

NeuroHex: A Brain-Inspired Hex Coordinate System to Enable Highly Computationally-Efficient World Models for Continuous Online-Adaptive Learning

Quinn Jacobson, Joe Luo, Jingfei Xu, Shanmuga Venkatachalam, Kevin Wang, Dingchao Rong, John Paul Shen
NeuroAI Computer Architecture Lab (NCAL), Carnegie Mellon University
 {qjacobso, jpshen}@andrew.cmu.edu

Abstract—*NeuroHex*¹ is a brain-inspired hexagonal coordinate system designed to support highly efficient world models and reference frames for online adaptive AI systems. Inspired by the hexadirectional firing structure of grid cells in the human brain, *NeuroHex* adopts a cubic isometric hexagonal coordinate formulation that provides full 60° rotational symmetry and low-cost translation, rotation and distance computation. We develop a mathematical framework that incorporates ring indexing, quantized angular encoding, and a hierarchical library of foundational, simple, and complex geometric shape primitives. These constructs allow low-overhead point-in-shape tests and spatial matching operations that are expensive in Cartesian coordinate systems. To support realistic settings, we also develop a novel tool (*OSM2Hex*) that can process OpenStreetMap (OSM) data sets and convert them into the *NeuroHex* coordinate system. The *OSM2Hex* spatial abstraction processing pipeline can achieve a reduction of 90-99% in geometric complexity while maintaining the relevant spatial structure map for navigation. Our initial results, based on actual city and neighborhood scale data sets, demonstrate that *NeuroHex* offers a highly efficient substrate for building dynamic world models to enable adaptive spatial reasoning in autonomous energy-efficient AI systems with continuous online-adaptive learning (COAL) capability.

Index Terms—Neuromorphic systems, world model, grid cells, reference frames, hexagonal coordinate systems, *NeuroAI*.

I. INTRODUCTION

The biological brain has served as an inspiration for artificial intelligence (AI) and a proof-point for what can be theoretically achievable. AI and generative AI have made great progress in the last decade. Although current AI approaches are impressive, they fall short in many ways in comparison to what biological brains can achieve. There are numerous efforts to create better AI, with one approach focusing on leveraging neuroscience insights to create fundamentally new AI approaches that incorporate world models and reference frames for online adaptive AI systems [2].

The heart of our approach is to reverse engineer how the biological brain learns and organizes information to enable efficient spatial recognition and reasoning, and at the same time online adaptive learning capability. Some key hypotheses

underlie our approach. Our first hypothesis is that the foundational building blocks of our long-term memory consist of many small reference frames where information is organized according to spatial relationships. This is inspired by Hawkins and the Thousand Brains Project [3] [4]. These reference frames must be optimized for creating hierarchical organizations of information and must be extremely computationally efficient.

Our second hypothesis is that our working memory is based on a world model that handles hierarchical information and synthesizes inputs from multiple sensory modalities [5]. Our sensory input and physical interaction with the world are based on an egocentric coordinate system, while our world model and reference frames are based on allocentric coordinates. It is important that any spatial encoding of information supports efficiently converting between egocentric (polar coordinates) and allocentric (coordinates relative to local landmarks). It is also important that we can efficiently match observed objects with objects stored in our short-term and long-term memories. When matching objects, the orientation and size we observe may differ from what we have in our memory.

In this work, we limit the scope of our work to learning, organizing, and recognizing spatial information in the physical world. We want to demonstrate this before extending to more general learning and handling of abstract concepts. We also focus initially on two dimensional information. For building mental maps of the physical world and our understanding of where we are in the physical world, we emphasize position in a horizontal plane. Based on how biology has solved these same problems and our observed computational costs, working in a hexagonal coordinate system provides many benefits. We start with background information and provide the evidence that biology actually uses a hexagonal coordinate system [6], [7]. Next, we cover our simple model of the brain and the key operations we are implementing as motivation for adopting a hexagonal coordinate system. We then discuss the specific hexagonal coordinate system and spatial building blocks that we have developed in *NeuroHex*. Finally, we explore how real spatial data on city scales can be encoded in *NeuroHex* and develop a conversion tool we called *OSM2Hex*.

¹This is an expanded version of the paper titled "*NeuroHex: Highly-Efficient Hex Coordinate System for Creating World Models to Enable Adaptive AI*" published in the proceedings of the 2026 Neuro Inspired Computational Elements (NICE) [1] conference. This is an archival version of the paper and is currently under review for an ACM journal publication.

II. BACKGROUND

Recently, leading experts in both communities have suggested the need to (re)connect research in Neuroscience and Artificial Intelligence [8] to accelerate the development of next-generation AI systems with more brain-like capabilities and efficiency. They term this two-way convergence: *NeuroAI*. We fully embrace this new vision and extend it to also include hardware fabric in a three-way convergence of Neuroscience, AI, and Computer Systems with the goal to build next-generation "*NeuroAI Computing Systems*."

A. Cortical Columns and Reference Frames

Our NeuroAI approach is influenced by the work of Jeff Hawkins and Numenta [3], which was based on the discovery by Vernon Mountcastle of the columnar organization of the neocortex [9] and the theoretical neuroscience work on reference frames [10] [11]. Hawkins' Thousand Brain Theory [4] suggests that *Cortical Columns* (CC) are the fundamental neocortical units that embody intelligence. CCs function as *Reference Frames* (RFs) that provide a sensorimotor modeling system. RFs record information through movement (sensorimotor learning), as the associated sensing agent moves through an environment, which could be physical or conceptual.

In Hawkins' model, multiple RFs targeting different sensory modalities can interact with each other and seek consensus on the output through voting within and between sensory modalities. In contrast to current DNNs that separate training and inference, CCs that store and process information in RFs can support real-time inference, as well as online, concurrent and continuous learning, and dynamically adapt to changes in sensory input from the environment. The continuous feedback loop of predict-sense-update within a reference frame effectively introduces an additional dimension of "memory" that remembers past observations and patterns. This "sequential" (or recurrent) behavior is missing in all feed-forward neural networks, e.g. CNNs and DNNs. We believe using reference frames with their associated world models, we can achieve continuous online-adaptive learning (COAL) capability.

B. Hexagonal Coordinate Systems

1) *Neuroscience Background and Inspiration*: The concept of navigation in mammals encompasses a wide range of sensory processing. The brain consolidates the perception of self-location, distance and direction with a topographically organized mapping of spatial data within grid cells in the medial entorhinal cortex (MEC) [6]. Structurally, the entorhinal cortex is located at the interface between the neocortex and the hippocampus. MEC contributes to the formation and reinforcement of allocentric spatial memories, while the lateral EC (entorhinal cortex) carries context information and enables processing of temporal events [12]–[14]. This division of labor supports activities such as path-finding, landmark anchoring, and reference frame translation.

Grid Cells function as a type of spatial correlation that establishes the sense of distance and direction over time. They

are discharged at multiple locations repetitively to symbolically explore the mental environmental map, reinforcing and memorizing the image. They also receive feedback from the hippocampus through the subiculum and update themselves routinely to integrate positional and directional information [15]. Deeper inspection of firing rates frequently yields a two-dimensional spatial autocorrelation function that exhibits distinct six-fold (60°) symmetric peaks [16].

Grid cells operate like an embedded coordinate system that provides a repeating lattice of activities to allow animals to estimate their relative positions with external cues. The discrete grid cell modules align with the environmental geometry by spontaneous and individualized updates corresponding to positional, orientational, and boundary information [17] [18] and interact with hippocampal *Place Cells* during remapping [19]. Together, they form a flexible and stable frame of reference that adapts to new environments while preserving previously learned maps. In humans, evidence from functional magnetic resonance imaging (fMRI) also indicates a hexadirectional (six-fold rotational symmetric) firing patterns of grid cells in EC during the navigation of a virtual environment [20] [21]. In addition to spatial cognition, a hexagonal grid-shaped signal is also capable of coding abstract knowledge and experiences [22].

2) *Current Hexagonal Coordinate Systems*: Hexagonal spatial data require a coordinate system to numerically represent locations within the grid. The H3Geo framework [23] (see Figure 1), which represents the Earth's surface using a hierarchy of mainly hexagonal cells projected onto the faces of an icosahedron, employs a (i, j, k) coordinate system defined along three 120° axes, normalized so that all coordinates are positive and one coordinate is always zero. With the (i, j, k) coordinate system, translation by a vector requires normalization to preserve these constraints, and the lattice exhibits symmetry only every 120° . Each cell is assigned a unique integer index that encodes its hierarchical resolution and location within each parent cell, with lookup tables used to convert between index values and the base coordinates (i, j, k) . Although this indexing system supports the transition between resolutions, it introduces additional computational overhead for normalization and index-to-coordinate conversion.

An alternative is the isometric cubic coordinate system, which is what NeuroHex is based on. (see Fig. 1) It is a fully symmetric representation of the hexagonal grid [24]. In this model, each hex cell is assigned a triplet (q, r, s) that resembles cubic coordinates on the plane $x + y + z = 0$ in a 3D Cartesian space. This formulation preserves all six directions of movement and achieves full 60° of symmetry. Such cubic coordinates require no normalization after translation operations, since addition and subtraction of coordinate triplets automatically preserve the zero-sum invariant. The redundant third axis for our 2D hexagonal grid enables highly efficient algorithms to calculate movement, rotation, and distance.

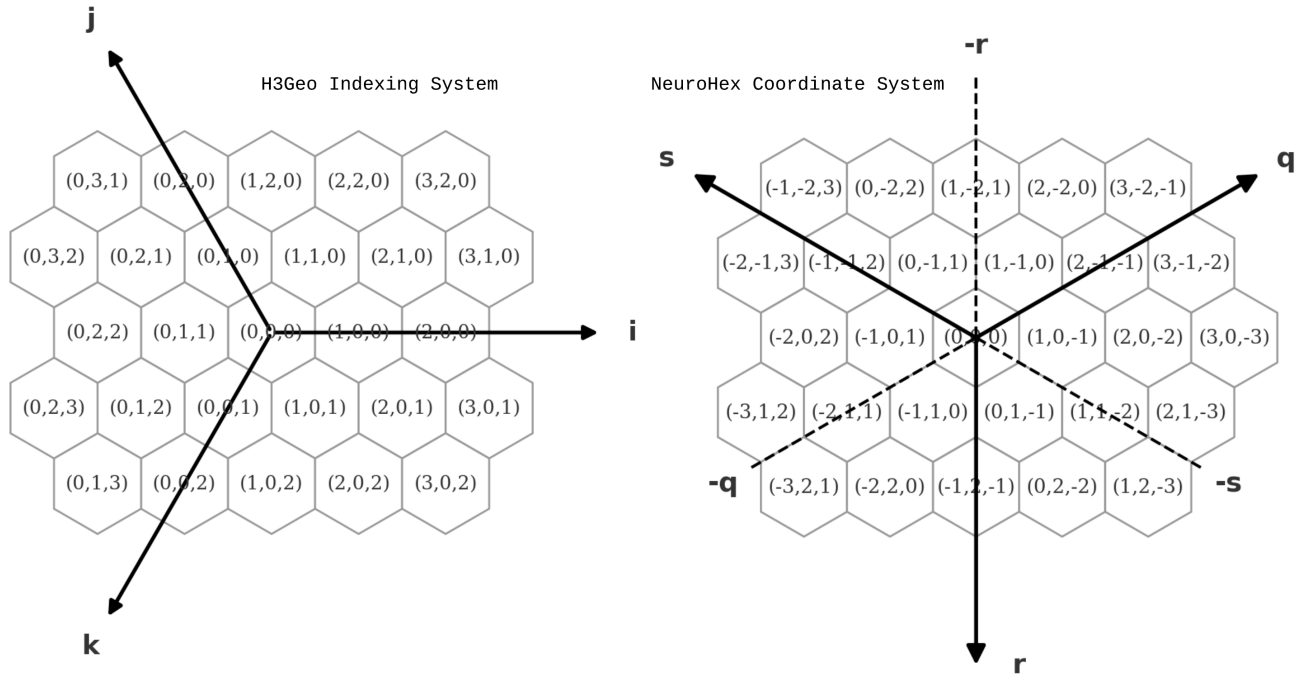


Fig. 1: Hexagonal Coordinate Systems: H3Geo (left) with positive coordinates (i,j,k) [23]; our NeuroHex (right) with positive and negative coordinates (q,r,s).

III. WORKING MODEL OF THE BRAIN AND MOTIVATION FOR NEUROHEX

This section introduces a high-level working model of the brain to provide context and motivation for NeuroHex. Our working model is based on the interpretation of neuroscience through the lens of trial and error simulation. We used a mouse in a maze (both constrained with walls and open fields) to develop an architecture where the mouse can learn, recognize, and navigate many mazes. We know that our brain model and reference frame (RF) organization are not exactly equivalent to biology, but we found them to be a useful abstraction.

Neuroscience research suggests that mammals have the tendency to form cognitive maps of what is present in the environment and where the elements of the environment are located in relation to each other. [25], [26]. Figure 2 shows a high-level brain working model. The ability to sense our orientation and movement in space helps us maintain a sense of place and is encoded in the grid cells. Sensory input, like vision, comes into the brain from an egocentric perspective. The sensory input needs to be processed into higher level features and moved into an allocentric coordinate system. Without moving into an allocentric coordinate system we would be overwhelmed as we moved through an environment, and everything constantly shifts in the egocentric perspective. Features and their relative positions are used to reference long-term memory and recognize known objects that can be put into a world model working memory. Objects in the world model and their relative positions can also be used to reference long-

term memory to recognize higher-level objects. For learning, new objects can be constructed in the working memory and can ultimately be used to create new long-term memories.

Long-term memory is a collection of reference frames. The heart of a reference frame is an associative structure that maintains a map of features and their relative positions, see Figure 3. The goal is to create a specific mental map of an environment, which is a hierarchical structure of reference frames, creating a 'what-where' representation of the physical world. Long-term memory is accessed by sending it a set of features with a relative (delta) position between them.

The reference frames in long-term memory attempt to match the set of features and their relative positions. To do this, each reference frame keeps track of a possible translation from the observed frame of reference to their local frame of reference. With multiple observations, based on possible translation matches, a reference frame gains confidence. The reference frames vote among themselves on the basis of their local confidence. An object is "recognized" when a reference frame reaches high confidence and responds with its local position translation (scaling, rotation, and position) and confidence.

We first consider the Cartesian coordinate system. If features are only points and we ignore rotation, the computational complexity is low. But to represent the world, we need a richer geometry including: straight lines, bent lines, polygons, and circles. As we extend the richness of the geometry, the computational cost of the matching operations becomes very

B. NeuroHex Ring Encoding, Positional, Angular Arithmetics, and Sign-based Wedge Indexing

The hexagonal canvas can be conceptualized as a central origin surrounded by successive concentric hexagonal rings of adjacent coordinates radiating outward. To effectively calculate displacement and angle within the hex space, we formulate a *Ring Encoding* strategy for NeuroHex to represent the relative position of hex coordinates.

Ring Encoding assigns a *RingIndex*(*RI*) to every ring to identify itself relative to an arbitrary origin, and a *RingSpot*, which indicates the position of the coordinate on the designated ring. We define that the origin always starts at *RI* "0". A full ring can be subdivided into 6 segments with inclusive endpoints that correspond to the 6 wedges of a hexagon. Within the ring segment of a wedge, any *RingSpot* will function as a relative *WedgeRingSpot*, effectively limiting its range from 0 to *RI* if both axis lines are included in a wedge sector. An illustration of the basic components of NeuroHex is shown in Figure 4.

NeuroHex coordinate system, derived from hexadecimal coordinates, possesses some other distinctive structural properties that make it particularly well-suited for geometric computation. The distance between two coordinates can conveniently be computed by taking the maximum of the absolute values of the differences of each cubic coordinate. Any coordinate on a ring is also *RI* away from the origin. See Figure 6 (left). Defined by an origin and six relative wedges, the hexagonal canvas provides a seamless and efficient transition between egocentric and allocentric representations of the 2D world. Each hex coordinate consists of three signed values that uniquely identify the wedge in which they reside. There exist two indexing methods for NeuroHex wedges: **order-based** and **sign-based** indexing. Order-based indexing method labels wedge from 0 to 5 following clockwise order, with the top wedge being index 0 as shown in Figure 4. The second indexing method characterizes them according to the non-negativity of their coordinate components. Starting from the top wedge counting clockwise, the six wedges can be identified by $\{010, 011, 001, 101, 100, 110\}$ in binary by extracting the sign bits of the three coordinate numbers. The boundaries of the wedges are subsequently adjusted as shown in Figure 5. This encoding is called *sign-based wedge indexing*. The redefinition of the wedge space and index eliminates the need to select the ordered wedge index through multiplexing, which is especially frequent within polar-related operations. The merged *q*, *r*, and *s* axis line boundaries do not compromise the accuracy of wedge-based localization. For directional comparison, a coarse partitioning of the space is sufficient, while for operations outputting precise coordinate values, the sign-based wedge index is typically converted back into ordered wedge index, and the coordinates are usually denormalized prior to exact computation.

Directional computation in NeuroHex can be achieved via angular quantization. Rotational references established by the positions of the wedges can be further quantized into arbitrary

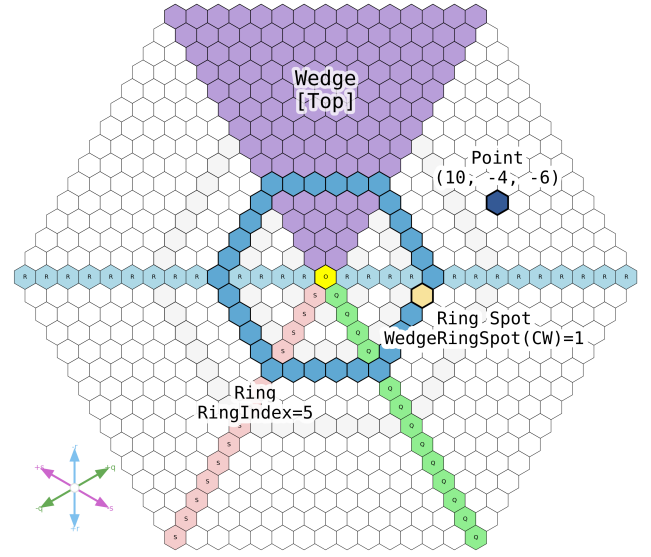


Fig. 4: Foundational Components in the NeuroHex coordinate system: Ordered Wedge, Ring, and Point.

levels that define the resolution of the hex canvas and the geometric operations. Hexagonal rings formed by the coordinates surrounding the origin can be interpreted as anchor structures, with lines extending to the origin that partition each wedge into equal angular regions as seen in Figure 6 (right). For example, with this approach, assuming 4-bit quantization on every wedge, the maximum angular resolution for the entire canvas is $\frac{360}{(6 \cdot 16)} = 3.75^\circ$. Both directional and angular transformations can be mapped to the reference ring to improve hardware performances, analogous to quantization of weights in machine-learning models. Within NeuroHex framework, such conversions are referred to as the normalization and denormalization processes. All geometric operations within the NeuroHex framework can be localized to a single wedge, and most reduce to the comparison of whether one coordinate on a given ring lies farther than another coordinate on a different ring. This constitutes the only operation in the NeuroHex coordinate system that requires multiplication; all other operations can be implemented using only addition and subtraction.

The computational benefits are not limited to the reduced multiplication counts. Divisions are excluded from all geometric operations, while the computation width is also effectively reduced. Assume an 8-bit wide signed NeuroHex canvas where possible coordinate values range from -128 to 127, localized geometric operations need only one additional bit for two's complement addition and subtraction, and at most 18 bits for multiplication. Moreover, the inherent geometric structure of hexagonal grids avoids the use of square roots and trigonometric functions entirely. All wedge-related operations are minimal in complexity and require only 3-bit representations.

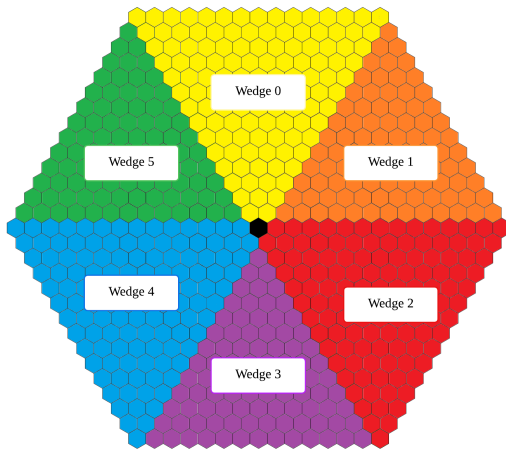


Fig. 5: Wedge Partitioning Based On Sign-based Wedge Indexing

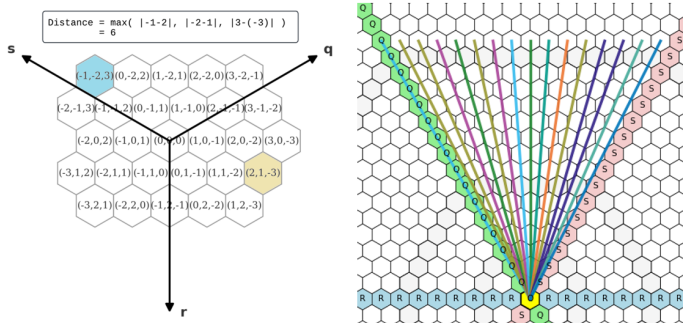


Fig. 6: Distance Computation (left) and Angular Direction Quantization (right) in the NeuroHex coordinate system.

C. NeuroHex Shape Library

With the concepts of location and direction established, we can transform the hex coordinate system into a canvas decorated with "shapes" composed of hex coordinates labeled with various features, similar to rasterization. Drawing shapes in NeuroHex is equivalent to determining whether a given coordinate lies within a defined boundary. The process can be reduced to repeated assessments of a point's position relative to a directed line segment, known as *orientation predicate* in computational geometry.

We classify the NeuroHex shapes into three categories: *Foundational*, *Simple*, and *Complex*, illustrated in Figure 7.

1) *Foundational Shapes*: A foundational shape is characterized by a hex coordinate, one magnitude, and two quantized angles. The angles share a common vertex and are labeled to ensure that the latter angle is always situated in the clockwise direction relative to the former angle, allowing wrap-arounds at the edge of a full rotation.

With a reference point, a magnitude, and two angles, it is possible to construct:

- a *point* by setting angles and magnitude to null.
- a *ray* by setting one of the angles to null.

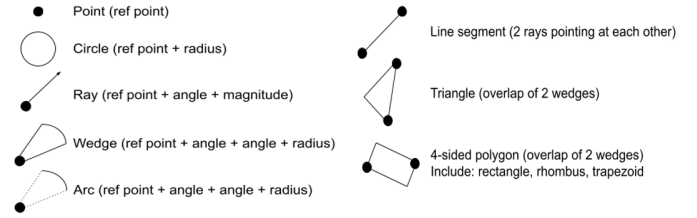


Fig. 7: Foundational Shapes (left) with single reference point, and Simple Shapes involving two reference points (right).

- a *wedge* by setting all four properties to not null.

The catalog of foundational shapes is not restricted to these examples and can be customized as needed by altering and configuring what the null mask symbolizes.

2) *Simple Shapes*: A simple shape consists of the intersection of two foundational shapes. That is, a coordinate is considered to be enclosed in a simple shape if and only if it is bounded by both of the foundational shapes. With such an intersection, we can obtain common shapes used for maps and computer graphics, such as:

- a *line segment* by rendering two rays facing each other.
- a *triangle* by arranging two wedges with acute angles together such that they coincide along a common edge.
- a *quadrilateral* like rectangle, rhombus, and trapezoid by positioning two wedges in opposing orientation.

A wide variety of other shapes can be built through the juxtaposition of foundational shapes.

3) *Complex Shapes*: By integrating simple Boolean logic into the shape-generation process, we can construct various geometries through additive overlap (*AO*) and subtractive overlap (*SO*). These two operations serve as complements: *SO* isolates coordinates contained by one shape but excluded from the other, whereas *AO* selects coordinates jointly bounded by both shapes. Some of the shapes generated include donuts (subtractive of two circles) and tear drops (additive of two circles). Using more than two foundational shapes further expands the geometric shapes that can be represented.

Even with support for rich geometric shapes, the point-in-polygon test in NeuroHex requires only polar angle computations and logic operations and incurs substantially less complex computation than conventional computational geometry.

D. NeuroHex Translation, Rotation, and Scaling of Shapes

Figure 8 shows the efficient shape operations in NeuroHex.

Translation: In addition to allowing for easy origin translation, NeuroHex also enables efficient translation of *magnitude* and quantized angle input to cubic coordinates. *RingIndex* is equal to $(\text{magnitude} + 1)$. The quantized angle allows for straightforward computation of direction: let the maximum per-wedge quantization level be a power of 2, encoded using n bits in binary, the higher order $(n + 2)$ to n bits of the angle encoding can be used to index the 6 wedges, while the lower n bits represent the wedge-local angular quantization. Determining the exact *WedgeRingSpot* requires a

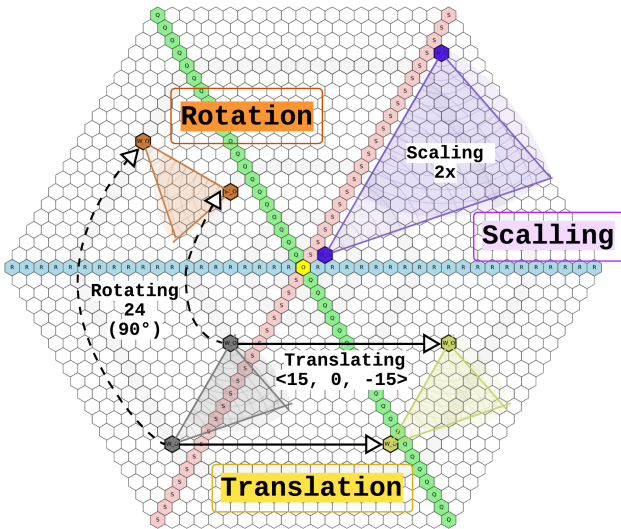


Fig. 8: Examples of Translation, Rotation, and Scaling of shapes in the NeuroHex coordinate system.

conversion from relative quantization to integer length, which involves a single instance of logical shift and multiplication. Once *RingIndex*, *WedgeIndex*, and *WedgeRingSpot* have been derived, the target cubic coordinate resolution becomes trivial, since only the multiplexing selection of directions from the six wedges and potential sign reversal is needed without further arithmetic. This is illustrated in Figure 8.

Rotation: The rotation operation likewise benefits from the lightweight translation of polar inputs, since it merely appends the rotational offset to the quantized angle. As a result, the dominant computation cost of rotating a point around an arbitrary reference within the NeuroHex framework would involve 3-4 additions and a single multiplication. As previously described, NeuroHex builds simple and complex shapes from foundational shapes composed of point, angle, and magnitude. Rotating a shape around an arbitrary point is structurally equivalent to rotating the reference points and linearly translating the angles within the shape. An example of point rotation is shown in Figure 8. In comparison, an analogous operation in Euclidean space is considerably more complex, requiring rotation matrices with trigonometry.

Scaling: Shape up-scaling in NeuroHex can be achieved through logical shifts of coordinates and magnitudes if the scaling factor is power of 2. Down-scaling requires an extra step of normalization (details covered in the next section) to maintain the constraints of the coordinate system.

E. Variable Resolution

To gracefully handle working with objects of different sizes, we support different resolutions in NeuroHex. In a cubic-coordinate representation, multiplying the grid size by a factor k is equivalent to tiling the plane with larger hexagonal cells whose centers lie at coordinates that are integer multiples of k . These centers maintain the exact geometric structure of

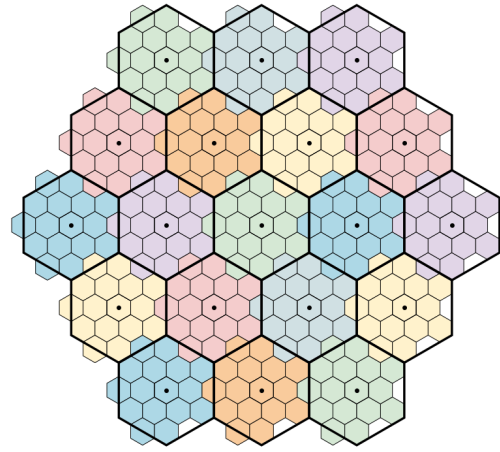


Fig. 9: Variable Resolutions of Hexagonal Tiles in NeuroHex.

the original lattice, which is important to reuse the existing arithmetic library. Limiting k to a power of 2 reduces the implementation cost, as division and remainder extraction can be performed with bit shifts and bit masks in two's-complement. Thus, to change to a coarser resolution, we must map a coordinate to the index of the coarser tile that contains it, as illustrated in Figure 9.

Dividing and truncating each coordinate by k for q , r , and s independently will lead to violations of the zero-sum constraint of the cubic coordinate system. After dividing each component by k , the sum of the 3 remainders can be represented as $n \cdot k$, where $n \in \{0, 1, 2\}$. To produce both valid cubic coordinate centers and tiles of equal size, we increment n of the divided components that produce the n largest remainder(s).

F. Computational Efficiency of NeuroHex Coordinate System

Table I compares the complexity of the computation for performing key geometric operations in the NeuroHex vs. Cartesian coordinate system and rasterization algorithm. The following is a major advantage: operations in the NeuroHex coordinate system avoid computationally intensive calculations. The required number of multiplications and divisions is limited and no trigonometric functions, that are often orders of magnitude more costly, are needed. For a world model defined within an N -bit resolution and range, NeuroHex requires at most $(N + B)$ bits to properly represent all features, since none exceed eight times the largest coordinate magnitude. In contrast, Cartesian coordinates inherently require up to $2N$ bits due to squaring. NeuroHex mitigates these complexities by employing much simpler ALU operations. Compared with rasterization rendering techniques commonly used in modern GPUs, NeuroHex eliminates the need for matrix arithmetic and edge functions, delivering superior efficiency with convenient and streamlined orientation determination. A much more in-depth analysis of computational complexity is presented in the next section. For each of the operations in Table I, we quantitatively show the significant computational efficiency of spatial world models based on NeuroHex.

TABLE I: Comparison of computation complexities between NeuroHex system and Cartesian computational geometry.

Operation	NeuroHex Coordinate System	Two-Dimensional Computational Geometry
Distance ^(a) ①	$\max(q - q' , r - r' , s - s')$	$\sqrt{(x - x')^2 + (y - y')^2}$
Radial Distance	$\max(q , r , s)$	$\sqrt{x^2 + y^2}$
Denormalization	$(R_{ws} \cdot R_i^{(b)}) \gg B^{(c)}$	$\phi_{ref1} \cdot \phi_{ref2} \gg B$
Polar Angle ($\phi_{point}^{(d)}$) ②	$R_{ws} + W_i^{(e)} R_i$	$\text{atan2}(x, y)$
Sign Decoding [*] ③	$(q, r, s) = (s_q q , s_r r , s_s s)$ $\mathcal{S}(\text{Wedge}) = (s_q, s_r, s_s)$ $s_q, s_r, s_s \in \{-1, 0, +1\}$	$(x, y) = (s_x x , s_y y)$ $\mathcal{S}(\text{Quadrant}) = (s_x, s_y)$ $s_x, s_y \in \{-1, 0, +1\}$
Orientation Predicate	Obtain ϕ_1 and ϕ_2 via ② and determine whether $\phi_1 < \phi_2$	$\det \begin{pmatrix} x_1 - x_2 & y_1 - y_2 \\ x - x_2 & y - y_2 \end{pmatrix} > 0$
Point Translation [*]	$(q - \Delta q, r - \Delta r, s - \Delta s)$	$(x - \Delta x, y - \Delta y)$
Point Rotation (CW)	1. Obtain W_p and $R_{ws,p}$ via ③ 2. Obtain W_{rot} and quantized $R_{ws,rot}$ 3. $W_{final} = \begin{cases} W_p + W_{rot} - 6, & W_p + W_{rot} \geq 6 \\ W_p + W_{rot}, & \text{otherwise} \end{cases}$ 4. $R_{ws,final} = R_{ws,p} + \text{denormalized} R_{ws,rot}$. 5. Trim overshoot denormalized wedge ring from $R_{ws,final}$. Increment W_{final} if needed. 5. Assign position and sign for final coordinates.	$x' = x \cos \theta_{rot} - y \sin \theta_{rot}$ $y' = x \sin \theta_{rot} + y \cos \theta_{rot}$
Point-in-Sector Test ④	Perform two ② to obtain polar angles of sector boundaries ϕ_{b1}, ϕ_{b2} and denormalize. Perform one ② to obtain polar angle ϕ_p for the point and check if $\phi_{b1} \leq \phi_p \leq \phi_{b2}$.	$\vec{v} = (x, y)$ $\vec{a} = (\cos \theta_1, \sin \theta_1)$ $\vec{b} = (\cos \theta_2, \sin \theta_2)$ $\text{InSector}(P) \iff \begin{cases} \ \vec{v}\ ^2 \leq R^2 \\ \text{cross}(\vec{a}, \vec{v}) \geq 0 \\ \text{cross}(\vec{v}, \vec{b}) \geq 0 \end{cases}$
Point-in-Triangle Test	Perform two ④ tests, since a triangle is defined by two wedges.	Edge function [28]: $E_{01}(P) = (P_x - V_{0x})(V_{1y} - V_{0y}) - (P_y - V_{0y})(V_{1x} - V_{0x})$ $\text{InTriangle}(P) \iff \begin{cases} E_{AB}(P) \geq 0 \\ E_{BC}(P) \geq 0 \\ E_{CA}(P) \geq 0 \end{cases}$

^(a) Marked operations are native to hexagonal grids. All other operations are developed in this work and are unique to NeuroHex.

^(b) R_i denotes the ring index. The wedge-local ring spot is R_{ws} .

^(c) $B = \log_2(Q)$ is the bit-width of quantization levels, where Q is the number of angular quantization levels per wedge.

^(d) ϕ denotes the quantized clockwise angle in the NeuroHex representation.

^(e) The six wedges of the flat-top hexagonal canvas are indexed by W_i from 0 to 5, beginning at the top wedge and increasing clockwise.

V. COMPARISON OF HARDWARE IMPLEMENTATIONS OF NEUROHEX VS. CARTESIAN SYSTEMS

A. Implementation Strategies for NeuroHex and Cartesian Geometric Operations

1) *Distance*: Due to the nature of cube coordinates, distance calculation within NeuroHex can be simplified to the maximum of the absolute coordinate values, as shown previously. In the Cartesian coordinate system, the accurate distance must be calculated using a fixed-point approximation of the square root of the sum of squared differences.

2) *Polar angle*: The conventional way to determine relative angles formed by coordinate points in the Cartesian coordinate system is usually the arctangent function, but the full precision afforded by the trigonometric function is unnecessary. Instead, we implement a low-cost integer approximation of atan2 that matches the angular granularity provided by the NeuroHex canvas. Given the input x and y coordinates, this method computes a quantized angle bin by quantizing the 360° unit circle into 4° snippets, thus avoiding trigonometry.

With NeuroHex, polar angle mostly refers to the relative position of the point along a hexagonal ring. In subsequent operations where angle inputs are involved, most of the angular

quantities are quantized with respect to rings whose scales are powers of two, which allows them to be mapped directly to the polar-angle representation through the denormalization procedure introduced earlier. The denormalization of degrees is a common recurring method employed throughout the implementations of other geometric operations mentioned in this paper.

3) *Rotation*: NeuroHex offers a major advantage in rotational operations by completely avoiding trigonometries. Through quantization, the angular resolution can be parameterized with respect to the rings scaled by powers of two. In this implementation, we adopt a 4-bit quantization scheme, which corresponds to 16 equal divisions of each wedge and an angular resolution 3.75° . In particular, the same or higher level of angular precision is achievable in NeuroHex with integer arithmetic alone.

Polar domain operations in the Cartesian Coordinate System are bound to floating-point or fixed-point arithmetic. To ensure a fair comparison with NeuroHex, the Cartesian sine and cosine operations are implemented in fixed point with Bhāskara I's sine approximation. The number of fractional bits is selected to align with the angular resolution of Neurohex. The compute kernel applies a standard 2D rotation matrix

to the input coordinates, after which the resulting values are rounded to the nearest integer.

Rotation in NeuroHex follows a fundamentally different approach. The natural wedge-based partition enables the use of compact lookup tables, typically made of 6 or 7 entries to efficiently look up the sign configuration and positional ordering of the q , r , and s cubic coordinates. Within each wedge, one and only one of the absolute values of q , r , and s increases in consecutive units of step from 0 along the clock-wise direction of a hex ring, while one and only one remains constant along a given hex ring and increases radially in unit step from the origin across successive rings. We describe the former as the ring wedge spot numerator (R_{ws}) and the latter as the ring denominator. The positional relationship between the numerator and the denominator across the six wedges is defined as the $\{1st \& 2nd, 3rd \& 1st, 2nd \& 3rd, 1st \& 2nd, 3rd \& 1st, 2nd \& 3rd\}$ elements of the cubic coordinates.

Rotating a point around an arbitrary center in the hexagonal grids is equivalent to translating that point by a specific distance along the hexagonal ring on which it resides. Consequently, the radial distance, or the ring denominator, of the initial and destination points remains unchanged during and after the rotation. For any rotation within a positive 360° range, the operation can be decomposed into the $\{\text{number of whole wedges} + \text{the } R_{ws} \text{ remainder}\}$. In a 4-bit quantization scheme, for example, a rotation of 150° corresponds to 50 quantization units moved along the hex ring that can be expressed as $\lfloor \frac{50}{16} \rfloor = 3$ full wedges rotated with a residual displacement of $50 - 3 \cdot 16 = 2$ quantization units along the ring.

Subsequently, the process of NeuroHex rotation can be generalized into three steps. First, the ordered wedge index of the initial point is added to the number of whole wedges traversed during rotation to obtain the preliminary result wedge. Second, R_{ws} of the initial point is added to the denormalized R_{ws} of the rotational angle remainder, after which it is checked to determine whether the combined value exceeds the amount of a single wedge on the denormalized ring. The result wedge is incremented by 1 if an overflow occurs and reduced by 6 if the traversal extends the unit circle. The excess full wedge portion of the remainder, if present, is also removed. Finally, the new R_{ws} is calculated within the result wedge. The values of q , r , and s and their signs are assigned according to the location of the wedge.

4) *Orientation Predicate*: The orientation predicate refers to the orientation test in computational geometry, which determines whether a point lies on the clockwise or counterclockwise side of a directed line. In the Cartesian coordinate system, it is typically evaluated through the determinant calculated from a square matrix formed by the coordinates of the point of interest and two points from the line.

The test becomes somewhat more complicated in NeuroHex. The origin of the directed line is chosen first as a local reference point, and the coordinate values of the remaining two points are recalculated relative to the new origin. The second point defining the directed line can then be assigned

deterministically to one of the six wedges by inspecting the signs of its coordinates, which is referred to as the line wedge. With the new canvas established, the position of the point of interest relative to the line may be categorized into three distinct scenarios:

- 1) it lies within the two wedges immediately clockwise from the line wedge,
- 2) it lies within the two wedges immediately counterclockwise from the line wedge, or
- 3) it lies in either the same wedge as the line wedge or the opposite wedge.

The first two cases, which together statistically account for $\frac{2}{3}$ of all possible situations, can be easily resolved by comparing the sign-based wedge index of the translated point of interest against a small lookup table. In the last $\frac{1}{3}$ of the situation when the point lies on the same or opposite wedge, the process requires comparisons of denormalized R_{ws} , which constitutes the critical path in the NeuroHex orientation predicate.

5) *Point in Sector Test*: The point in sector (PIS) test determines whether a point lies within a circular sector. A sector is defined by a center and two bounding infinite rays. To perform the test in the Cartesian coordinate system, the two boundary rays are converted into direction vectors with sine and cosine functions and checked against the sign of the 2D cross product formed between the direction vectors and the point vector relative to the sector center.

One of the principal advantages of NeuroHex is that polar operations require substantially less arithmetic computation. For the PIS test, three examinations are needed to determine the result:

- (1) if the point lies in the same wedge as the left boundary ray, its R_{ws} must be greater than or equal to that of the left boundary ray,
- (2) if the point lies in the same wedge as the right boundary ray, its R_{ws} must be smaller than or equal to that of the right boundary ray, and
- (3) the wedge containing the point must be the same as, or sandwiched between, the ray wedges.

The first two evaluations require denormalization of the angular R_{ws} for comparison, whereas the final evaluation only requires proper handling of wrap-around cases. None of these steps involves trigonometry or fixed-point arithmetic.

6) *Point in Triangle Test*: As discussed in the shape library section, a triangle in NeuroHex can be constructed by intersecting two wedges so that one of their edges overlaps. Consequently, the PIT test becomes particularly straightforward, requiring only two instances of the PIS tests. The strategy may also be extended to more general polygons. On the other hand, in Cartesian coordinate systems, the PIT test is typically performed through three edge-function evaluations, one for each side of the triangle.

B. Hardware Integration with Zynq-7000 FPGA

The NeuroHex and Cartesian compute kernels are implemented with Vitis HLS design flow targeting the Xilinx Zynq-7000 FPGA platforms. Each kernel is embedded within a

wrapper that stages input operands from AXI interface into local BRAM buffers, delivers the data to the kernels, stores the output arrays, and writes the results back to the host over AXI. Synthesis settings are chosen to minimize DSP usage in favor of fabric-only implementation. Pipelining is also deliberately disabled to maintain a streamlined and traceable implementation that aligns more closely with the extracted arithmetic operation counts of the geometric functions.

Once synthesized, the compute kernels are packaged in Vitis and exported as IP cores for Vivado. These IPs are instantiated within a Vivado block diagram and connected to the Zynq processing system through AXI interconnects, after which the complete system is implemented and a bitstream is generated. The hardware configuration is then exported as an .xsa file that is used to create a platform project in Vitis and build the application project. The host executable uses DMA to access the assigned AXI address space, programs the FPGA, transfers the kernel input, and invokes the kernel wrapper. During deployment, the target board is booted from an SD card containing a Petalinux system image, a bitstream file, and the executable. At runtime, the FPGA is first programmed with the generated bitstream before executing the host application to launch the kernels.

C. Discussion on Hardware Implementation Results

Table II presents the DSP-free synthesis results generated in Vitis HLS. Table III reports the post-implementation resource utilization and power estimate from Vivado. The breakdown of the arithmetic operations invoked by each kernel is provided in Table IV. In general, for geometric shapes that can be localized to wedge/ring coordinates and evaluated through low-width ratio-based comparisons, NeuroHex is superior to Euclidean arithmetic by a huge margin, yielding approximately 2x to 12x resource reduction than their Cartesian counterparts for distance, polar angle, rotation, and PIS testing. These characteristics make NeuroHex well-suited for reference frames, as rotational and polar operations are prevalent, and offer substantial resource savings for hardware implementation. For tasks that can be reduced naturally to one or more signed 2D determinants, Cartesian arithmetic is more advantageous. However, even for operations such as the orientation predicate, NeuroHex is more efficient in $\frac{2}{3}$ of the cases, since the early clockwise and counter-clockwise classifications can often be resolved after only a few comparisons. NeuroHex can also be aggressively scaled down to a lower bitwidth as a result of wedge-based arithmetic, but this implementation keeps data types to int8, int16, and int32 for the sake of universality.

It is also important to note that the additional coordinate in the NeuroHex coordinates increases memory traffic and demands more interconnects and storage, which is reflected in the comparatively heavier fabric utilization of operations that require higher counts of coordinates as input. One practical mitigation is to transfer only two hex coordinates as input and perform two additional subtraction/additions to reconstruct the third coordinates when needed. More broadly speaking, if storage is strategically planned to contain only the signed

Kernel	FF	LUT
NeuroHex Distance	50	240
Cartesian Distance	104	885
NeuroHex Polar Angle	53	352
Cartesian Polar Angle	821	1419
NeuroHex Rotate	142	660
Cartesian Rotate	1057	4797
NeuroHex Orientation Predicate	154	775
Cartesian Orientation Predicate	53	392
NeuroHex PIS Test	53	526
Cartesian PIS Test	1729	6226
NeuroHex PIT Test	184	1019
Cartesian PIT Test	35	933

TABLE II: Post-synthesis Resource Utilization of NeuroHex and Cartesian Compute Kernels from Vitis suggests that the NeuroHex implementation is most competent for primitives dealing with angle displacement, averaging 4 to 10 times less FPGA fabric usage. On the other hand, Cartesian kernels are more cost-efficient for operations that exploit edge functions, with a slight lead on PIT test and 2x resource efficiency for orientation predicate.

ring wedge spot numerator and ring denominator, the compute kernels will become even more simplified and memory traffic will reduce significantly.

VI. EXPERIMENTATION WITH REAL-WORLD SPATIAL DATA

For NeuroHex to support spatial data in the real world at scale, we develop a spatial data processing tool that transforms OpenStreetMap (OSM) [29] data into our NeuroHex framework, analogous to the Sensory Processing Pipeline shown in Figure 2.

We use this OSM-to-NeuroHex conversion tool (*OSM2Hex*) to better understand the efficiency of NeuroHex in encoding real-world information. Looking at real spatial data also helps us to scope the needed sizing of working memory and reference frames in long-term memory going forward. Figure 10 shows the complete end-to-end processing pipeline of the OSM2Hex conversion tools flow.

This section evaluates NeuroHex using OSM2Hex and geographic data for the Pittsburgh metropolitan area and a higher-resolution analysis of the Oakland area (containing Carnegie Mellon University and University Of Pittsburgh campuses). These experiments allow us to empirically determine parameters for reference frame size, object limits, geometric simplification thresholds, and multi-scale resolution behavior.

A. From Raw OSM to NeuroHex Spatial Abstraction

We begin with a full extraction of OpenStreetMap (OSM) data covering the Pittsburgh metropolitan area (Figure 11.A). To support scalable real-world spatial abstraction, we use OSM2Hex conversion tool to transform raw OSM geometries into a compact, navigation-relevant NeuroHex world model. Given a user-defined geographic bounding box, OSM2Hex

Name	Slice LUTs	Slice Registers	F7 Muxes	F8 Muxes	Slice	LUT as Logic	DSP	Power (W)
NeuroHex Distance	28	50	0	0	20	28	0	0.001
Cartesian Distance	191	91	0	0	96	191	0	0.004
NeuroHex Polar Angle	54	53	0	0	0	0	0	0.001
Cartesian Polar Angle	351	475	0	0	0	0	0	0.003
NeuroHex Rotation	143	84	0	0	0	0	0	0.001
Cartesian Rotation	830	465	0	0	0	0	0	0.001
NeuroHex Ori Pred	313	148	0	0	113	313	0	0.003
Cartesian Ori Pred	80	53	0	0	61	80	0	0.001
NeuroHex PIS Test	35	45	0	0	22	35	2	0.001
Cartesian PIS Test	1270	778	0	0	425	1266	0	0.001
NeuroHex PIT Test	141	80	0	0	99	141	4	0.001
Cartesian PIT Test	177	35	0	0	142	177	0	0.003

TABLE III: Post-implementation Resource Utilization and Power Report from Vivado displays similar trends to those shown in Table 2, taking into account the added usage of DSP. The power report indicates that NeuroHex outperforms Cartesian with a steady 3x-4x less power consumption in most cases but falls behind on orientation predicate.

parses OSM XML data into primitive geometric candidates (polylines, polygons, and nodes) and groups them by semantic class using OSM tags (e.g., rivers, transportation corridors, buildings, parks, and water features).

Each feature class is processed using a parameterized geometric simplification pipeline. Linear features are simplified using Douglas–Peucker reduction [30] to preserve global topology while reducing vertex density, followed by Chaikin smoothing [31] to improve perceptual continuity. Polygonal features undergo area-based filtering, vertex capping, and boundary smoothing to remove low-impact structures while retaining salient landmarks. The features are further filtered according to relevance. The remaining geometries are replaced with primitive mosaics compatible with NeuroHex composed of straight-line and arc segments and a small palette of simple polygonal shapes. The aggressiveness of filtering and primitive substitution is controlled by a resolution policy enabling adaptive trade-offs between abstraction and detail.

The raw OSM extraction for the Pittsburgh metropolitan area contains approximately 27,000 road polylines, 18,500 building polygons, and 2,300 park and water features, totaling roughly 45,000 geometries. After normalization, feature grouping, and geometric preprocessing using Douglas–Peucker simplification [30] and Chaikin filtering [31], average polyline vertex counts are reduced by approximately 82%, and polygon vertex counts are reduced from an average of 95 to 21. Features below an area threshold of $2.5 \times 10^{-6} \text{ deg}^2$ (e.g., small sheds, garages, and interior courtyards) are completely removed. Following preprocessing, the dataset is reduced to approximately 11,200 meaningful features, representing a 75% reduction relative to the original OSM geometry count.

B. Relevance-Based Feature Prioritization

To further compress the data set while maintaining a meaningful navigation structure, we apply a hierarchical three-stage filtering process. All line-based features are converted into NeuroHex-compatible straight-line and arc primitives:

- 1) *Identity Features*: Primary rivers (Allegheny, Monongahela, Ohio) are merged from 85 polyline fragments into

14 stylized splines, representing an 84% reduction.

- 2) *Structural Features*: Major highways and arterial roads are filtered and simplified from 9,400 polyline segments to 720 primitives (92% reduction).
- 3) *Contextual Features*: Building footprints and landmarks are reduced from 18,500 polygons to 1,200 composite structures (93% reduction).

This process intentionally preserves the features humans rely on for orientation—rivers, major corridors, and landmark clusters—while removing redundant or low-impact geometries. After filtering, the prioritized navigational structure consists of 1,934 key objects (14 identity features, 720 structural features, and 1,200 contextual features), a reduction of roughly 96% relative to the original 45,000 raw OSM geometries.

C. Shape Abstraction Using NeuroHex Primitive Mosaics

Each object can be represented as a mosaic of NeuroHex simple shapes selected from a small palette: rectangles, circles, and triangles. Each polygonal object is replaced by 2 to 8 primitives, depending on the area and perceptual relevance. On average: 1) Parks use 12 to 20 primitives; 2) Building clusters use 3 to 5 rectangles or triangles; and 3) Landmarks use 4 to 9 blended primitives. Across the filtered data set of 1,934 objects, mosaic abstraction resulted in approximately 12,000 total NeuroHex simple shapes, corresponding to another 70% reduction in geometric complexity relative to raw polygon and vertex counts. The resultant abstraction is shown in Figure 11.B.

The 1,934 objects on the metro scale exceed the number of objects humans can maintain simultaneously. Cognitive studies suggest that people manage mental maps containing dozens to hundreds of landmarks [32]. This motivates dividing the metro region into hierarchical reference frames. This is something people do naturally. The Pittsburgh metro area consists of multiple townships, and the City of Pittsburgh itself contains more than 90 neighborhoods. Strong evidence that people logically divide their world down to units they can hold in reference frames and their working memory.

Kernel	Type	Add		Sub		Mul		Div		Cmp		Sqrt	
		NH	C	NH	C	NH	C	NH	C	NH	C	NH	C
op_distance	combined	0	2	6	2	0	4	0	0	8	0	0	1
	16-bit	0	0	6	2	0	2	0	0	8	0	0	1
	32-bit	0	1	0	0	0	2	0	0	0	0	0	0
	10-bit fixed	0	1	0	0	0	0	0	0	0	0	0	0
op_polar_angle	combined	1	6	2	6	1	4	0	4	7	14	0	0
	8-bit	1	0	2	0	1	0	0	0	7	0	0	0
	16-bit	0	6	0	6	0	4	0	4	0	14	0	0
op_rotate	combined	3	5	10	14	1	6	0	2	10	12	0	0
	8-bit	1	0	10	0	1	0	0	0	10	0	0	0
	16-bit	2	2	0	9	0	2	0	0	0	10	0	0
	16-bit fixed	0	0	0	2	0	0	0	2	0	0	0	0
	28-bit fixed	0	3	0	3	0	4	0	0	0	2	0	0
op_ori_pred	combined	0	0	13	5	2	4	0	0	24	2	0	0
	8-bit	0	0	0	0	2	0	0	0	4	0	0	0
	16-bit	0	0	13	4	0	2	0	0	20	0	0	0
	32-bit	0	0	0	1	0	2	0	0	0	2	0	0
op_pis	combined	0	5	6	26	2	13	0	4	13	28	0	0
	8-bit	0	0	0	0	2	1	0	0	2	0	0	0
	16-bit	0	4	6	20	0	6	0	0	11	20	0	0
	32-bit int	0	1	0	0	0	2	0	0	0	2	0	0
	16-bit fixed	0	0	0	4	0	0	0	4	0	0	0	0
	28-bit fixed	0	0	0	2	0	4	0	0	0	6	0	0
op_pit	combined	0	0	12	15	4	6	0	0	26	3	0	0
	8-bit	0	0	0	0	4	0	0	0	4	0	0	0
	16-bit	0	0	12	12	0	6	0	0	22	0	0	0
	32-bit int	0	0	0	3	0	0	0	0	0	3	0	0

TABLE IV: Arithmetic Operation Counts for NeuroHex (NH) and Cartesian (C) Compute Kernels With 4-bit Quantization. The algorithmic structure of geometric operations is consistent with the observed utilization of resources. NeuroHex excels at converting geometric problems into integer selection, wedge localization, and bounded comparison.

D. High-Resolution Example: Pittsburgh’s Oakland Area

NeuroHex naturally supports zooming adaptive spatial resolutions. To evaluate fine-grained behavior in a dense urban environment, we apply the OSM2Hex conversion tool to the Oakland area of Pittsburgh (Figure 11.C). This area contains two university campuses (Carnegie Mellon University and the University of Pittsburgh), professional, retail, residential, and green-spaces representative of high-density urban subregions. Fine-grained navigable information for a region like this can provide insight to explore how spatial data can be organized into reference frames and working memory.

We use the same algorithms to reduce the complexity of the OSM data for this subregion, but adapt the relevance to preserve more information as we zoom in. The metro-scale policy retains only identity and structural features, yielding a sparse corridor-level abstraction, whereas the zoom-scale policy preserves dense contextual features such as individual buildings and paths, resulting in a several-fold increase in feature and primitive density over the same geographic area. The OSM2Hex conversion tool ultimately converts all spatial

data down to simple NeuroHex shapes. For this subregion, the polyline data (roads, paths, rivers) can be represented with 5,599 NeuroHex primitives (line segments, and arcs). The polygon data (parks, buildings, administrative boundary) consist of 4,612 polygons. The geometry of parks on average takes approximately 2.7 simple NeuroHex shapes. The geometry of many buildings can be captured with a single quadrilateral NeuroHex shape. Some complex buildings may require tens of simple NeuroHex shapes. With our current fidelity, we average 2 simple NeuroHex shapes per building. Overall, we used 9,237 NeuroHex shapes to represent the polygon data. Overall, this large complex urban area shown in Figure 11.C can be encoded in less than 15,000 simple NeuroHex shapes.

E. Summary of Applying NeuroHex on Real World Data Sets

For Pittsburgh city and Oakland neighborhood, OSM2Hex conversion tool successfully reduces real-world spatial data through successive stages of geometric simplification, relevance filtering, primitive substitution down to NeuroHex simple shapes. For the metro-scale, we see over two orders

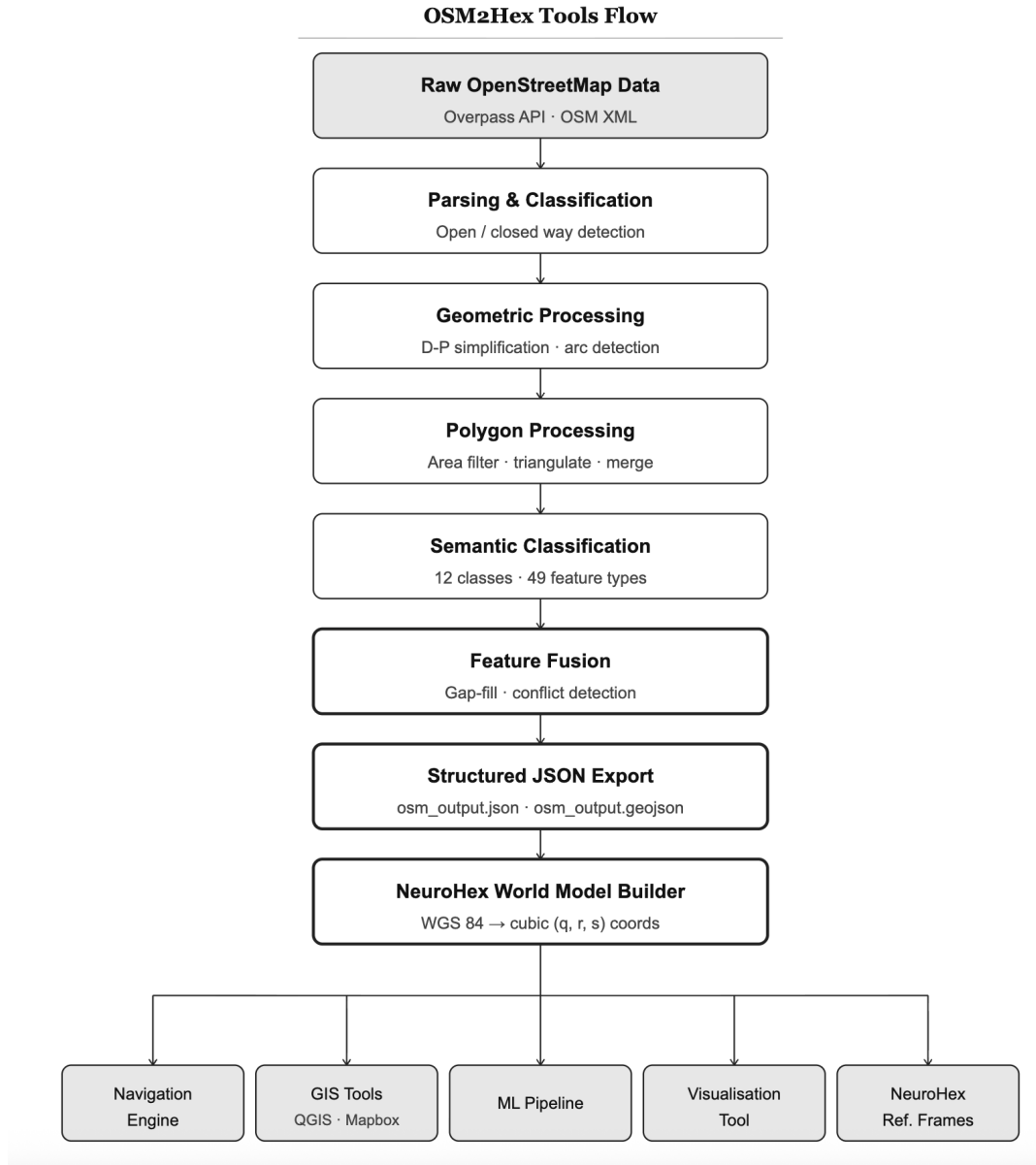


Fig. 10: Complete end-to-end data flow of the OSM2Hex system. Raw OpenStreetMap data is acquired via the Overpass API, parsed, classified, geometrically simplified, triangulated, and merged into NeuroHex-compatible shape primitives. At the feature level primitives are exported as a structured JSON file consumed by the NeuroHex world model builder and downstream applications.

of magnitude in reduction of geometric information down to under 12,000 simple NeuroHex shapes, while still providing information sufficient for general orientation and even navigation between neighborhoods. For the zoomed-in Oakland neighborhood, we preserve all the roads and even pedestrian paths along with nearly all buildings for this subregion, resulting in a detailed navigable map encoded in less than 15,000 simple NeuroHex shapes.

These empirical results provide quantitative guidance for determining spatial coordinate bounds, reference-frame object limits, simplification tolerances, and abstraction granularity.

Together, they demonstrate that large-scale environments can be efficiently encoded using compact and human-interpretable visual primitives suitable for hierarchical spatial memory map and intelligent adaptive navigation within it. The OSM2Hex conversion framework takes advantage of publicly available OSM data sets for the creation of spatial maps of arbitrary geographic areas based on the NeuroHex coordinate system. Using OSM2Hex we can quickly generate the initial world map of a chosen physical area at varying degrees of resolutions. Such world maps can be dynamically updated to adapt to potentially changing environments.

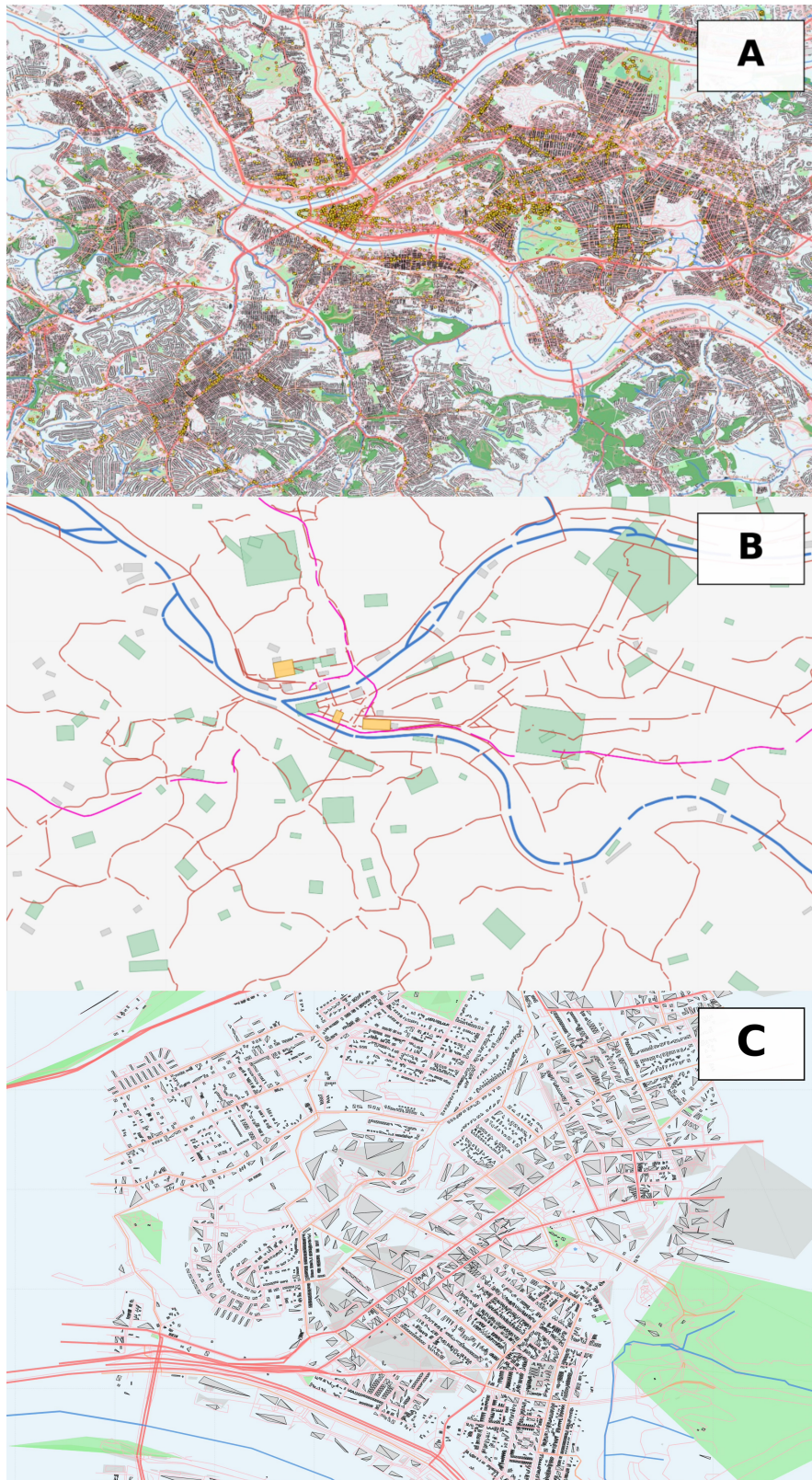


Fig. 11: *OSM2Hex* conversion tool: (A) Raw OpenStreetMap data for the Pittsburgh metro area. (B) Pittsburgh metro area filtered and replaced with NeuroHex primitives. (C) Zoomed in Oakland area with higher resolution NeuroHex primitives.

VII. SUMMARY

We present a novel hexagonal coordinate system that offers a highly efficient substrate for building dynamic world models to enable adaptive spatial reasoning and continuous online-adaptive learning (COAL). Inspired by grid cells in the human brain, *NeuroHex* employs a hexagonal isometric coordinate formulation that provides rotational symmetry and constant-cost translation, rotation, and distance computation. The *NeuroHex* framework incorporates ring indexing, quantized angular encoding, and a hierarchical library of foundational, simple, and complex geometric shape primitives. We also developed *OSM2Hex*, a tool for converting OpenStreetMap (OSM) data sets into *NeuroHex* world models. We show example results using the city of Pittsburgh. The results demonstrate that *NeuroHex* offers a highly computationally efficient platform for online world-model construction and adaptive spatial reasoning. *NeuroHex* platforms can enable edge-native, energy-efficient AI capabilities for autonomous and online-adaptive perception, cognition, and navigation applications.

REFERENCES

- [1] Q. Jacobson, J. Luo, J. Xu, S. Venkatachalam, K. Wang, D. Rong, and P. John Shen, "Neurohex: Highly-efficient hex coordinate system for creating world models to enable adaptive ai," in *2026 Neuro Inspired Computational Elements (NICE)*. IEEE, 2026.
- [2] T. Taniguchi, S. Murata, M. Suzuki, D. Ognibene, P. Lanillos, E. Ugur, L. Jamone, T. Nakamura, A. Cirià, B. Lara *et al.*, "World models and predictive coding for cognitive and developmental robotics: frontiers and challenges," *Advanced Robotics*, vol. 37, no. 13, pp. 780–806, 2023.
- [3] J. Hawkins, S. Ahmad, and Y. Cui, "A theory of how columns in the neocortex enable learning the structure of the world," *Frontiers in neural circuits*, vol. 11, p. 295079, 2017.
- [4] J. Hawkins, *A thousand brains: A new theory of intelligence*. Basic Books, 2021.
- [5] M. Quak, R. E. London, and D. Talsma, "A multisensory perspective of working memory," *Frontiers in human neuroscience*, vol. 9, p. 197, 2015.
- [6] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–806, 2005.
- [7] F. Sargolini, M. Fyhn, T. Hafting, B. L. McNaughton, M. P. Witter, M.-B. Moser, and E. I. Moser, "Conjunctive representation of position, direction, and velocity in entorhinal cortex," *Science*, vol. 312, no. 5774, pp. 758–762, 2006.
- [8] A. Zador, S. Escola, B. Richards, B. Ölveczky, Y. Bengio, K. Boahen, M. Botvinick, D. Chklovskii, A. Churchland, C. Clopath *et al.*, "Catalyzing next-generation artificial intelligence through neuroai," *Nature communications*, vol. 14, no. 1, p. 1597, 2023.
- [9] V. B. Mountcastle, "The columnar organization of the neocortex," *Brain: a journal of neurology*, vol. 120, no. 4, pp. 701–722, 1997.
- [10] L. W. Barsalou, "Perceptual symbol systems," *Behavioral and brain sciences*, vol. 22, no. 4, pp. 577–660, 1999.
- [11] W. Petersen, "Representation of concepts as frames," *Meaning, frames, and conceptual representation*, vol. 2, pp. 43–67, 2015.
- [12] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, "Path integration and the neural basis of the 'cognitive map'," *Nature Reviews Neuroscience*, vol. 7, no. 8, pp. 663–678, 2006.
- [13] T. Solstad, C. N. Boccara, E. Kropff, M.-B. Moser, and E. I. Moser, "Representation of geometric borders in the entorhinal cortex," *Science*, vol. 322, no. 5909, pp. 1865–1868, 2008.
- [14] Ø. A. Høydal, E. R. Skytøen, S. O. Andersson, M.-B. Moser, and E. I. Moser, "Object-vector coding in the medial entorhinal cortex," *Nature*, vol. 568, no. 7752, pp. 400–404, 2019.
- [15] J. S. Taube, "The head direction signal: origins and sensory-motor integration," *Annu. Rev. Neurosci.*, vol. 30, no. 1, pp. 181–207, 2007.
- [16] J. Jacobs, C. T. Weidemann, J. F. Miller, A. Solway, J. F. Burke, X.-X. Wei, N. Suthana, M. R. Sperling, A. D. Sharan, I. Fried *et al.*, "Direct recordings of grid-like neuronal activity in human spatial navigation," *Nature neuroscience*, vol. 16, no. 9, pp. 1188–1190, 2013.
- [17] H. Stensola, T. Stensola, T. Solstad, K. Frøland, M.-B. Moser, and E. I. Moser, "The entorhinal grid map is discretized," *Nature*, vol. 492, no. 7427, pp. 72–78, 2012.
- [18] J. Krupic, M. Bauza, S. Burton, C. Barry, and J. O'Keefe, "Grid cell symmetry is shaped by environmental geometry," *Nature*, vol. 518, no. 7538, pp. 232–235, 2015.
- [19] M. Fyhn, T. Hafting, A. Treves, M.-B. Moser, and E. I. Moser, "Hippocampal remapping and grid realignment in entorhinal cortex," *Nature*, vol. 446, no. 7132, pp. 190–194, 2007.
- [20] C. F. Doeller, C. Barry, and N. Burgess, "Evidence for grid cells in a human memory network," *Nature*, vol. 463, no. 7281, pp. 657–661, 2010.
- [21] S. Maidenbaum, J. Miller, J. M. Stein, and J. Jacobs, "Grid-like hexadirectional modulation of human entorhinal theta oscillations," *Proceedings of the National Academy of Sciences*, vol. 115, no. 42, pp. 10798–10803, 2018.
- [22] A. O. Constantinescu, J. X. O'Reilly, and T. E. Behrens, "Organizing conceptual knowledge in humans with a gridlike code," *Science*, vol. 352, no. 6292, pp. 1464–1468, 2016.
- [23] H3Geo. H3geo coordinate systems. [Online]. Available: <https://h3geo.org/docs/core-library/coordsystems/>
- [24] I. Her, "A symmetrical coordinate frame on the hexagonal grid for computer graphics and vision," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 97720. American Society of Mechanical Engineers, 1992, pp. 187–190.
- [25] E. C. Tolman, "Cognitive maps in rats and men," *Psychological review*, vol. 55, no. 4, p. 189, 1948.
- [26] J. O'Keefe and L. Nadel, *The hippocampus as a cognitive map*. Oxford university press, 1978.
- [27] A. Patel, "Hexagonal grids. red blob games," 2013.
- [28] J. Pineda, "A parallel algorithm for polygon rasterization," in *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, 1988, pp. 17–20.
- [29] OpenStreetMapOrganization. About openstreetmap. [Online]. Available: <https://www.openstreetmap.org/about/>
- [30] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [31] G. M. Chaikin, "An algorithm for high-speed curve generation," *Computer graphics and image processing*, vol. 3, no. 4, pp. 346–349, 1974.
- [32] N. Burgess, E. A. Maguire, and J. O'Keefe, "The human hippocampus and spatial and episodic memory," *Neuron*, vol. 35, no. 4, pp. 625–641, 2002.