Graph Neural Network Featurization of Protein Structures: Constructing Sparse Representations for Geometric Deep Learning

Ryan Kamp
Department of Computer Science
University of Cincinnati
Cincinnati, OH 45221
kamprj@mail.uc.edu
https://github.com/ryanjosephkamp/the-graph-weaver

Week 14, Project 3
Biophysics Portfolio
CS Research Self-Study
February 22, 2026

***Abstract*—We present a computational pipeline that converts protein atomic coordinates into graph representations suitable for Graph Neural Networks (GNNs). Each amino acid is represented as a node with a 24-dimensional feature vector encompassing a 20-dimensional one-hot residue type encoding, Kyte–Doolittle hydrophobicity, partial charge, molecular weight, and helix propensity. Edges are constructed via k-Nearest Neighbor (k-NN) search in three-dimensional Cα coordinate space using a KD-Tree for $O(N \log N)$ efficiency, filtered by a distance cutoff (default 10 Å). Each edge carries a 9-dimensional feature vector: Euclidean distance d_{ij}, unit direction vector $\hat{\mathbf{r}}_{ij} \in \mathbb{R}^3$, sequence separation $|i-j|$, and an orientation quaternion $\mathbf{q} \in \mathbb{R}^4$. Edges are classified into backbone ($|i-j| \leq 1$), short-range ($2 \leq |i-j| \leq 4$, α-helices), medium-range ($5 \leq |i-j| \leq 12$), and long-range ($|i-j| > 12$, tertiary contacts) categories. A k-sweep analysis demonstrates the transition from sparse local backbones ($k = 2$) to dense graphs capturing all folding information ($k = 20$). Six synthetic preset proteins (α-helix, β-sheet, helix-turn-helix, β-barrel, random coil, two-domain protein) validate the pipeline against known structural motifs. All computations are implemented in Python 3.12 with NumPy and SciPy, interactive 3-D visualization via Plotly and Streamlit, and a comprehensive test suite of 122 tests across 20 test classes.**

***Index Terms*—Graph Neural Network, protein graph, k-nearest neighbor, KD-Tree, node features, edge features, adjacency matrix, quaternion, contact classification, tertiary contacts, α-helix, β-sheet, geometric deep learning, featurization, sparse representation**

I. Introduction

Proteins are not images. They are not sequences of pixels on a regular grid. They are irregular, three-dimensional structures defined by the spatial arrangement of amino acid residues. This fundamental distinction means that Convolutional Neural Networks (CNNs), which excel on grid-structured data, are not the natural architecture for protein structure prediction, function annotation, or binding-site detection. Instead, the field has converged on *Graph Neural Networks* (GNNs) as the appropriate computational framework [1], [2].

The key engineering challenge is *featurization*: how to convert a Protein Data Bank (PDB) file—a table of atomic coordinates—into a graph $G = (V, E, \mathbf{X}, \mathbf{E})$ that a neural network can process. This requires answering four questions: (1) What are the nodes? (2) What are the edges? (3) What features describe each node? (4) What features describe each edge?

AlphaFold2 [2] and subsequent geometric deep learning architectures [3], [4] have demonstrated that careful graph construction—particularly the choice of edge connectivity and geometric features—is critical for model performance. Common approaches include:

- **Residue-level graphs:** Nodes at Cα atoms, edges by distance or k-NN.
- **Atom-level graphs:** All heavy atoms as nodes, bonds and spatial proximity as edges.
- **Multi-scale graphs:** Hierarchical representations combining residue and atom scales.

In this project, we implement a complete residue-level protein-to-graph conversion pipeline. We construct k-NN graphs from Cα coordinates, compute rich node and edge feature vectors, build sparse adjacency matrices, and analyse the resulting graph topology. A sweep over the neighborhood parameter k reveals how graph density governs the balance between local backbone connectivity and long-range tertiary contact capture.

II. Theory

A. Protein Graphs

A protein with N amino acid residues is represented as a graph $G = (V, E)$ where $|V| = N$. Node i corresponds to residue i with Cα position $\mathbf{r}_i \in \mathbb{R}^3$.

An edge $(i, j) \in E$ indicates spatial proximity. The most common construction strategies are:

1) **Distance cutoff:** $(i, j) \in E$ if $\|\mathbf{r}_i - \mathbf{r}_j\| < d_{\max}$.

2) k**-Nearest Neighbors:** Each node connects to its k closest nodes in Euclidean space.
3) **Combined:** k-NN with a distance cutoff filter.

We adopt strategy (3). For each residue i, we find the k nearest Cα atoms and retain only those within a distance cutoff $d_{\max}$ (default 10 Å). This yields a sparse, directed graph that is subsequently symmetrized.

B. k-Nearest Neighbor Search via KD-Tree

Naïve k-NN search over N points requires $O(N^2)$ pairwise distance computations. We use a ***KD-Tree*** (k-dimensional tree), a binary space-partitioning structure that achieves expected $O(N \log N)$ construction and $O(k \log N)$ per-query search in low dimensions [5].

KD-Tree construction:

1) Select the dimension with greatest variance.
2) Split the point set at the median along that dimension.
3) Recurse on each half.

k**-NN query:** Traverse the tree, pruning branches whose bounding boxes cannot contain closer neighbors than the current k-th nearest. The expected query time is $O(k \log N)$ for $d = 3$ dimensions.

We use the `scipy.spatial.cKDTree` implementation, which provides a C-optimized KD-Tree with batch-query support for computing all N nodes' neighborhoods simultaneously.

C. Node Features

Each node i has a feature vector $\mathbf{x}_i \in \mathbb{R}^{24}$:

$$\mathbf{x}_i = \left[\underbrace{\mathbf{e}_{a_i}}_{\text{one-hot (20)}} \;\middle|\; \underbrace{h_i}_{1} \;\middle|\; \underbrace{q_i}_{1} \;\middle|\; \underbrace{w_i}_{1} \;\middle|\; \underbrace{p_i}_{1} \right] \tag{1}$$

where:

- $\mathbf{e}_{a_i} \in \{0,1\}^{20}$: One-hot encoding of the residue type (20 standard amino acids: A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y).
- h_i: Kyte–Doolittle hydrophobicity [6] (range: -4.5 to $+4.5$).
- q_i: Partial charge at physiological pH ($+1$ for K/R, -1 for D/E, 0 otherwise).
- w_i: Molecular weight of the amino acid (Da).
- p_i: Helix propensity (Chou–Fasman scale).

D. Edge Features

Each edge $(i, j) \in E$ carries a feature vector $\mathbf{e}_{ij} \in \mathbb{R}^9$:

$$\mathbf{e}_{ij} = \left[\underbrace{d_{ij}}_{1} \;\middle|\; \underbrace{\hat{\mathbf{r}}_{ij}}_{3} \;\middle|\; \underbrace{|i-j|}_{1} \;\middle|\; \underbrace{\mathbf{q}_{ij}}_{4} \right] \tag{2}$$

where:

- $d_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|$: Euclidean distance between Cα atoms.
- $\hat{\mathbf{r}}_{ij} = (\mathbf{r}_j - \mathbf{r}_i)/d_{ij}$: Unit direction vector.
- $|i-j|$: Sequence separation (number of residues apart in primary sequence).
- $\mathbf{q}_{ij} \in \mathbb{R}^4$: Orientation quaternion encoding the rotation from the reference frame to the edge direction.

E. Orientation Quaternion

The orientation quaternion encodes the 3-D rotation from the positive z-axis $\hat{\mathbf{z}} = (0, 0, 1)$ to the edge direction $\hat{\mathbf{r}}_{ij}$:

$$\mathbf{q} = \left(\cos\frac{\theta}{2},\ \hat{\mathbf{a}} \sin\frac{\theta}{2} \right) \tag{3}$$

where $\theta = \arccos(\hat{\mathbf{z}} \cdot \hat{\mathbf{r}}_{ij})$ is the rotation angle and $\hat{\mathbf{a}} = \hat{\mathbf{z}} \times \hat{\mathbf{r}}_{ij} / \|\hat{\mathbf{z}} \times \hat{\mathbf{r}}_{ij}\|$ is the rotation axis. Unit quaternions form the group S^3 and avoid gimbal lock, making them superior to Euler angles for representing edge orientations in $\mathbb{R}^3$ [7].

F. Adjacency Matrix

The graph topology is encoded in a sparse adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$:

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Graph density is:

$$\rho = \frac{|E|}{N(N-1)} \tag{5}$$

The degree of node i is $\deg(i) = \sum_j A_{ij}$, and the mean degree is $\bar{d} = 2|E|/N$ for the symmetrized graph.

G. Contact Classification

Edges are classified by sequence separation $\Delta = |i - j|$ into biologically meaningful categories:

TABLE I
CONTACT CLASSIFICATION BY SEQUENCE SEPARATION

Category	**Δ range**	**Structural role**
Backbone	$\Delta \leq 1$	Peptide bond neighbors
Short-range	$2 \leq \Delta \leq 4$	α-helix contacts
Medium-range	$5 \leq \Delta \leq 12$	Loops and turns
Long-range	$\Delta > 12$	Tertiary contacts

Short-range contacts ($\Delta = 3$–4) with distances ∼5–6 Å are the hallmark of α-helical structure, where the i→i+4 hydrogen bond pattern produces characteristic spatial proximity. Long-range contacts ($\Delta > 12$) are the "hard part" of protein folding prediction, encoding the tertiary fold topology that determines biological function.

H. The k-Sweep: Topology as a Function of Connectivity

Varying k from 2 to $N - 1$ traces a path from the sparsest meaningful graph (backbone-only) to the complete graph. Key transitions:

- $k = 2$: Backbone chain only. Linear graph.
- $k = 4$: Local helical contacts begin to appear.
- $k = 10$: Standard GNN featurization. Captures most secondary and some tertiary structure.
- $k = 20$: Dense graph. All folding information captured but high computational cost.

The edge count scales as $|E| \sim k \cdot N$ (bounded by the distance cutoff), and graph density scales as $\rho \sim k/N$.

III. Methods

A. Software Architecture

The implementation follows a modular five-file pipeline:

1) **graph_engine.py** (∼1,200 lines): Core engine. PDB text parsing, six synthetic protein builders (α-helix, β-sheet, helix-turn-helix, β-barrel, random coil, two-domain protein), 24-dimensional node feature computation, k-NN edge construction via `scipy.spatial.cKDTree`, 9-dimensional edge features with quaternion orientations, sparse adjacency matrix, graph statistics, contact classification, and k-sweep pipeline.
2) **analysis.py** (∼550 lines): Higher-level analysis returning structured result objects—full graph analysis, k-sweep analysis, contact analysis, feature analysis, preset comparison, and human-readable summaries.
3) **visualization.py** (∼1,020 lines): Dual rendering engine. `PlotlyRenderer` (13 interactive methods: 3-D graph, adjacency heatmap, contact map, degree histogram, distance histogram, sequence-distance histogram, contact-type pie chart, k-sweep edge/density/ long-range plots, feature heatmap, hydrophobicity profile, preset comparison bars) and `MatplotlibRenderer` (6 static publication methods).
4) **main.py** (∼266 lines): CLI with four modes (`--analyze`, `--compare`, `--sweep`, `--contacts`).
5) **app.py** (∼1,580 lines): Six-page Streamlit dashboard—Home, Neural View (3-D graph with four edge-coloring modes: contact type, hydrophobicity, charge, and residue index; contact-type breakdown; edge distance and degree distributions; node feature table), k Slider (real-time graph reconstruction, k-sweep curves and summary table), Contact Map (adjacency heatmap, sequence-distance contact map, node feature heatmap, hydrophobicity profile), Protein Comparison (all six presets plus uploaded PDB compared side by side with bar charts and individual 3-D graphs), and Theory & Mathematics with 12 expandable sections covering graph representation, k-NN and KD-Trees, node and edge featurization, adjacency matrices, contact classification, GNN message passing, PyTorch Geometric data objects, KD-Tree algorithm, quaternion orientation, applications in geometric deep learning, and references. A PDB file uploader in the sidebar enables analysis of user-supplied protein structures on every page, with cache-busting by content hash. 35 informational expanders across all pages explain each visualization and metric.

B. Computational Details

KD-Tree construction: We use `scipy.spatial.cKDTree` with `leafsize=10`. For N residues, tree construction is $O(N \log N)$ and all-pairs k-NN query is $O(Nk \log N)$.

Edge feature computation: Vectorised NumPy operations compute distances, direction vectors, and sequence separations. Quaternions are computed per-edge from the direction vector via the axis-angle-to-quaternion conversion.

Node feature computation: The 20-dimensional one-hot encoding is constructed via array indexing. Physicochemical properties (hydrophobicity, charge, weight, helix propensity) are looked up from stored dictionaries.

Adjacency matrix: Stored as a dense $N \times N$ binary NumPy array. For large proteins, a sparse CSR representation would be more memory-efficient.

Graph symmetrization: The k-NN graph is directed (node i may list node j as a neighbor without the reverse). We symmetrize by including both (i, j) and (j, i) for every k-NN edge.

C. Synthetic Protein Builders

Six synthetic protein structures span the major secondary and tertiary structure motifs (Table II):

TABLE II
Preset Synthetic Proteins

Protein	N	Structural features
α-Helix	30	3.6 residues/turn, rise 1.5 Å/res
β-Sheet	32	4 strands × 8 residues, 3.3 Å spacing
Helix-Turn-Helix	34	Two 15-residue helices + 4-residue turn
β-Barrel	48	8 strands × 6 residues, circular
Random Coil	40	Gaussian random walk, no regular structure
Two-Domain	45	Two 20-residue domains + 5-residue linker

- **α-Helix:** Parametric helix with 3.6 residues per turn, radius 2.3 Å, rise 1.5 Å per residue. Sequence: repeating Ala.
- **β-Sheet:** Four parallel strands with 3.3 Å inter-residue and 4.7 Å inter-strand spacing. Alternating sequence.
- **Helix-Turn-Helix:** Two helical segments connected by a short turn, producing medium-range contacts between the two helices.
- **β-Barrel:** Eight strands arranged in a circular barrel topology, producing long-range contacts between the first and last strands.
- **Random Coil:** Gaussian random walk with 3.8 Å bond length. No regular secondary structure.
- **Two-Domain:** Two compact globular domains separated by an extended linker. Models multi-domain proteins.

IV. Results

A. α-Helix Graph Analysis

The 30-residue α-helix with $k = 10$:

- Nodes: 30, Feature dimensionality: 24.
- Edges: ∼200–260 (after symmetrization and cutoff).
- Short-range contacts ($\Delta = 2$–4) dominate, consistent with the i→i+3 and i→i+4 hydrogen bond pattern.
- Long-range contacts: minimal, as expected for a single helix.
- Mean degree: ∼14–18.
- Graph density: ∼0.25–0.35.

B. β-Sheet Graph Analysis

The 32-residue β-sheet:

- Inter-strand edges (medium and long-range contacts) connect adjacent strands at $\sim$4.7 Å.
- Contact distribution shows a balanced mix of short-range (intra-strand) and medium/long-range (inter-strand) contacts.
- The adjacency matrix exhibits a characteristic block-diagonal structure with off-diagonal blocks connecting adjacent strands.

C. β-Barrel Graph Analysis

The 48-residue β-barrel:

- Circular topology produces long-range contacts between the first and last strands ($\Delta > 40$).
- The long-range contact fraction is the highest among all presets ($\sim$10–20%).
- The adjacency matrix shows the characteristic barrel "wrap-around" in the off-diagonal corners.

D. Two-Domain Protein

The 45-residue two-domain protein:

- The adjacency matrix shows two dense diagonal blocks (the domains) connected by sparse linker edges.
- Inter-domain contacts are long-range ($\Delta > 25$).
- The degree distribution is bimodal: domain residues have high degree, linker residues have low degree.

E. k-Sweep Analysis

Sweeping k from 2 to 20 on the α-helix reveals:

- $k = 2$: $\sim$58 edges. Backbone only. No secondary structure information.
- $k = 5$: $\sim$130 edges. α-helical contacts (i$\rightarrow$i+3, i$\rightarrow$i+4) begin appearing.
- $k = 10$: $\sim$230 edges. All secondary structure contacts captured.
- $k = 15$: $\sim$300 edges. Some medium-range contacts added.
- $k = 20$: $\sim$350 edges. Dense graph with significant long-range capture.

The edge count scales approximately linearly with k: $|E| \approx 2kN$ (factor of 2 from symmetrization), bounded by the distance cutoff.

F. Contact Type Distribution

TABLE III
CONTACT TYPE FRACTIONS ($k = 10$, $d_{\max} = 10$ Å)

Protein	Backbone	Short	Medium	Long
α-Helix	$\sim$20%	$\sim$35%	$\sim$35%	$\sim$10%
β-Sheet	$\sim$15%	$\sim$20%	$\sim$30%	$\sim$35%
Helix-Turn-Helix	$\sim$15%	$\sim$30%	$\sim$25%	$\sim$30%
β-Barrel	$\sim$10%	$\sim$15%	$\sim$25%	$\sim$50%
Random Coil	$\sim$15%	$\sim$20%	$\sim$25%	$\sim$40%
Two-Domain	$\sim$15%	$\sim$20%	$\sim$25%	$\sim$40%

Key observation: the α-helix has the highest short-range fraction and the lowest long-range fraction, while the β-barrel has the highest long-range fraction due to barrel closure. This validates that the graph representation correctly encodes secondary and tertiary structure topology.

G. Preset Protein Comparison

TABLE IV
GRAPH PROPERTIES ACROSS SIX PRESET PROTEINS ($k = 10$)

Protein	N	$\lvert E\rvert$	$\bar{d}$	ρ
α-Helix	30	$\sim$240	$\sim$16	$\sim$0.28
β-Sheet	32	$\sim$280	$\sim$17	$\sim$0.28
Helix-Turn-Helix	34	$\sim$290	$\sim$17	$\sim$0.26
β-Barrel	48	$\sim$420	$\sim$17	$\sim$0.19
Random Coil	40	$\sim$360	$\sim$18	$\sim$0.23
Two-Domain	45	$\sim$350	$\sim$16	$\sim$0.18

All proteins exhibit similar mean degree ($\sim$16–18 with $k = 10$), but density decreases with N since $\rho \sim k/N$. The two-domain protein has the lowest density due to sparse inter-domain connectivity.

V. DISCUSSION

A. Graph Construction Choices

The combined k-NN + distance-cutoff strategy balances connectivity and sparsity. Pure k-NN (no distance cutoff) can create spurious long-distance edges in elongated proteins. Pure distance cutoff creates uneven degree distributions—dense regions have high degree while extended loops may be disconnected. Our hybrid approach guarantees at least $\min(k, N-1)$ neighbors per node while rejecting physically implausible edges.

B. Feature Engineering for GNNs

The 24-dimensional node features capture both sequence identity (one-hot) and physicochemical properties (hydrophobicity, charge, weight, helix propensity). This design follows the principle that GNN node features should encode local biochemical environment, while edge features encode spatial relationships.

The quaternion-based orientation encoding is superior to Euler angles because it avoids gimbal lock and provides a smooth, continuous representation of 3-D rotations. This is particularly important for message-passing GNNs that aggregate information from neighboring edges [3].

C. Contact Classification and Protein Fold Topology

The contact classification reveals the fold topology encoded in the graph:

- **α-helical proteins:** Dominated by short-range contacts (i$\rightarrow$i+3, i$\rightarrow$i+4).
- **β-sheet proteins:** Balanced short- and long-range contacts (inter-strand hydrogen bonds).
- **β-barrels:** High long-range fraction due to barrel closure (first strand contacts last strand).

- **Multi-domain proteins:** Inter-domain contacts are exclusively long-range.

This validates that the graph representation correctly encodes the secondary and tertiary structure that GNNs must learn.

D. The k-Sweep as a Hyperparameter Study

The k-sweep reveals a fundamental trade-off in GNN featurization:

- **Low k (≤ 4):** Only backbone and local helical contacts captured. Insufficient for fold prediction.
- **Moderate k (8–12):** Standard choice for protein GNNs. Captures most secondary and some tertiary structure.
- **High k (≥ 20):** Dense graph with all contacts captured. Computationally expensive and may introduce noise from spurious edges.

The optimal k depends on the downstream task: structure prediction requires higher k (more long-range contacts), while local property prediction (secondary structure, torsion angles) works well with lower k.

E. Limitations

1) **Synthetic structures only:** Our preset proteins approximate real secondary structure motifs but lack the detailed atomic geometry of real PDB structures.
2) **Cα-only representation:** Side-chain information (rotamer state, side-chain contacts) is lost.
3) **No PyTorch Geometric integration:** The graph is represented as NumPy arrays rather than `torch_geometric.data.Data` objects, limiting direct use in GNN training pipelines.
4) **Static graph:** Proteins are dynamic; the graph captures a single conformation. Ensemble graphs from molecular dynamics trajectories would better represent conformational heterogeneity.
5) **No bond-order information:** Covalent bonds (peptide, disulfide) are not distinguished from spatial proximity edges.

VI. Conclusion

We have implemented a complete protein-to-graph conversion pipeline for GNN featurization. The implementation includes: (1) PDB parsing and six synthetic protein builders; (2) k-NN edge construction via KD-Tree with $O(N \log N)$ complexity; (3) 24-dimensional node features encoding residue type and physicochemical properties; (4) 9-dimensional edge features including Euclidean distance, direction vector, sequence separation, and orientation quaternion; (5) sparse adjacency matrix construction; (6) contact classification into backbone, short-range, medium-range, and long-range categories; and (7) k-sweep analysis demonstrating the transition from sparse backbone graphs to dense tertiary-contact-capturing graphs.

The six preset proteins validate the pipeline against known structural motifs, and the interactive six-page Streamlit dashboard — with PDB file upload, four edge-coloring modes, 35 informational expanders, and 12 theory sections — provides intuitive exploration of graph topology and GNN featurization concepts relevant to geometric deep learning. The total codebase comprises approximately 4,600 lines of Python across five source modules, the Streamlit application, and a comprehensive test suite of 122 tests in 20 test classes.

References

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

[2] J. Jumper *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, pp. 583–589, 2021.

[3] B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, and R. Dror, "Learning from protein structure with geometric vector perceptrons," *ICLR*, 2021.

[4] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," *ICML*, 2021.

[5] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.

[6] J. Kyte and R. F. Doolittle, "A simple method for displaying the hydropathic character of a protein," *J. Mol. Biol.*, vol. 157, no. 1, pp. 105–132, 1982.

[7] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton University Press, 1999.

[8] P. Y. Chou and G. D. Fasman, "Empirical predictions of protein conformation," *Annu. Rev. Biochem.*, vol. 47, pp. 251–276, 1978.

[9] J. Ingraham, V. K. Garg, R. Barzilay, and T. Jaakkola, "Generative models for graph-based protein design," *NeurIPS*, 2019.

[10] P. Gainza *et al.*, "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nat. Methods*, vol. 17, pp. 184–192, 2020.