WuBu Nesting: A Comprehensive Geometric Framework for Adaptive Multi-Scale Hierarchical Representation with Integrated Rotational Dynamics

Wubu WaefreBeorn[*]

Abstract

Real-world data frequently exhibits complex characteristics including multi-scale hierarchical organization, rotational symmetries, dynamic evolution, and regional uncertainty. While geometric deep learning offers powerful tools, existing paradigms often specialize: Euclidean models struggle with hierarchies, hyperbolic models typically handle single static hierarchies without rotational mechanics, and quaternion networks excel at rotation but lack hierarchical structure. To address these limitations, we introduce **WuBu Nesting** ("layered nesting"), a novel conceptual framework unifying these geometric properties. **WuBu Nesting** employs a recursively nested structure of hyperbolic spaces ($\mathcal{H}^{n_1}_{c_1,s_1} \supset \mathcal{H}^{n_2}_{c_2,s_2} \supset \dots$) where dimensionality ($n_i$), curvature ($c_i > 0$), and scale ($s_i > 0$) are learnable, allowing dynamic adaptation to data complexity. Each level i incorporates learnable Boundary Sub-Manifolds representing scale-specific structures. Inter-level transitions ($i \to i+1$) occur in Euclidean tangent spaces ($T_p(\mathcal{H}^{n_i}_{,}) \cong \mathbb{R}^{n_i}$), featuring a learned Rotation (R_i, e.g., SO(n_i) or Quaternions) applied simultaneously to primary data, boundary representations, and a learnable Level Descriptor ($\mathbf{ld}_i$). This is followed by a non-rotational Mapping ($\tilde{T}_i$) adjusting features and dimension. Relative Vectors ($\mathbf{d}_{i+1,j,k}$) are computed in the target tangent space, encoding rotation-aware structure. Each level also has a learnable Spread Parameter (σ_i) for uncertainty context and allows Intra-Level Tangent Flow (F_i) for local dynamics. The resulting rich information flow enables **WuBu Nesting** to capture intertwined hierarchical, rotational, dynamic, and uncertain characteristics, offering a highly flexible geometric inductive bias for complex data modeling.

1 Introduction

Effective data representation is fundamental to machine learning. While standard deep learning models achieve remarkable success, their predominant reliance on Euclidean geometry limits their ability to capture intrinsic data structures not naturally suited to flat spaces. Hierarchical data—taxonomies, phylogenetic trees, molecular structures, articulated objects, parse trees—presents a key challenge. Embedding hierarchies in Euclidean space incurs significant distortion due to the mismatch between Euclidean polynomial volume growth and the typically exponential expansion of hierarchical nodes [15].

Hyperbolic geometry, with its constant negative curvature and exponential volume growth, provides a more natural embedding space for such structures [5, 11, 15]. Models utilizing the Poincaré ball ($\mathcal{H}^n$) or other hyperbolic models have shown substantial advantages in graph embedding, NLP [6, 9], computer vision [1, 5, 11], and category discovery [13], demonstrating the benefit of aligning geometric inductive bias with data structure.

[*]Wubu WaefreBeorn, X: @WaefreBeorn Twitch: @WaefreBeorn Youtube: @WuBuStreams @WaefreBeorn

However, real-world systems often exhibit complexities beyond single, static hierarchies. Data frequently possesses **nested hierarchies** (structures within structures) and involves components with **intrinsic orientations** where transformations between levels or viewpoints include **rotations**. For instance, modeling articulated objects requires understanding both part hierarchies and their relative rotational movements. Existing hyperbolic models typically focus on a single hierarchy level with fixed geometry, lacking mechanisms for adaptive multi-scale nesting or integrated rotational modeling.

Conversely, Quaternions [10] offer efficient rotation representation, leveraged by Quaternion Neural Networks (QNNs) [7, 16] for tasks involving orientation. However, QNNs operate in Euclidean spaces, lacking intrinsic hierarchical embedding capabilities. Product manifolds ($\mathbb{R}^n \times S^m \times \mathcal{H}^k$) [8] combine geometries in parallel but do not directly model nested structures or integrated inter-level rotational transformations.

This paper introduces **WuBu Nesting**, a comprehensive conceptual framework designed to bridge these gaps by unifying adaptive multi-scale hierarchical representation with explicit modeling of rotational dynamics, dynamic evolution, and regional uncertainty. Instead of a single space or parallel product, **WuBu Nesting** proposes a nested "Russian doll" architecture of recursively embedded hyperbolic manifolds. Key innovations include:

1) **Adaptive Nested Hyperbolic Geometry:** A sequence of nested hyperbolic spaces $\mathcal{H}^{n_1}_{c_1,s_1} \supset \mathcal{H}^{n_2}_{c_2,s_2} \supset \dots$, where dimensionality n_i, curvature $c_i > 0$, and scale $s_i > 0$ are learnable, allowing dynamic geometric adaptation.

2) **Boundary Sub-Manifolds:** Learnable manifolds $B_{i,j}$ within each level $\mathcal{H}^{n_i}_{,}$ (e.g., parameterized by points $\{\mathbf{b}_{i,j,k}\}$) representing scale-specific substructures or landmarks.

3) **Tangent Space Transitions:** Inter-level transitions ($i \to i+1$) mediated within Euclidean tangent spaces $T_p(\mathcal{H}^{n_i}_{,}) \cong \mathbb{R}^{n_i}$, enabling complex yet tractable transformations.

4) **Explicit Tangent Space Rotations (R_i):** A learnable rotation R_i (e.g., SO(n_i) or Quaternions) applied within $T_o(\mathcal{H}^{n_i}_{,})$.

5) **Simultaneous Transformation:** Rotation R_i applied consistently to the main tangent vector $\mathbf{v}_i$, boundary tangent vectors $\mathbf{v}_{b_{i,j,k}}$, and a learnable Level Descriptor Vector $\mathbf{ld}_i$.

6) **Non-Rotational Mapping ($\tilde{T}_i$):** A learnable mapping $\tilde{T}_i : T_o(\mathcal{H}^{n_i}_{,}) \to T_o(\mathcal{H}^{n_{i+1}}_{,})$ following rotation, handling feature transformation and dimension changes. Full tangent transform: $T_{i \to i+1} = \tilde{T}_i \circ R_i$.

7) **Relative Vector Generation ($\mathbf{d}_{i+1}$):** Computed in the target tangent space $T_o(\mathcal{H}^{n_{i+1}}_{,})$ as $\mathbf{d}_{i+1,j,k} = \mathbf{v}_{i+1} - \mathbf{v}''_{b_{i,j,k}}$, encoding rotation-aware structure.

8) **Learnable Level Descriptor Vector ($\mathbf{ld}_i$):** An intrinsic vector $\mathbf{ld}_i \in T_o(\mathcal{H}^{n_i}_{,})$ capturing level-specific geometric properties, transformed to $\mathbf{ld}_{i+1}$ for the next level.

9) **Learnable Level Spread Parameter (σ_i):** A scalar $\sigma_i > 0$ representing scale-specific uncertainty or density, passed as context to level $i+1$.

10) **Intra-Level Tangent Flow (F_i):** A learnable field $F_i : T_o(\mathcal{H}^{n_i}_{,}) \to T_o(\mathcal{H}^{n_i}_{,})$ modeling dynamics or adjustments within level i.

11) **Rich Hierarchical Information Flow:** Level $i+1$ processing uses the primary representation, relative vectors $\{\mathbf{d}_{i+1,j,k}\}$, transformed descriptor $\mathbf{ld}_{i+1}$, and contextual spread σ_i.

We posit that this integrated geometric structure provides a powerful and flexible inductive bias for modeling complex real-world systems exhibiting intertwined hierarchical, rotational, dynamic, and uncertain characteristics.

2 Related Work

The **WuBu Nesting** framework builds upon and synthesizes concepts from several areas of geometric deep learning.

2.1 Hyperbolic Deep Learning

Pioneered by Nickel and Kiela [15], hyperbolic geometry offers superior embedding for hierarchical data compared to Euclidean spaces due to its negative curvature and exponential volume growth. Subsequent work extended this to tree embeddings [11], graph embeddings [3, 5, 18], ontologies [1], hyperbolic neural network operations [6, 9], and applications in computer vision [1, 5, 11, 13].

Critique & WuBu Distinction: Existing methods typically use a single, fixed-curvature hyperbolic space. They lack mechanisms for adaptive nested hierarchies and explicit rotational modeling. **WuBu Nesting** introduces learnable, nested geometries (n_i, c_i, s_i), boundary manifolds, level descriptors, spread parameters, tangent flows, and integrated rotation-aware tangent space transitions.

2.2 Quaternion Neural Networks (QNNs)

QNNs [7, 16] leverage the efficiency of quaternions [10] for representing 3D/4D rotations, achieving parameter efficiency and respecting rotational symmetries in tasks like 3D vision and robotics.

Critique & WuBu Distinction: QNNs operate in Euclidean spaces and lack intrinsic hierarchical embedding capabilities. **WuBu Nesting** integrates rotational modeling (potentially via quaternions) within tangent space transitions between nested hyperbolic levels, combining rotational awareness with adaptive hierarchy.

2.3 Product Manifolds and Multi-Scale Approaches

Product manifolds [8] (e.g., $\mathbb{R}^n \times S^m \times \mathcal{H}^k$) combine different geometries in parallel, increasing capacity but lacking nested structure and sophisticated inter-geometry transformations. Traditional multi-scale methods (e.g., feature pyramids) operate in Euclidean space without specific geometric biases for hierarchy or rotation.

Critique & WuBu Distinction: WuBu Nesting proposes a fundamentally different recursive embedding architecture, not parallel composition. Its inter-level transitions are designed to be geometrically meaningful, incorporating learned rotations and mappings, unlike the simpler aggregation often used in product spaces. It integrates hierarchy, scale, rotation, dynamics, and uncertainty in a unified framework distinct from standard multi-scale techniques.

3 The WuBu Nesting Framework

WuBu Nesting provides a recursive, multi-level geometric architecture. Data flows through nested hyperbolic spaces, with inter-level transitions orchestrated in tangent spaces, incorporating rotations, mappings, relative vector generation, and propagation of level-specific context (descriptors, spread).

3.1 Conceptual Architecture

Input data is initially encoded and mapped to the tangent space of the outermost level ($\mathcal{H}^{n_1}_{c_1,s_1}$). Within level i, processing occurs, potentially including intra-level tangent flow F_i. For transition $i \to i+1$, the primary representation, boundary manifold representations $\{\mathbf{b}_{i,j,k}\}$, and level descriptor $\mathbf{ld}_i$ are mapped to tangent vectors ($\mathbf{v}_i^{\text{out}}, \{\mathbf{v}_{b_{i,j,k}}\}, \mathbf{ld}_i^{\text{param}}$) via $\text{Log}^{c_i}_{o,s_i}$. A rotation R_i is applied simultaneously to these vectors in $T_o(\mathcal{H}^{n_i}_{,})$. A mapping $\tilde{T}_i$ then transforms the rotated vectors into the target tangent space $T_o(\mathcal{H}^{n_{i+1}}_{,})$, yielding $\mathbf{v}_{i+1}, \{\mathbf{v}''_{b_{i,j,k}}\}, \mathbf{ld}_{i+1}$. Relative vectors $\mathbf{d}_{i+1,j,k} = \mathbf{v}_{i+1} - \mathbf{v}''_{b_{i,j,k}}$ are computed. Level $i+1$ processing uses $\mathbf{v}_{i+1}$ (often mapped to $\mathbf{x}_{i+1}$ via $\exp^{c_{i+1}}_{o,s_{i+1}}$), relative vectors $\{\mathbf{d}_{i+1,j,k}\}$, descriptor $\mathbf{ld}_{i+1}$, and spread σ_i from level i. Aggregated information from relevant levels forms the final output. (See Figure 1).

3.2 Component Details

We elaborate on the framework's components.

3.2.1 Nested Hyperbolic Spaces & Adaptive Geometry ($\mathcal{H}^{n_i}_{c_i,s_i}, n_i, c_i, s_i$)

The structure comprises nested Poincaré Balls $\mathcal{H}^{n_i}_{c_i,s_i}$.

- **Nesting:** Allows multi-scale hierarchical modeling ($\mathcal{H}^{n_1}_{,} \supset \mathcal{H}^{n_2}_{,} \supset \dots$).
- **Dimensionality (n_i):** Learnable or fixed dimension per level, adapting capacity.
- **Curvature ($c_i > 0$):** Learnable parameter controlling geometry steepness. Requires constrained optimization.
- **Scale ($s_i > 0$):** Learnable parameter modulating tangent space mapping (Eq. 1), controlling density/zoom. Requires constrained optimization.

$$\exp^{c_i}_{o,s_i}(\mathbf{v}) = \tanh\left(s_i \cdot \frac{\sqrt{c_i}\|\mathbf{v}\|}{2}\right) \frac{\mathbf{v}}{\sqrt{c_i}\|\mathbf{v}\|} \tag{1}$$

3.2.2 Boundary Sub-Manifolds ($B_{i,j}$)

Learnable manifolds within $\mathcal{H}^{n_i}_{,}$, often parameterized by points $\{\mathbf{b}_{i,j,k}\}$, representing scale-specific landmarks or substructures. Mapped to tangent space for transformations.

3.2.3 Tangent Space Logic ($T_p(\mathcal{H}^{n_i}_{,}) \cong \mathbb{R}^{n_i}$)

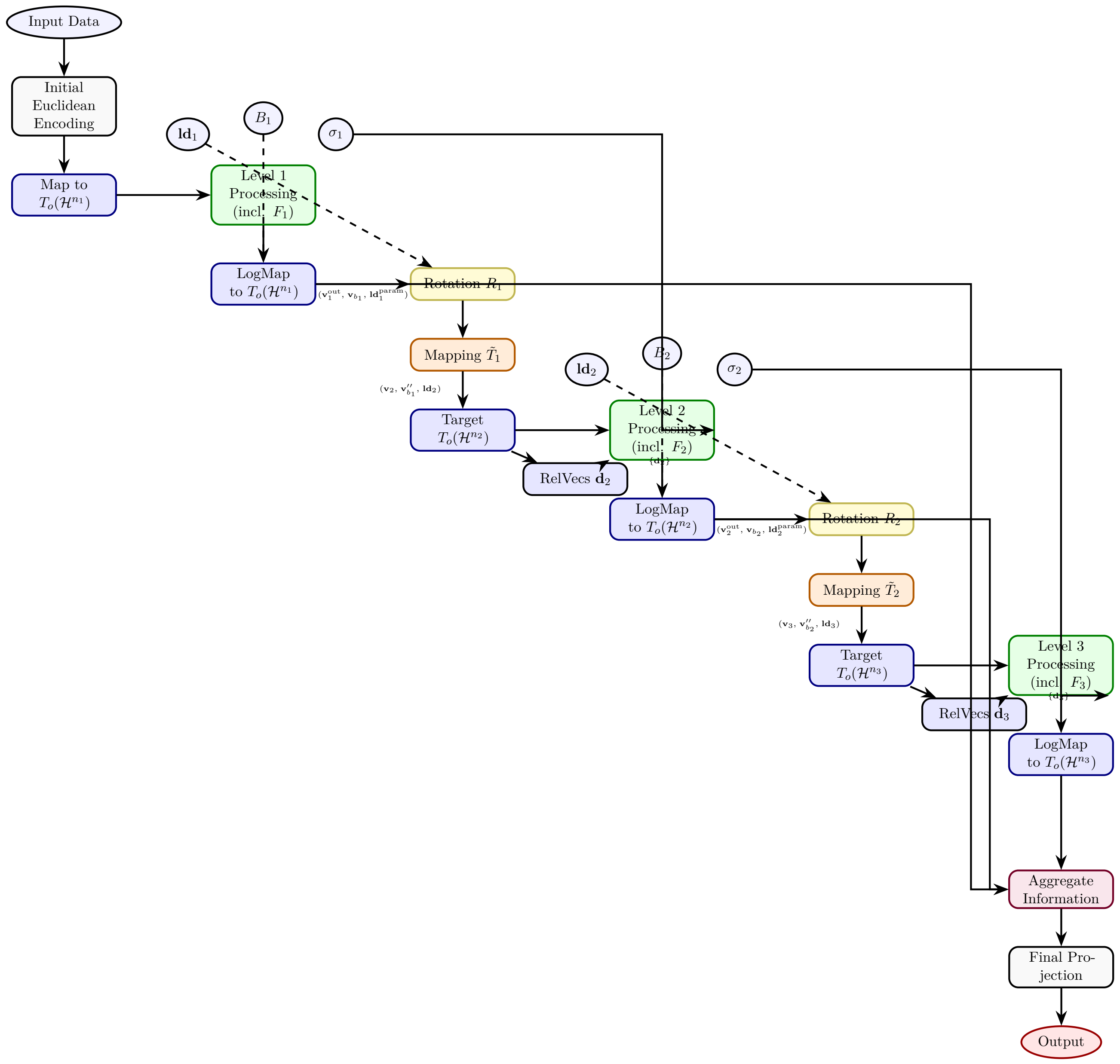
Complex transformations occur in Euclidean tangent spaces $T_p(\mathcal{H}^{n_i}_{,}) \cong \mathbb{R}^{n_i}$, using Logarithmic ($\text{Log}^{c_i}_{p,s_i}$) and Exponential ($\exp^{c_i}_{p,s_i}$) maps [12] for transitions between hyperbolic and tangent spaces.

3.2.4 Tangent Space Rotations (R_i)

Learnable rotation R_i applied in $T_o(\mathcal{H}^{n_i}_{,})$ during transition $i \to i+1$. Implemented via unit quaternions (if $n_i = 4$) or $\text{SO}(n_i)$ matrices (parameterized via exponentiation, Cayley maps, etc. [14]). Applied simultaneously to main vector $\mathbf{v}_i$, boundary vectors $\{\mathbf{v}_{b_{i,j,k}}\}$, and descriptor $\mathbf{ld}_i$.

3.2.5 Non-Rotational Mapping ($\tilde{T}_i$)

Learnable mapping $\tilde{T}_i : T_o(\mathcal{H}^{n_i}_{,}) \to T_o(\mathcal{H}^{n_{i+1}}_{,})$ applied after rotation R_i. Handles feature transformation and dimension change using MLPs, linear layers, etc.



Input Data
Initial Euclidean Encoding
Map to $T_o(\mathcal{H}^{n_1})$
$\mathbf{ld}_1$
B_1
σ_1
Level 1 Processing (incl. F_1)
LogMap to $T_o(\mathcal{H}^{n_1})$
$(\mathbf{v}_1^{\mathrm{out}}, \mathbf{v}_{b_1}, \mathbf{ld}_1^{\mathrm{param}})$
Rotation R_1
Mapping $\tilde{T}_1$
$(\mathbf{v}_2, \mathbf{v}''_{b_1}, \mathbf{ld}_2)$
Target $T_o(\mathcal{H}^{n_2})$
RelVecs $\mathbf{d}_2$
$\mathbf{ld}_2$
B_2
σ_2
Level 2 Processing (incl. F_2)
$(\mathbf{d}_2)$
LogMap to $T_o(\mathcal{H}^{n_2})$
$(\mathbf{v}_2^{\mathrm{out}}, \mathbf{v}_{b_2}, \mathbf{ld}_2^{\mathrm{param}})$
Rotation R_2
Mapping $\tilde{T}_2$
$(\mathbf{v}_3, \mathbf{v}''_{b_2}, \mathbf{ld}_3)$
Target $T_o(\mathcal{H}^{n_3})$
RelVecs $\mathbf{d}_3$
Level 3 Processing (incl. F_3)
$(\mathbf{d}_3)$
LogMap to $T_o(\mathcal{H}^{n_3})$
Aggregate Information
Final Projection
Output

Figure 1: Conceptual Architecture of the Comprehensive WuBu Nesting Framework (TikZ Diagram). Illustrates data flow through nested hyperbolic levels ($\mathcal{H}^{n_i}_{c_i,s_i}$) with adaptive parameters (n_i, c_i, s_i). Key components: Boundary Manifolds (B_i), Level Descriptors ($\mathbf{ld}_i$), Level Spreads (σ_i), Intra-Level Tangent Flows (F_i). Inter-level transitions use tangent space mapping (LogMap), simultaneous Rotation (R_i), and Mapping ($\tilde{T}_i$). Relative Vectors ($\mathbf{d}_{i+1}$) are computed. Level $i+1$ uses $\mathbf{v}_{i+1}$, $\mathbf{d}_{i+1}$, $\mathbf{ld}_{i+1}$, and σ_i. Aggregated information yields the final output.

3.2.6 Relative Vector Generation ($\mathbf{d}_{i+1,j,k}$)

Computed in target tangent space $T_o(\mathcal{H}^{n_{i+1}}_{,})$ after full transformation $T_{i \to i+1} = \tilde{T}_i \circ R_i$:

$$\mathbf{d}_{i+1,j,k} = \mathbf{v}_{i+1} - \mathbf{v}''_{b_{i,j,k}} \tag{2}$$

Encodes rotation-aware structure relative to boundaries. Used as input for level $i+1$.

3.2.7 Learnable Level Descriptor Vector ($\mathbf{ld}_i$)

Learnable vector $\mathbf{ld}_i \in T_o(\mathcal{H}^{n_i}_{,})$ capturing intrinsic level properties (e.g., anisotropy). Transformed via R_i and $\tilde{T}_i$ to $\mathbf{ld}_{i+1}$ and passed as input to level $i+1$.

3.2.8 Learnable Level Spread Parameter (σ_i)

Learnable positive scalar $\sigma_i > 0$ representing scale-specific uncertainty or density. Passed as context to level $i+1$. Requires constrained optimization.

3.2.9 Intra-Level Tangent Flow (F_i)

Learnable function $F_i : T_o(\mathcal{H}^{n_i}_{,}) \to T_o(\mathcal{H}^{n_i}_{,})$ applied during level i processing (e.g., $\mathbf{v}_{\text{flowed}} = \mathbf{v} + \text{MLP}_i(\mathbf{v})$ or $\mathbf{v}_{\text{flowed}} = M_i\mathbf{v}$). Models local dynamics or adjustments. Can use Neural ODEs [4].

3.2.10 Hierarchical Information Flow

Level $i+1$ processing module receives primary vector $\mathbf{v}_{i+1}$ (or $\mathbf{x}_{i+1}$), relative vectors $\{\mathbf{d}_{i+1,j,k}\}$, descriptor $\mathbf{ld}_{i+1}$, and spread σ_i. Enables decisions based on position, relative structure, orientation, level characteristics, and source uncertainty.

3.2.11 Scale-Aware Aggregation

Information from multiple levels ($\mathbf{v}_i^{\text{out}}, \mathbf{d}_{i,j,k}, \mathbf{ld}_i$, etc.) is aggregated for final prediction. Requires mapping representations to a common space (e.g., outermost tangent space via inverse transforms, or a dedicated output space) followed by concatenation, attention, or pooling.

4 Mathematical Formulation (Conceptual)

We outline the conceptual flow from level i to $i+1$.

Inputs to Level i Processing: Primary tangent vector $\mathbf{v}_i^{\text{in}}$; Relative vectors $\{\mathbf{d}_{i,j,k}\}$; Transformed descriptor $\mathbf{ld}_i^{\text{in}}$; Contextual spread σ_{i-1}.

Parameters for Level i: Geometry c_i, s_i; Boundary points $\{\mathbf{b}_{i,j,k}\}$; Descriptor $\mathbf{ld}_i^{\text{param}}$; Spread σ_i; Flow function F_i.

A. Intra-Level Processing (Level i):

1. Map $\mathbf{v}_i^{\text{in}}$ to $\mathbf{x}_i^{\text{in}} = \exp_{o,s_i}^{c_i}(\mathbf{v}_i^{\text{in}})$ (optional).
2. Internal module computes intermediate state using $\mathbf{x}_i^{\text{in}}$ (or $\mathbf{v}_i^{\text{in}}$), $\{\mathbf{d}_{i,j,k}\}$, $\mathbf{ld}_i^{\text{in}}$, σ_{i-1}.
3. Apply flow F_i to intermediate tangent representation $\mathbf{v}_{\text{intermediate}}$ to get $\mathbf{v}_{\text{flowed}}$.
4. Generate level output state $\mathbf{x}_i^{\text{out}} \in \mathcal{H}^{n_i}_{c_i,s_i}$.

B. Inter-Level Transition ($i \to i+1$):

1. Map to Tangent Space $T_o(\mathcal{H}^{n_i}_{,})$: $\mathbf{v}_i^{\text{out}} = \text{Log}^{c_i}_{o,s_i}(\mathbf{x}_i^{\text{out}})$; $\mathbf{v}_{b_{i,j,k}} = \text{Log}^{c_i}_{o,s_i}(\mathbf{b}_{i,j,k})$; $\mathbf{ld}_i = \mathbf{ld}_i^{\text{param}}$.
2. Apply Rotation R_i: $\mathbf{v}_i'^{\text{out}} = R_i(\mathbf{v}_i^{\text{out}})$; $\mathbf{v}'_{b_{i,j,k}} = R_i(\mathbf{v}_{b_{i,j,k}})$; $\mathbf{ld}'_i = R_i(\mathbf{ld}_i)$.
3. Apply Mapping $\tilde{T}_i$: $\mathbf{v}_{i+1} = \tilde{T}_i(\mathbf{v}_i'^{\text{out}})$; $\mathbf{v}''_{b_{i,j,k}} = \tilde{T}_i(\mathbf{v}'_{b_{i,j,k}})$; $\mathbf{ld}_{i+1} = \tilde{T}_i(\mathbf{ld}'_i)$. (Vectors now in $T_o(\mathcal{H}^{n_{i+1}}_{,})$).
4. Generate Relative Vectors: $\mathbf{d}_{i+1,j,k} = \mathbf{v}_{i+1} - \mathbf{v}''_{b_{i,j,k}}$.
5. Gather Inputs for Level $i+1$: $\mathbf{v}_{i+1}$, $\{\mathbf{d}_{i+1,j,k}\}$, $\mathbf{ld}_{i+1}$, σ_i.

This process repeats recursively.

5 Potential Experiments and Evaluation

While this paper introduces the **WuBu Nesting** framework conceptually, rigorous empirical validation is crucial future work. Evaluation should target datasets exhibiting the complex geometric properties the framework is designed to capture. Potential experimental directions include:

- **Hierarchical Classification/Regression:** Tasks involving natural hierarchies (e.g., WordNet noun hierarchy embedding [15], biological taxonomies, molecular property prediction based on functional group hierarchies). Evaluate against standard Euclidean, single-level hyperbolic models (e.g., [6]), and product manifold models [8]. Metrics: Accuracy, Mean Average Precision (MAP), distortion measures, parameter efficiency.
- **Articulated Object Pose Estimation/Reconstruction:** Datasets like Human3.6M or animal pose datasets. Evaluate ability to model joint rotations (R_i) and part hierarchies. Compare against QNNs [7] and standard pose estimation networks. Metrics: Mean Per Joint Position Error (MPJPE), angular errors. Assess the contribution of boundary manifolds and relative vectors.
- **Graph Representation Learning on Hierarchical Graphs:** Node classification and link prediction on graphs with clear hierarchical or multi-scale structure (e.g., citation networks, social networks, biological networks). Compare against Euclidean GNNs, Hyperbolic GNNs (HGCN [3], HGAT [18]), and potentially product space GNNs. Evaluate the impact of adaptive geometry (c_i, s_i) and rotational components.
- **Generative Modeling of Structured Data:** Generating realistic 3D shapes, molecules, or other data with inherent hierarchical and orientational properties. Evaluate using standard generative metrics (e.g., FID, Inception Score adapted for the domain) and measures of structural validity/diversity. Compare against geometric VAEs/GANs/Flows operating in Euclidean, hyperbolic [17], or product spaces.
- **Ablation Studies:** Systematically remove or simplify components (e.g., disable rotation R_i, remove boundary manifolds $B_{i,j}$, fix geometry c_i, s_i, remove flow F_i) to quantify the contribution of each element to overall performance on specific tasks.

- **Visualization and Interpretability:** Use visualization tools (like those in `wubu_nesting_visualization.py`) to analyze the learned geometries (c_i, s_i), boundary positions $(\mathbf{b}_{i,j,k})$, level descriptors $(\mathbf{ld}_i)$, and flows (F_i) to gain insights into how the model represents the data structure.

Success in these experiments would validate the hypothesis that integrating adaptive nested hyperbolic geometry with explicit modeling of rotations, boundaries, relative geometry, and level-specific context provides a significant advantage for modeling complex, structured real-world data.

6 Implementation Considerations & Strategy

Implementing **WuBu Nesting** involves significant challenges in mathematical rigor, numerical stability, and computational efficiency.

- **Mathematical Rigor:** Requires consistent scale-aware hyperbolic maps/metrics, stable tangent space handling, and differentiable parameterizations for $\mathrm{SO}(n_i)$ rotations [14] and flows F_i.
- **Numerical Stability:** Hyperbolic operations near boundaries require robust implementations (e.g., norm clipping [13], precision). Tangent space operations need normalization. Learning positive parameters (c_i, s_i, σ_i) requires constraints (e.g., parameterizing logarithms, using softplus).
- **Computational Cost:** Multiple levels, boundary points, and complex transformations $(R_i, \tilde{T}_i, F_i)$ increase computational load. Efficiency is key.
- **Component Design:** Effective parameterization of boundaries and flows, and designing modules to fuse the rich inter-level information stream are critical.
- **Optimization:** The complex loss landscape may necessitate Riemannian optimization methods [2, 12], careful initialization, and tailored regularization.

Incremental Implementation Strategy: A staged approach is advised: 1) Foundational 2-level fixed geometry with basic rotation/mapping. 2) Add adaptive geometry (c_i, s_i). 3) Add boundaries and relative vectors. 4) Add level descriptors and spread. 5) Add intra-level flow. 6) Expand levels. 7) Refine aggregation and optimize.

7 Discussion & Future Work

WuBu Nesting is presented as an ambitious conceptual framework unifying multi-scale hierarchy, rotation, relative geometry, dynamics, and uncertainty within a nested hyperbolic structure.

Potential Impact: Offers a path towards unified geometric modeling, potentially leading to improved representation learning, new modeling capabilities for complex systems (e.g., protein dynamics, articulated objects), and enhanced interpretability through its structured components.

Limitations & Challenges: Include significant implementation complexity, high computational cost, potential need for large structured datasets, lack of current theoretical analysis (stability, convergence, expressivity), and the challenge of interpreting complex interactions between components.

Future Work: Key directions include: 1) Formal mathematical development. 2) Robust and efficient implementation (e.g., using `geoopt` [12]). 3) Exploring component variations (boundaries, flows [4], mappings). 4) Developing tailored optimization strategies (Riemannian methods [2]). 5) Rigorous empirical validation across diverse tasks. 6) Theoretical analysis of the framework's properties. 7) Architecture search for optimal configurations. 8) Developing visualization tools for learned structures.

8 Conclusion

WuBu Nesting introduces a novel conceptual framework for deep geometric learning, uniquely integrating adaptively nested hyperbolic spaces, explicit boundary sub-manifolds, tangent space rotations and mappings, relative vector computations, learnable level descriptors, contextual level spread parameters, and intra-level tangent flows. This aims to provide an unprecedentedly rich geometric inductive bias suitable for capturing the complex interplay of multi-scale hierarchy, orientation, relative structure, scale-specific characteristics, density/uncertainty, and local dynamics inherent in many real-world datasets. While substantial theoretical and implementation challenges remain, **WuBu Nesting** represents a promising direction for developing next-generation deep learning models capable of handling profound geometric complexity, with potential applications across numerous scientific and engineering domains. Future work will focus on formalization, implementation, and empirical validation.

References

[1] Mina Ghadimi Atigh, Julian Schoep, Elmira Acar, Nanne Van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations (ICLR)*, 2019.

[3] Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[4] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[5] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[7] Eleonora Grassucci, Danilo Comminiello, and Aurelio Uncini. Quaternion neural networks: State-of-the-art and research challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[8] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning semantic representations using diffusion kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[9] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. Hyperbolic attention networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[10] William Rowan Hamilton. *Elements of quaternions*. Longmans, Green, & Company, 1866.

[11] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in PyTorch. *arXiv preprint arXiv:2005.02819*, 2020. Code: https://github.com/geoopt/geoopt.

[13] Yuzhen Liu, Zelin He, and Keke Han. Hyperbolic category discovery. *arXiv preprint arXiv:2504.06120*, 2025. (Note: Placeholder date/ID - Update when official).

[14] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. *arXiv preprint arXiv:1705.04107*, 2017.

[15] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[16] Titouan Parcollet, Mohamed Morchid, Pierre-Michel Bousquet, Renaud Dufour, Georges Linarès, and Renato De Mori. Quaternion recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[17] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations (ICLR)*, 2020.

[18] Yiding Zhang, Xiao Wang, Chuan Shi, and Yanfang Ye. Hyperbolic graph attention network. In *Proceedings of The Web Conference (WWW)*, 2021.