n}$ that satisfy $X = X^\top$ and all eigenvalues $\lambda_i > 0$. The space of SPD matrices forms a Riemannian manifold $\mathcal{S}^n_{++}$, which cannot be treated as a Euclidean vector space [22].

Distances on SPD manifolds can be defined via metrics such as the **Affine-Invariant Riemannian Metric (AIRM)**:

$$d_{\text{AIRM}}(X, Y) = \| \log(X^{-1/2} Y X^{-1/2})\|_F,$$

where $\log$ denotes the matrix logarithm and $\| \cdot \|_F$ is the Frobenius norm. Another popular choice is the **Log-Euclidean metric**:

$$d_{\text{LEM}}(X, Y) = \| \log(X) - \log(Y)\|_F.$$

SPD-based neural networks propagate SPD matrices through layers while preserving positive definiteness, using operations such as matrix exponential, logarithm, and parallel transport [23]. This allows extraction of structured and geometrically meaningful features from data like covariance matrices or diffusion tensors.

1.5 Generalization in GDL

Generally, aforementioned architectures all fit nicely within the GDL framework, as seen in the Table 1, presented in the proto-book by Bronstein et al. [24].[1]

Table 1: Geometric Deep Learning Definitions

Architecture	**Domain Ω**	**Symmetry Group $\mathfrak{G}$**
CNN	Grid	Translation
GNN	Graph	Permutation $\mathbf{\Sigma_n}$
Transformer	Complete Graph	Permutation $\mathbf{\Sigma_n}$

2 Methodology

2.1 Dataset Description

In this study, we have used The Cancer Imaging Archive (TCIA) Osteosarcoma-Tumor-Assessment dataset (`https://www.cancerimagingarchive.net/collection/osteosarcoma-tumor-assessment/`) [25], compiled by clinical scientists at the University of Texas Southwestern Medical Center at Children's Medical Center in Dallas from 1995 to 2015. The dataset consists of 1,144 histopathological images (JPG format), and features (CSV format), consisting of 69 columns, which include various static features extracted from the images, and image classification.

Images are categorized in 4 classes:

1. NON-TUMOR (NT): healthy tissue, unaffected by cancer (bones, muscles, organs etc.)

[1] We refer reader to this book in hopes of deepening their understanding of GDL concepts

2. NON-VIABLE TUMOR (NVT): dead or necrotic tumor cells, incapable of growth / spread
3. VIABLE TUMOR (VT): living cancer cells, capable of growth, spread and metastasis
4. NON-VIABLE RATIO (NVR): proportion of living tumor cells to dead tumor cells

This dataset suffers from severe class imbalance, with NT containing the most images (536), NVT and VT containing 263 and 292 respectively, and NVR only 53 images. Due to aforementioned imbalance, in this study, we will measure model performance on 3-class (NT, NVT, VT) classification tasks.

2.2 Data Pre-processing

In this study we employ the same set of transformations for all images, due to images being high-resolution and in consideration for compatibility with pretrained models used. Namely, we resize all images to $(3, 224, 224)$, with 3 being the RGB channels, and apply normalization (originally used to train popular architectures on ImageNet [26]) with $mean = (0.485, 0.456, 0.406)$ and $std = (0.229, 0.224, 0.225)$. Additionally, to improve generalization throughout the training, in 3-class classification task, we apply a set of minority transformations (random horizontal/vertical flip, random rotation of $\pm 15^\circ$ and color jitter) to NVT samples. This augmentation is not present, i.e. is only present in only certain subset of experiments.

2.3 Training

Data split is fixed on 70-10-25 for training-validation-test, respectively. Each model has been trained on at most 50 epochs (with early stopping after at most 15 epochs, in case of no validation accuracy improvement) and batch size being 32. Euclidean learning rate starts at 10^{-4} with weight decay of 10^{-2} (heavy regularization), and Riemannian learning rate of $5 \cdot 10^{-4}$. Depending on the model, we utilize Adam optimizer for Euclidean parameters, and geoopt's [27] RiemannianAdam for Riemannian manifold parameters. Loss function is set to a weighted cross-entropy (weights being reciprocal of frequency of class, i.e. $\frac{1}{num\_class\_samples}$).

Training has been conducted on a single RTX3060 6GB GPU, 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz CPU and 16GB RAM machine.

3 Architectures

3.1 OsteoGNN: Osteosarcoma Graph Neural Network

Overview. OsteoGNN combines local patch-level visual representations with graph-based relational reasoning. An input image is partitioned into fixed-size patches, each patch is embedded by a CNN backbone, patches are connected via a k-nearest-neighbor (kNN) graph in feature space, and a Graph Neural Network (GNN) encodes the resulting patch graph. A multilayer perceptron (MLP) performs slide-level classification.

Patch extraction and embedding. Given an image $I \in \mathbb{R}^{C \times H \times W}$ and a patch size p, we extract N non-overlapping patches

$$\{P_i\}_{i=1}^{N}, \qquad P_i \in \mathbb{R}^{C \times p \times p}.$$

Each patch is embedded by a convolutional backbone f_{backbone} (classification head removed) and a linear projection:

$$z_i \;=\; \text{Dropout}\big(\, W \, \text{vec}(f_{\text{backbone}}(P_i)) \,\big) \;\in\; \mathbb{R}^d,$$

where $W \in \mathbb{R}^{d \times d_{\text{feat}}}$ projects the backbone feature to a d-dimensional latent space.

Graph construction. We form a patch graph $G = (V, E)$ with node set $V = \{1, \ldots, N\}$ and node features $\{z_i\}$. Undirected edges are built via k-NN in the embedding space:

$$(i, j) \in E \quad \Longleftrightarrow \quad j \in \text{NN}_k(i; \{z_\ell\}_{\ell=1}^{N}),$$

optionally using Euclidean distance.

GNN encoder. Let $X^{(0)} = [z_1, \dots, z_N]^\top \in \mathbb{R}^{N \times d}$ be the node-feature matrix. OsteoGNN stacks L layers of graph convolution with batch normalization, nonlinearity, and dropout:

$$X^{(\ell+1)} \;=\; \mathrm{Dropout}\Big(\sigma\big(\mathrm{BN}^{(\ell)}\big(\Phi^{(\ell)}(X^{(\ell)}, E)\big)\big)\Big), \qquad \ell = 0, \dots, L-1,$$

where $\Phi^{(\ell)}$ is a graph operator (e.g., GCN, GAT), BN is batch normalization, and σ is an activation (e.g., ReLU). A permutation-invariant readout aggregates node embeddings into a slide-level representation h:

$$h \;=\; \mathrm{POOL}\Big(\, X^{(L)} \Big) \in \mathbb{R}^{d_h},$$

with POOL chosen as mean / max / add pooling.

Classifier. An MLP maps the graph representation to class logits:

$$\hat{y} \;=\; f_{\mathrm{MLP}}(h) \in \mathbb{R}^{K},$$

where K is the number of classes. Training minimizes cross-entropy on $\hat{y}$.

Summary (end-to-end). The full pipeline is

$$I \xrightarrow{\text{patching}} \{P_i\} \xrightarrow{f_{\text{backbone}}+W} \{z_i\} \xrightarrow{k\text{-NN}} G = (V, E) \xrightarrow{\text{GNN+POOL}} h \xrightarrow{\text{MLP}} \hat{y}.$$

Optionally, fixed positional encodings can be added to $\{z_i\}$ to retain coarse spatial context before graph construction.

3.1.1 OEHNet: Osteosarcoma Euclidean-Hyperbolic Network

OEHNet (Osteosarcoma Euclidean-Hyperbolic Network) extends standard backbones with a hyperbolic head for classification. The backbone $f_\theta : \mathbb{R}^{C \times H \times W} \to \mathbb{R}^d$ extracts Euclidean feature embeddings $x \in \mathbb{R}^d$ from input images. These embeddings are then mapped to the Poincaré ball manifold $\mathbb{B}^d_c = \{z \in \mathbb{R}^d : \|z\| < 1/\sqrt{c}\}$ of constant negative curvature $-c$ via the exponential map at the origin:

$$z = \exp^c_0(x) = \tanh(\sqrt{c}\,\|x\|)\frac{x}{\|x\|}. \tag{6}$$

Within the hyperbolic space, we define a sequence of *Möbius linear layers* (HyperbolicLinear), which generalize Euclidean linear transformations to the Poincaré ball. Each layer performs a logarithmic map to the tangent space at the origin, applies a standard Euclidean linear map, and maps the result back to the manifold via the exponential map:

$$v = \log^c_0(z) = \frac{1}{\sqrt{c}}\mathrm{atanh}(\sqrt{c}\,\|z\|)\frac{z}{\|z\|}, \tag{7}$$

$$y = Wv + b, \tag{8}$$

$$\hat{z} = \exp^c_0(y) = \tanh(\sqrt{c}\,\|y\|)\frac{y}{\|y\|}. \tag{9}$$

The final classification is performed by comparing the embedded feature $\hat{z}$ to a set of *learnable class points* $\{p_i\}_{i=1}^K \subset \mathbb{B}^d_c$ via the Poincaré distance:

$$\mathrm{logits}_i = -d_{\mathbb{B}^d_c}(\hat{z}, p_i), \quad d_{\mathbb{B}^d_c}(u, v) = \frac{2}{\sqrt{c}}\mathrm{artanh}\Big(\sqrt{c}\| \ominus u \oplus v\|\Big), \tag{10}$$

where $\ominus$ and $\oplus$ denote the Möbius subtraction and addition, respectively. Training minimizes a standard cross-entropy loss on these logits.

This construction allows OEHNet to exploit the hierarchical and tree-like structures present in feature space, as hyperbolic spaces naturally encode exponential growth of distances, which can improve class separation for complex histopathological patterns.

3.2 OSPNet: Osteosarcoma Symmetric Positive Definite Manifold Network

OSPNet is a deep learning architecture designed to exploit the Riemannian geometry of *Symmetric Positive Definite (SPD)* matrices for image classification. The network consists of a feature extractor, an SPD manifold pipeline, and a classifier in the tangent space of the manifold.

Feature Extractor. Given an input image $X \in \mathbb{R}^{C\times H\times W}$, we first extract spatial features using a convolutional backbone f_{backbone}:

$$F = f_{\text{backbone}}(X), \quad F \in \mathbb{R}^{C'\times H'\times W'}.$$

We then apply an adaptive average pooling and a 1×1 convolution to reduce the channel dimension to d:

$$\tilde{F} = \text{Conv}_{1\times 1}(\text{Pool}(F)) \in \mathbb{R}^{d\times H'\times W'}.$$

SPD Manifold Pipeline. The pooled features are reshaped into matrices $Z \in \mathbb{R}^{N\times d}$, where $N = H' \cdot W'$, representing d-dimensional feature vectors per spatial location. The sample covariance matrix is computed as

$$C = \frac{1}{N-1}(Z - \bar{Z})^\top (Z - \bar{Z}) \in \mathcal{S}^d_{++},$$

where $\mathcal{S}^d_{++}$ denotes the space of $d \times d$ SPD matrices, and $\bar{Z}$ is the mean feature vector.

The SPD matrices are then processed as follows:

1. **SPDReLU**: applies an elementwise ReLU in the eigenbasis, preserving positive definiteness. If $C = U\Lambda U^\top$ is the eigendecomposition, then

$$\text{SPDReLU}(C) = U \max(\Lambda, 0)U^\top.$$

2. **Logarithmic map (Tangent projection)**: maps C onto the tangent space at the identity matrix I using the matrix logarithm:

$$T = \log(C) \in T_I\mathcal{S}^d_{++},$$

where $T_I\mathcal{S}^d_{++}$ is a Euclidean space of symmetric matrices.

3. **Flattening**: the symmetric matrix T is vectorized by stacking its upper triangular entries, producing a vector

$$v = \text{vec}(T) \in \mathbb{R}^{d(d+1)/2}.$$

Classifier. The tangent-space vectors v are fed into a fully connected network:

$$\hat{y} = \text{MLP}(v) \in \mathbb{R}^{\text{num\_classes}},$$

which outputs logits over the C classes. By operating in the tangent space of SPD matrices, the network respects the Riemannian geometry, improving representation of second-order statistics in histopathological images.

4 Experiments

4.1 Specific architectures

We evaluate each architectures in three distinct versions: T, M and L (standing for tiny, medium and large, respectively).

Table 2: OsteoGNN models

Architecture	**Backbone**	**K**	**GNN Convolution**	**GNN dims**	Pool	**MLP dims**
OsteoGNN-T	ResNet50	6	SAGEConv	$(256, 256, 128)$	Mean	$(128, 128, 3)$
OsteoGNN-M	ResNet50	6	SAGEConv	$(512, 512, 256)$	Mean	$(256, 256, 3)$
OsteoGNN-L	ResNet50	6	SAGEConv	$(1024, 1024, 512)$	Mean	$(256, 256, 3)$

Table 3: OEHNet models

Architecture	Backbone	C	Hyperbolic head dimension
OEHNet-T	ResNet18	2.0	64
OEHNet-M	ResNet18	2.0	128
OEHNet-L	ResNet18	2.0	256

Table 4: OSPNet models

Architecture	Backbone	Reduced Dimension
OSPNet-T	ResNet18	64
OSPNet-M	ResNet18	128
OSPNet-L	ResNet18	256

Metrics Analysis

We use several complementary metrics to assess model performance:

- **Accuracy**: Measures the proportion of correct predictions overall. Useful for balanced datasets, but may be misleading under strong class imbalance.
- **Precision**: Defined as $\frac{TP}{TP+FP}$. High precision indicates that when the model predicts a positive class, it is usually correct. Important when false positives are costly.
- **Recall**: Defined as $\frac{TP}{TP+FN}$. High recall indicates that the model is effective at finding most of the positive samples. Important when missing positives is costly.
- **F1 Score**: Harmonic mean of precision and recall, $F1 = 2 \cdot \frac{precision \cdot recall}{precision+recall}$. Balances both metrics, useful when classes are imbalanced.
- **AUC-ROC**: Area under the receiver operating characteristic curve. Captures the tradeoff between true positive rate (sensitivity) and false positive rate across thresholds. Robust to class imbalance.
- **Confusion Matrix**: Provides a breakdown of true positives, true negatives, false positives, and false negatives, enabling a fine-grained view of errors.

In the following table, we present test-time metrics under aforementioned experimental conditions:

Table 5: Test Results

Architecture	Accuracy	Precision	Recall	F1
OsteoGNN-T	94.14%	93.89%	94.01%	93.93%
OsteoGNN-M	93.77%	93.31%	93.81%	93.53%
OsteoGNN-L	90.84%	91.92%	89.44%	90.50%
OEHNet-T	94.87%	94.82%	94.61%	94.71%
OEHNet-M	93.78%	92.67%	93.51%	93.03%
OEHNet-L	93.41%	93.10%	92.90%	93.00%
OSPNet-T	91.21%	92.45%	89.95%	91.01%
OSPNet-M	91.94%	91.45%	91.65%	91.55%
OSPNet-L	91.94%	91.05%	92.27%	91.46%

First, both OsteoGNN and OEHNet achieve very high performance, with accuracies exceeding 93% across most variants. Notably, OEHNet-T achieves the best overall performance, with an accuracy of 94.87% and balanced precision (94.82%), recall (94.61%), and F1 score (94.71%). This suggests that the compact hyperbolic embedding head of OEHNet-T is sufficiently expressive to capture hierarchical relations in the data while avoiding the potential overfitting or optimization difficulties associated with larger heads. The larger OEHNet variants (M and L) perform slightly worse, with F1 scores dropping to 93.03% and 93.00%, respectively, indicating that simply increasing dimensionality does not necessarily yield performance gains in this domain.

In the OsteoGNN family, the tiny variant (OsteoGNN-T) also outperforms its larger counterparts, achieving 94.14% accuracy and an F1 score of 93.93%. OsteoGNN-M maintains competitive performance (93.77% accuracy, 93.53% F1), while OsteoGNN-L exhibits a more pronounced drop (90.84% accuracy, 90.50% F1). The decline observed in the large configuration may be attributed to overparameterization relative to dataset size, leading to reduced generalization capacity. This observation aligns with prior findings in graph-based models, where deeper or wider GNNs can suffer from over-smoothing and diminished discriminative power.

The OSPNet models, while still achieving strong results, consistently underperform compared to the other two families, with accuracies around 91˘92%. Interestingly, OSPNet-T, -M, and -L achieve nearly identical results, with F1 scores ranging narrowly between 91.01% and 91.55%. This stability suggests that the SPD-based representation pipeline provides robust but limited discriminative power, less sensitive to model scaling than the graph or hyperbolic approaches. However, the lack of improvement with larger configurations indicates that OSPNet may be inherently capacity-limited in its current design, possibly due to information bottlenecks introduced in the SPD layer.

Additionally, it is worth mentioning that between graph global mean, max and add pooling (which simply take the average, maximum value and add the embeddings, respectively), there has been no significant difference. In certain scenarios, max pooling tends to converge quicker, while add tends to converge slower.

Online Hard Example Mining (OHEM) [28] was applied in several experiments to help mitigate class imbalance; however, it did not produce any significant improvements in the test-time metrics.

5 Hyperparameter Study

To further investigate the expressiveness and sensitivity of the proposed architectures, we conducted a hyperparameter study focusing on key structural parameters. For the graph-based OsteoGNN models, we varied the number of neighbors K used in the KNN graph construction to assess its impact on feature aggregation and classification performance. For the hyperbolic OEHNet models, we explored different values of the curvature parameter C, which controls the geometry of the embedding space and influences how well hierarchical relationships are captured. These experiments aim to provide insight into the effect of these hyperparameters on the overall predictive capabilities of each model.

5.1 K for Dynamic KNN-based Graph Construction

For K, we chose 3 additional values, and studied their impact on feature aggregation and general generalization capacity, on OsteoGNN-L model. From Table 6, we observe that varying the number of neighbors K in the KNN graph construction has a modest impact on the model's performance. While $K = 2$ yields slightly lower overall metrics, increasing K to 4 or 8 improves recall and F1-score, suggesting that a larger neighborhood allows for more informative feature aggregation. However, the differences across K values are relatively small, indicating that the model is robust to the choice of K within this range.

Table 6: Results for varying K in KNN graph construction

K	Accuracy	Precision	Recall	F1
2	92.67%	92.38%	92.10%	92.22%
4	93.04%	92.45%	93.32%	92.80%
8	93.04%	92.15%	94.04%	92.96%

5.2 Replacing SAGEConv with GATConv in OsteoGNN

We further investigated the impact of the convolutional operator by replacing SAGEConv with GATConv in OsteoGNN, across three model sizes (T, M, L). As shown in Table 7, the choice of convolution influences both performance and stability. The medium-sized model (OsteoGNN-M) achieved the best overall results, with accuracy, precision, recall, and F1-score all around 94.5%, indicating that the attention mechanism in GATConv enables more effective neighbor feature selection and improves generalization.

In contrast, both the tiny (T) and large (L) variants achieved slightly lower performance (around 93.4% accuracy), suggesting that model size plays a critical role in balancing capacity and overfitting when using GATConv. Importantly, precision and recall remained well balanced across all three settings, which is desirable in medical classification tasks. Overall, these results highlight that the integration of GATConv provides measurable improvements over SAGEConv, particularly for the large-scale model, while maintaining robustness across different scales.

Table 7: Results for replacing SAGEConv with GATConv in OsteoGNN

Model	**Accuracy**	**Precision**	**Recall**	**F1**
OsteoGNN-T (GATConv)	93.41%	93.35%	92.85%	93.08%
OsteoGNN-M (GATConv)	94.51%	94.32%	94.32%	94.32%
OsteoGNN-L (GATConv)	93.41%	93.19%	93.26%	93.19%

5.3 Replacing ResNet50 with Vision Transformer (ViT) in OsteoGNN

To explore the impact of backbone choice on patch-level feature extraction, we replaced ResNet50 with a Vision Transformer (ViT) in OsteoGNN models of different sizes. The results, shown in Table 8, indicate that ViT backbones exhibit mixed performance compared to the standard ResNet50 models.

For the tiny and medium variants, ViT led to slightly lower overall metrics, with accuracies around 91.2% and 91.9%, respectively, compared to 94.1% and 93.8% for ResNet50. However, for the large model, the ViT backbone substantially improved accuracy (93.4%) and recall (94.5%) relative to OsteoGNN-L (90.8% accuracy), suggesting that ViT can better capture complex patch-level relationships in larger feature spaces.

These results suggest that while ResNet50 remains competitive for smaller models due to its efficient convolutional feature extraction, ViT backbones offer advantages for larger models, particularly in improving recall, likely due to their global attention mechanism capturing long-range dependencies between patches.

Table 8: Results for replacing ResNet50 with ViT in OsteoGNN models

Model	**Accuracy**	**Precision**	**Recall**	**F1**
OsteoGNN-T (ViT)	91.21%	90.17%	91.00%	90.27%
OsteoGNN-M (ViT)	91.94%	92.70%	90.17%	90.74%
OsteoGNN-L (ViT)	93.41%	92.52%	94.55%	93.35%

5.4 Curvature Parameter C in OEHNet

The curvature parameter C in OEHNet controls the geometry of the hyperbolic embedding space and can influence how hierarchical relationships between features are captured. To assess its impact, we performed experiments across three model sizes (T, M, L) while varying C, analyzing changes in classification performance. This study helps identify how sensitive OEHNet is to the choice of curvature and its interaction with embedding dimensionality.

Table 9: Results for varying C in OEHNet architecture

Architecture	**C**	**Accuracy**	**Precision**	**Recall**	**F1**
OEHNet-T	1.0	94.14%	93.93%	93.55%	93.70%
OEHNet-M		94.51%	93.76%	94.32%	94.02%
OEHNet-L		94.14%	94.11%	93.55%	93.80%
OEHNet-T	1.2	94.14%	93.28%	93.86%	93.56%
OEHNet-M		94.51%	94.11%	94.32%	94.21%
OEHNet-L		94.51%	94.81%	93.29%	93.94%
OEHNet-T	1.5	94.51%	94.57%	94.06%	94.30%
OEHNet-M		94.51%	94.47%	93.96%	94.11%
OEHNet-L		94.87%	95.06%	93.80%	94.35%

From the results in Table 9, several trends can be observed regarding the impact of the curvature parameter C and the size of the hyperbolic embeddings (T, M, L) on classification performance.

Increasing the embedding size generally yields modest improvements in accuracy and F1 score, with the largest embedding (OEHNet-L, 256-dimensional) achieving the highest accuracy of 94.87% for $C = 1.5$. This suggests that larger embedding sizes provide more expressive feature representations in the hyperbolic space, allowing the model to better capture hierarchical relations among the osteosarcoma classes.

The curvature parameter C exhibits a notable influence on the model performance. Across all embedding sizes, the highest metrics are consistently observed for $C = 1.5$, indicating that a slightly higher curvature is beneficial for mapping hierarchical features into the hyperbolic manifold. Conversely, smaller curvature values ($C = 1.0$) tend to slightly degrade performance, particularly for the medium and large embeddings, highlighting the importance of tuning C to match the intrinsic geometry of the data.

Conclusively, while the trends are clear for both embedding size and curvature, the improvements are incremental. This indicates that OEHNet is relatively robust to small variations in these hyperparameters, although careful tuning of both C and embedding dimension can yield small but meaningful gains in classification accuracy and F1 score.

6 Conclusions

In this study we present 3 architectures and their performances on osteosarcoma 3-class classification task. In addition to a graph-based model (OsteoGNN), we explore two manifold-based architectures: OEHNet, which embeds data in hyperbolic space to capture hierarchical structures, and OSPNet, which leverages the geometry of the Symmetric Positive Definite (SPD) manifold to encode covariance-based feature representations. Our findings indicate comparable performance with quite different model size, e.g. hyperbolic embeddings used in OEHNet-T yielded higher accuracy than those more structured ones of a graph.

Results highlight two key insights: (1) smaller, more compact variants (OsteoGNN-T and OEHNet-T) deliver the best performance, underscoring the importance of balancing representational power with model simplicity; (2) scaling up models beyond a certain capacity does not necessarily improve performance in histopathological contexts, and in fact may degrade generalization. The complementary strengths of the three model families - graph relational reasoning in OsteoGNN, hierarchical embedding in OEHNet, and manifold-based feature encoding in OSPNet — suggest potential for hybrid approaches that could integrate these paradigms for further performance gains.

7 Acknowledgements

To the best of the author's knowledge, no directly comparable architectures have been published in the existing research literature. The naming conventions adopted in this work (e.g., OEHNet, OSPNet) are symbolic and chosen to reflect the nature of the methods and the problem setting rather than to claim novelty of the individual components. It is acknowledged that some submodules of the proposed architectures (such as backbones, GNN layers, or manifold encoders) may have been explored in related contexts or combined in different ways in prior work. The intent of this study is not to suggest exclusivity of design, but rather to evaluate and contrast three distinct modalities—graphs, hyperbolic embeddings, and SPD manifold representations—within the osteosarcoma classification problem. Any similarity with previously developed approaches is unintentional and purely coincidental.

References

[1] QY Long, FY Wang, Y Hu, B Gao, C Zhang, BH Ban, et al. Development of the interpretable typing prediction model for osteosarcoma and chondrosarcoma based on machine learning and radiomics: a multicenter retrospective study. *Frontiers in Medicine*, 11:1–10, 2024. doi: 10.3389/fmed.2024.1497309.

[2] Daisuke Komura, Mieko Ochi, and Shumpei Ishikawa. Machine learning methods for histopathological image analysis: Updates in 2024. *Computational and Structural Biotechnology Journal*, 27:383–400, 2025. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2024.12.033. URL `https://www.sciencedirect.com/science/article/pii/S2001037024004549`.

[3] Dan C. Cireşan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 411–418, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40763-5.

[4] Angel Cruz-Roa, Ajay Basavanhally, Fabio González, Hannah Gilmore, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, and Anant Madabhushi. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In Metin N. Gurcan and Anant Madabhushi, editors, *Medical Imaging 2014: Digital Pathology*, volume 9041, page 904103. International Society for Optics and Photonics, SPIE, 2014. doi: 10.1117/12.2043872. URL `https://doi.org/10.1117/12.2043872`.

[5] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2017.07.005. URL `https://www.sciencedirect.com/science/article/pii/S1361841517301135`.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

[7] Ioannis A. Vezakis, George I. Lambrou, and George K. Matsopoulos. Deep learning approaches to osteosarcoma diagnosis and classification: A comparative methodological approach. *Cancers*, 15(8), 2023. ISSN 2072-6694. doi: 10.3390/cancers15082290. URL `https://www.mdpi.com/2072-6694/15/8/2290`.

[8] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. URL `http://arxiv.org/abs/1801.04381`.

[9] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/tan19a.html`.

[10] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022, feb 2019. doi: 10.1088/1742-6596/1168/2/022022. URL `https://dx.doi.org/10.1088/1742-6596/1168/2/022022`.

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021. URL `https://arxiv.org/abs/2103.14030`.

[12] Yi He, Ian Pan, Bing Bao, Hao Zhang, Shek Leung, Jin Chen, Jing Zhang, Hao Wang, Xin Zhou, Yuankai Wu, and Jiaping Zhang. Deep learning models in classifying primary bone tumors and bone infections based on radiographs. *npj Precision Oncology*, 9(1):45, 2025. doi: 10.1038/s41698-025-00855-3. URL `https://www.nature.com/articles/s41698-025-00855-3`.

[13] Arezoo Borji, Gernot Kronreif, Bernhard Angermayr, and Sepideh Hatamikia. Advanced hybrid deep learning model for enhanced classification of osteosarcoma histopathology images, 2024. URL `https://arxiv.org/abs/2411.00832`.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

[15] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *CoRR*, abs/1611.08097, 2016. URL `http://arxiv.org/abs/1611.08097`.

[16] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL `http://arxiv.org/abs/1609.02907`.

[17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL `https://arxiv.org/abs/1710.10903`.

[18] Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(2):127–146, 2006.

[19] Silvère Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[20] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, 2017.

[21] Octavian-Eugen Ganea, Guillaume Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *NeurIPS*, 2018.

[22] Rajendra Bhatia. *Positive definite matrices*. Princeton University Press, 2009.

[23] Zhiwu Huang and Luc Van Gool. Riemannian manifold networks with spd matrices. *IEEE TPAMI*, 2020.

[24] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021. URL `https://arxiv.org/abs/2104.13478`.

[25] Patrick Leavey, Anita Sengupta, Dinesh Rakheja, Ovidiu Daescu, Harish Babu Arunachalam, and Rashika Mishra. Osteosarcoma ut southwestern/ut dallas for viable and necrotic tumor assessment, 2019. URL `https://www.cancerimagingarchive.net/collection/osteosarcoma-tumor-assessment/`.

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[27] Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch. *CoRR*, abs/2005.02819, 2020. URL `https://arxiv.org/abs/2005.02819`.

[28] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. *CoRR*, abs/1604.03540, 2016. URL `http://arxiv.org/abs/1604.03540`.