Enabling Deep Geometric Learning on Cryo-EM Maps Using Neural Representation

Nathan Ranno

A thesis
submitted in partial fulfillment of the
requirements of the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

2021

Committee:

Dong Si, Chair

Wooyoung Kim

Renzhi Cao

Jie Hou

Program Authorized to Offer Degree:
Computing and Software Systems

University of Washington

Abstract

Enabling Deep Geometric Learning on Cryo-EM Maps Using Neural Representation

Nathan Ranno

Chair of the Supervisory Committee:
Dr. Dong Si
Computing and Software Systems

Advances in imagery at atomic and near-atomic resolution, such as cryogenic electron microscopy (cryo-EM), have led to an influx of high resolution images of proteins and other macromolecular structures to data banks worldwide. Deep geometric learning is intriguing for use in structure segmentation, but the native voxel format of cryo-EM maps is unsuitable as input to such methods. We present a novel data format called the neural cryo-EM map that accurately parameterizes cryo-EM maps and provides native, spatially continuous density and gradient data to serve as the basis for a graph-based interpretation of cryo-EM maps. Density values interpolated using the non-linear neural cryo-EM format are more accurate than conventional tri-linear interpolation. Our graph-based interpretations of 115 experimental cryo-EM maps from 1.15 to 4.0 Å resolution provide high coverage of the underlying amino acid residue locations, while accuracy of nodes is correlated with resolution. The nodes of graphs created from atomic resolution maps (higher than 1.6 Å) provide greater than 99% residue coverage as well as 85% full atomic coverage with a mean of 0.19 Å root mean squared deviation (RMSD). Other graphs have a mean 84% residue coverage with less specificity of the nodes due to experimental noise and differences of density context at lower resolutions. Graphs created from atomic resolution maps may serve as input to downstream deep geometric learning applications and may be generalized to transform any 3D grid-based data format into non-linear, continuous, and differentiable format.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

CRYO-EM: Cryogenic Electron Microscopy

3DEM: Three Dimensional Electron Microscopy

NMR: Nuclear Magnetic Resonance

CASP: Critical Assessment of protein Structure Prediction

GCN: Graph Convolutional Network

PPI: Protein-Protein Interaction

SIREN: SInusoidal REpresentation Network

MAE: Mean Absolute Error

MSE: Mean Squared Error

RMSD: Root Mean Squared Deviation

Cα: Alpha carbon atom, the first carbon atom of the backbone in an amino acid molecule

PDB: Protein Data Bank

GPU: Graphics Processing Unit

Chapter 1

INTRODUCTION

Proteins serve an enormous amount of functions within organisms. Their functionality is prescribed by the form of their tertiary structure, which is the three-dimensional spatial arrangement of the composite amino acids. The sequence of amino acids that form the polypeptide chain, or the primary structure, ranges from tens to many hundreds of residues. Each primary structure is deterministic and, when folded into its native state, produces a unique tertiary structure. A viral capsid, for example, is composed of one or more repeating protein tertiary structures[27]. The SARS-CoV-2 pandemic demonstrates the importance of modeling protein functions, as they relate to understanding the virus's interactions, propagation, drug treatment, and infection prevention via vaccines.

The field of 3D electron microscopy (3DEM) is fundamental to the determination and validation of protein structures. Traditional methods such as X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy have helped fill protein data banks with tens of thousands[9] of structures that are used in fields such as drug and vaccine development. Cryo-EM, a relatively newer single-particle technique for samples prepared at cryogenic temperatures[37], has been shown as a source of high quality, high resolution structure maps[21]. The recent improvements in data processing and computation speed have given cryo-EM the ability to capture atomic resolution[20] and near-atomic resolution images of protein quarternary structures and other macromolecular structures. High resolution 3DEM is crucial to further solving and refining of protein structures.

Protein structure determination via computational methods are based on the readily available primary structure. They are faster than cryo-EM in producing a structure output, which may take many months per structure[18], however the complexity posed by large sequences

and inter-woven structures is a limiting factor. Though methods are rapidly improving, even the state-of-the-art methods[29, 42] as demonstrated in recent CASP competitions[15] do not extend to predicting multi-domain structures. In contrast, cryo-EM imaging techniques observe structures in their natively folded state, providing the role of both structure determination and experimental validation. Though once a map is produced, there still remains a non-trivial step of aligning the primary structure to density regions within the map. A number of solutions exist for the partial[7, 35] and full[36] automation of this process. Deep learning has also demonstrably improved both the automation and execution time of producing predicted protein structures in cryo-EM maps[30, 25].

Graphs are an intriguing data format for proteins and other molecular structures due to their physical similarity to the underlying data, and compared to cryo-EM images, the graph data format is much more condensed and concise. Graph-based methods and graph convolutional networks (GCN)[14] are gaining popularity for tasks related to proteins, such as protein-protein interaction (PPI)[40, 39], protein function classification[41, 8], and primary structure alignment onto tertiary structures[33, 17]. In order for graph nodes to best correlate to a spatially continuous region, we must eschew the native grid format of cryo-EM maps, but we must also maintain a high level of accuracy to the contents of the region in between voxels.

In this paper, we present a novel data format for high-resolution cryo-EM maps that can produce a fully continuous, non-linear interpolation of the EM data using neural network representation. Our implementation automatically converts native 3D array data to the so-called neural cryo-EM maps and retains the ability to accurately reproduce the original input. This format may be extended in many ways, and as a case study, we create a novel graph-based interpretation of cryo-EM maps based on the neural network representation on the basis that there is a correlation between atomic locations to points of high density within the cryo-EM map. We show that the graph coverage of the cryo-EM data and node placement is well-suited for additional predictive methods, including deep geometric learning, to determine molecular structure.

Following this chapter, in Chapter 2, we elaborate on important background concepts and works that form the foundation for our contribution to this domain. In Chapter 3, we detail the creation method for the neural cryo-EM format as well as the density graphs that extend the neural data format. Additionally we provide an overview of the software implementation. We present the rationale for our experiments, the results, and a discussion on the results in Chapter 4. Lastly, in Chapter 5, we summarize our work and suggest possible future work that utilizes or extends the work presented in this paper.

Chapter 2

BACKGROUND

In this chapter, we provide information on the data format of cryo-EM maps, its problems, and existing works in order to better motivate this thesis. We describe the basics of the voxel cryo-EM data format and the challenges intrinsic to the use of automated methods with experimental cryo-EM maps in Section 2.1. In Section 2.2 we describe existing work that serves as the foundation for our implementation of the neural cryo-EM format and subsequent density graph.

2.1 Cryo-EM Map Format

Many tools exist to supplement the production of cryo-EM maps in the various stages of map development[34, 19, 43]. Such tools help automate the process of molecular sample preparation, imaging, and image refinement. Once a cryo-EM map is produced and deposited into databanks, it is in the form of a file following the MRC format[3], which includes metadata for describing how to interpret the raw voxel data. The cryo-EM map is composed of 3D voxels, each dimension of the voxel corresponding to a fixed unit of distance, given in Angstroms (Å). Each voxel contains a single channel of values, similar to a grayscale image, discretizing the density of the region bounded by the voxel edges. Given a point within the map to serve as the spatial origin, we are able to use the voxel size to reproduce a grid-aligned representation of the imaged molecular structure, with each voxel corresponding to a small region in 3D-space.

Voxel size is important in the rendering of high quality maps. Typical atomic bonds in the amino acid chain are around 1.5 Å in distance, and this distance serves as a minimum requirement to capture atomic-level detail. However, another metric, resolution, is used

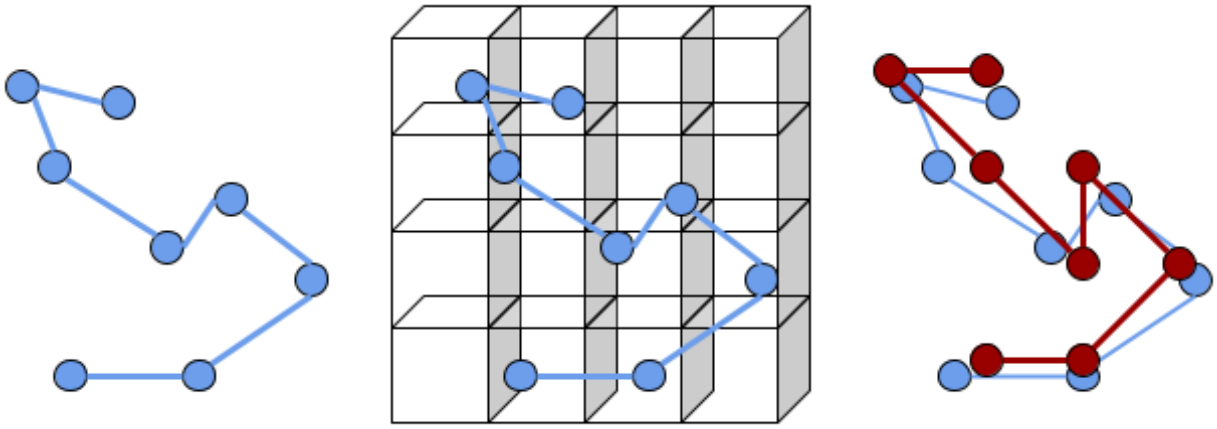

Figure 2.1: A grid, when interposed on a pseudo-structure (blue), will not accurately represent the structure. The structure created from the center of the voxels that contain the pseudo-structure (red) shows that the voxel point representation does not align with the true structure.

to convey the information density of the region covered by the cryo-EM map. Maps with small voxels are not intrinsically higher resolution; the resolution is impacted by the imaging techniques and post-processing methods. "High resolution," often called "near-atomic," maps are those with resolution higher than 4 Å, and "atomic resolution" maps offer a resolution higher than 1.6 Å, which is inclusive of the upper bound for typical bond lengths[32].

Despite the high resolution maps provided by cryo-EM, there are numerous challenges to automating the processing and classification of content in the maps. Some of the challenges are related to the file format itself. Though the MRC format is a standard file format, there is no standard voxel size or map size given the variations of the size of the sample being imaged and the equipment performing the microscopy. There is no standard unit for density values within cryo-EM maps, and the possible range of values is not standardized either. Resolution is similarly not standardized, and it is a self-reported value provided by the submitter of maps to the databanks. This leads to the necessity to pre-process maps in order to apply automated methods for map comparison and structure determination.

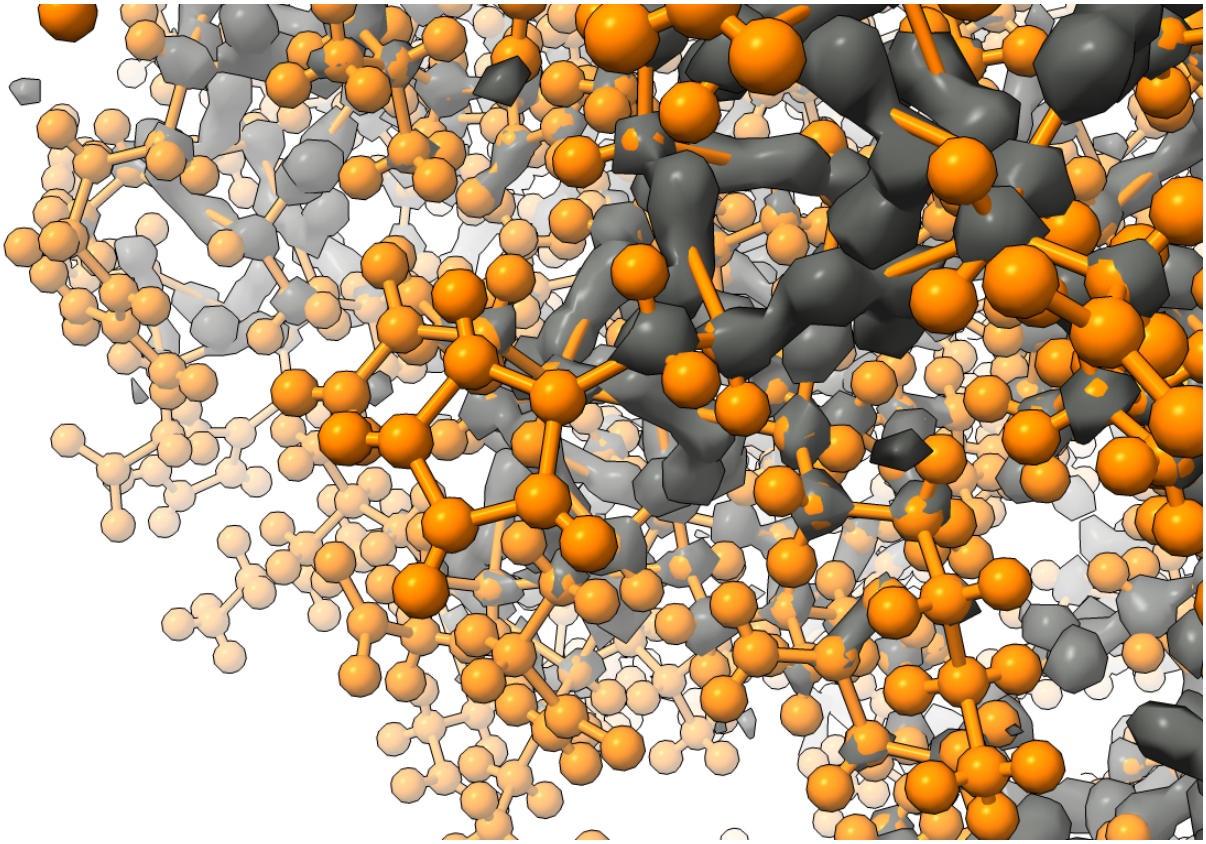

Figure 2.2: A threshold high enough to distinguish separate clusters of voxels is too high for many side chain atoms and atoms on the periphery of the protein structure (orange), demonstrated using EMD-11103. The dark gray voxel data is not uniformly representative of all atoms given a high-pass filter value, which illustrates the need for a better representation of the map.

In addition to issues related to the raw values, there are issues due to the nature of experimentally produced data. Present in experimental data are outliers of density values surrounding the molecular structure that are not associated with the structure itself. Such features are called "noise" and are ubiquitously present in experimental maps. Despite the protein structure being composed of common amino acids, the groupings of portions of the structure into common secondary structures impacts how individual atoms are represented by electron density. Helix and sheet secondary structures typically have higher visibility than loop structures in cryo-EM maps, and the overall shape of those protein segments are more easily recognized. Ideally, each atom has a deterministic density representation globally in the map, but as shown in Figure 2.2 variations in atomic density prevent simple high-pass filtering of the voxels to find atomic locations. The issues with the format and experimental nature of cryo-EM map data impede the ability of automatic methods to reliably process the data.

2.2 Existing Work

Typically, high resolution cryo-EM maps serve to determine protein or other macromolecular structures. Despite the high resolution, the challenges of experimental data discussed previously contribute difficulties to aligning the primary structure to regions within the map, and some efforts to combat the issues, such as noise, prior to deposition present their own problems[28]. Visualization and data manipulation tools, such as Chimera[23] and Coot[5], are instrumental to this largely manual process, but it remains time-consuming.

Fundamental to the operation of existing automatic tools is the transition from the 3D grid-aligned voxel data format of the cryo-EM map to a continuous spatial coordinate system. Simply labeling the original voxels with atomic types will not produce an accurate tertiary structure prediction (Figure 2.1) as the native size of high resolution cryo-EM voxels generally range from 0.5 Å to 1.5 Å. The automatic cryo-EM structure solutions provided by Phenix[36] are established as successful. It performs image processing techniques to find continuous regions of electron density, and uses these collections to constrain the given

primary structure into the map. Once the primary structure is aligned with the cryo-EM map, the Phenix software performs refinement of the structure based on likely physical constraints and local atomic interactions, which effectively moves the structure from the voxel domain to the continuous spatial domain. Another method called MAINMAST[35] accomplishes voxel-to-spatial domain shifting by applying the mean shift algorithm[2] to calculate possible Cα locations. Then it finds a minimum spanning tree across the possible atom positions to create a prediction of a protein chain. DeepTracer[25], a deep learning protein prediction pipeline, uses linear interpolation to first resample the cryo-EM map to a common voxel size before feeding the map into the network. Convolutional neural networks, such as the UNet[26] employed by DeepTracer, label individual voxels. Only after the voxel classification has occurred does the pipeline move from the voxel domain to the continuous domain, when it applies a local center-of-mass calculation to identified points predicted to be atoms. Non linear interpolation methods perform well locally, but there is no way to intuit where to apply the methods within the map without being provided that information.

Deep geometric learning contains similar methods to deep learning with images, such as convolution[14] and attention[38] operations, but the classification is on nodes of a graph rather than pixels in an image. Nodes with intrinsic spatial relationships may also be used in deep geometric learning[4]. In such graphs, the nodes occupy continuous space, and in order to produce suitable input, cryo-EM maps must have their features transformed into a continuous spatial domain for maximum spatial accuracy. The output of current protein prediction methods is one possible way to create graphs. However, these methods are common in that they are many steps removed from the original map features and are already making atomic type and location predictions. To preserve the key density features of cryo-EM maps, creating graphs directly from those density features is important. We surmise that DeepTracer could utilize deep geometric learning to improve the results and speed of many portions of the prediction pipeline, such as the atomic location prediction of atomic resolution cryo-EM maps or the primary structure alignment, and transforming the voxel representation into a graph representation is crucial to applying such methods.

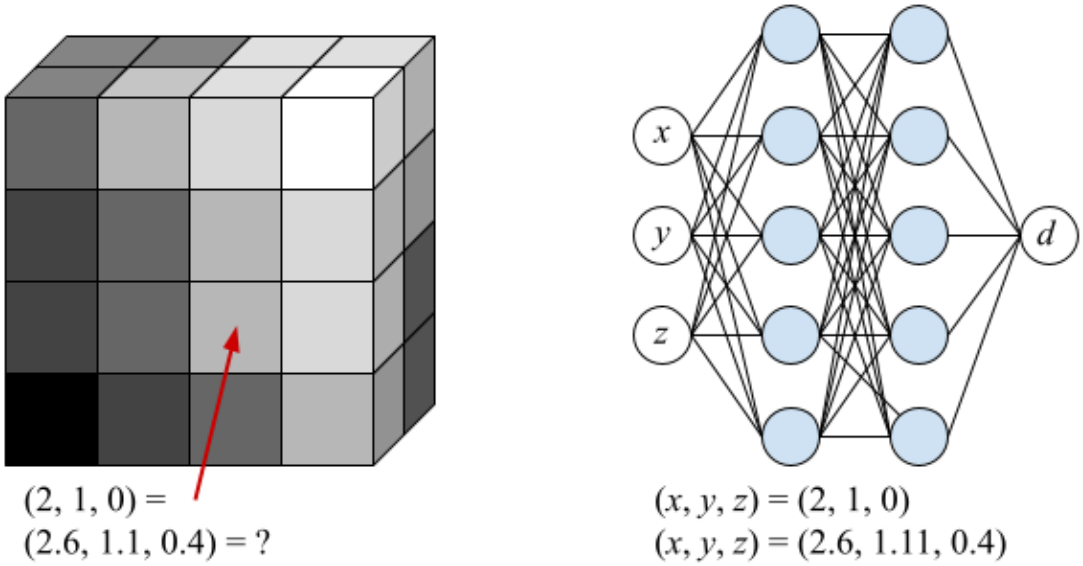

Figure 2.3: The native voxel representation of cryo-EM maps has a discrete number of density values, but the neural representation can represent virtually any spatial coordinate (constrained by floating point representations), including those between voxels.

The use of neural networks in representing and interpolating images is not a common task, but given the constraints of a voxel cryo-EM map, interpolation is a necessary part of predicting protein structures within them. There are a large number of possible network architectures to choose from, and the goal of such architectures is largely to generalize large amounts of data. Figure 2.3 shows the concept of transitioning from the voxel format to a neural format and the benefits gained toward the task of interpolation. Our task is essentially overfitting a network, or set of networks, to reproduce the original image, while also assuming that coordinates input to the network that are not originally in the map are accurately interpolated. The networks that serve neural representation are not generalized across multiple images; there is a bespoke network per image or region of interest. The SIREN[31] architecture is exceptional in this task. The sine layer presented by Sitzmann et al. uses, along with a specific weight initialization scheme, a sine function to wrap a linear transformation of the vector $\mathbf{x}$, with a weight matrix $\mathbf{W}$ and biases b: $y = \sin(\mathbf{W}\mathbf{x} + b)$. The

success of SIRENs in preserving the derivatives of the original signals is also of particular note, underlying the architecture's performance in interpolation tasks. We propose that extending the architecture to three-dimensional images would show success in the representation and interpolation of cryo-EM maps.

Chapter 3

METHODS

The basis for our continuous and non-linear interpolation of cryo-EM maps is the use of one or more neural networks that parameterize the underlying map, producing EM density values given spatial coordinates. Though there is an extremely diverse set of possible neural network architectures for this task, the SIREN[31] network architecture has been shown superior to other network types in ability to accurately represent natural signals. In this chapter, we present the sequence of operations performed on cryo-EM maps to create from them neural cryo-EM maps, including the pre-processing (Section 3.1.1), training(Section 3.1.2), and post-processing steps(Section 3.1.3). Following that, in Section 3.2, we detail the extension of the neural format into density graphs. An overview of the entire process of both the neural and graph formats is shown in Figure 3.1. We give a brief overview of the software implementation by showing the software architecture (Section 3.3). Last, in Section 3.4 we show how we create our datasets for the experiments detailed in the next chapter.

3.1 Neural Cryo-EM Map

3.1.1 Pre-processing

Cryo-EM maps are deposited and stored in the EM databases as three-dimensional arrays, where each index contains a density value, d. The array axes, commonly referred with the labels i, j, and k, are correlated to real spatial coordinates by their respective voxel size in that axis. Each map contains a header that describes voxel sizes, which may not be uniform in each axis, as well as the relationship of the i, j, and k axes to the spatial coordinate axes, which we label x, y, and z. Cryo-EM maps may not be consistent with other maps in terms of the arrangements of axes in relation to spatial coordinates. We account for

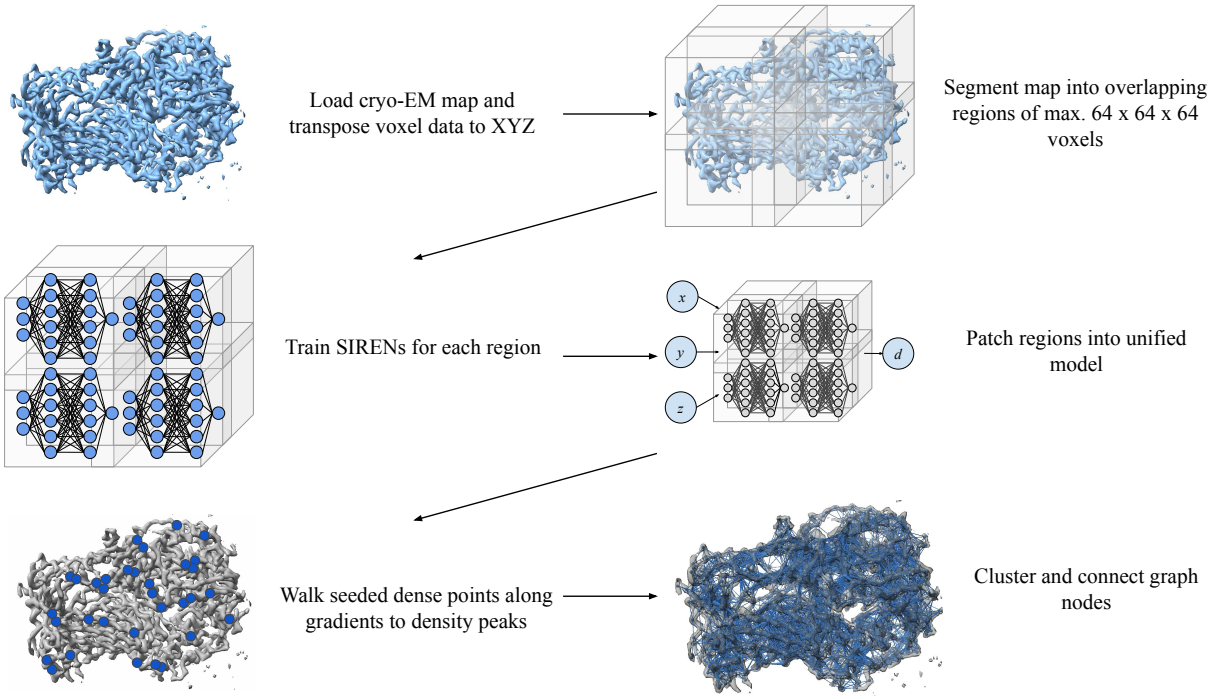

Figure 3.1: Overview of the creation process of the density map graph. 1) Transpose, if necessary, voxel data in the map to align with a consistent XYZ view. 2) Divide the map into regions of max. 64x64x64 voxels that overlap by no less than four voxels in each axis. 3) Train a SIREN for each region using one or more GPUs in parallel. 4) Patch the regions together such that any input coordinate produces an output by Equations 3.4 and 3.5. 5) Use the neural cryo-EM format to seed the spatial area with points and iteratively walk the points along their gradients to density peaks. 6) Cluster the points into nodes and connect them based on an adjacency threshold.

this by maintaining that the overall inputs are the x, y, and z coordinates respectively and transposing the voxel data to be consistent with this arrangement.

The density values of cryo-EM maps are unit-less and are not consistent between maps. Therefore we apply a normalization to the values across the entire map to the range of [-1, 1] which is suitable for training SIRENs. This is a two-step process. The first pass of normalization sets the negative density values of the original map to the lower bound of zero and scales the remaining values to the range of [0, 1]:

$$d_1 = \begin{cases} 0 & d_0 \leq 0 \\ \frac{d_0 - d_{min}}{d_{max} - d_{min}} & d_0 > 0 \end{cases} \tag{3.1}$$

The second pass expands the range to [-1, 1]:

$$d = 2d_1 - 1 \tag{3.2}$$

Initially, the variation in cryo-EM map shape led to the approach of varying the neural network size accordingly. However, after experimenting with the parameters for scaling the networks, the range of cryo-EM map sizes proved to be too large to effectively scale the networks. Even medium-sized maps took an incredibly long time in the SIREN training stage, due to both the size of the network and number of voxels used in training. By using a static neural network architecture across one or more sub-regions of the cryo-EM map, the training time of the entire map scales linearly with the voxel count. This approach also leads to a straightforward multi-GPU strategy when creating neural representations for cryo-EM maps that require multiple SIREN networks.

The voxel data of cryo-EM maps are divided into three-dimensional sub-regions of voxels with the maximum length of any axis of the sub-region being limited to 64 voxels. Each sub-region overlaps with its neighboring sub-regions along each coordinate axis by no less than four voxels. If the total number of voxels in a given axis is n_v, the starting index for each region i_n, is calculated using the region size v_r, the number of regions n_r, and the spacing

interval s.

$$v_r = \begin{cases} n_v & n_v < 64 \\ 64 & n_v \geq 64 \end{cases}$$

$$n_r = \lceil \frac{n_v - v_r}{v_r - 4} \rceil + 1$$

$$s = \frac{n_v - v_r}{\max(n_r - 1, 1)}$$

$$i_n = \lfloor (n-1)s \rfloor, n \in \{\mathbb{Z} \mid 1 \leq n \leq n_r\} \tag{3.3}$$

These calculations are performed per axis, and the results are used to slice the voxel data into sub-regions. The sub-region voxels are used to train distinct SIRENs, and the boundaries are used in querying density and gradient values, as shown in Equations 3.4 and 3.5. For each sub-region in the both training and data retrieval, the x, y, and z coordinates contained within the region are interpreted as a floating point value in the range of [-1, 1], normalized from the minimum and maximum values of x, y, and z in the given sub-region.

3.1.2 Training

Cryo-EM maps are pre-processed and divided into overlapping sub-regions with a maximum size of 64 voxels in each axis, where each sub-region is allocated a distinct SIREN neural network. Each network is unique per the given voxel region, and it is not interchangeable between maps. The network architecture (Figure 3.2) is a fully connected multi-layer perceptron with a sinusoidal input layer, four hidden sinusoidal layers with 256 features, and a final linear output layer. Weights are initialized in the network from uniform distributions $\mathcal{U}(-\sqrt{6/n}, \sqrt{6/n})$ for the first layer and $\mathcal{U}(-(1/\omega_0)\sqrt{6/n}, (1/\omega_0)\sqrt{6/n})$ for the subsequent layers, with $n = 256$ and $\omega_0 = 30$. With three input dimensions and one output dimension, the network fully loaded with the maximum sub-region size results in roughly 4 GB of memory space, which may be accommodated by most GPUs commonly used for deep learning. The network architecture and sub-region size were determined to balance spatial coverage, training

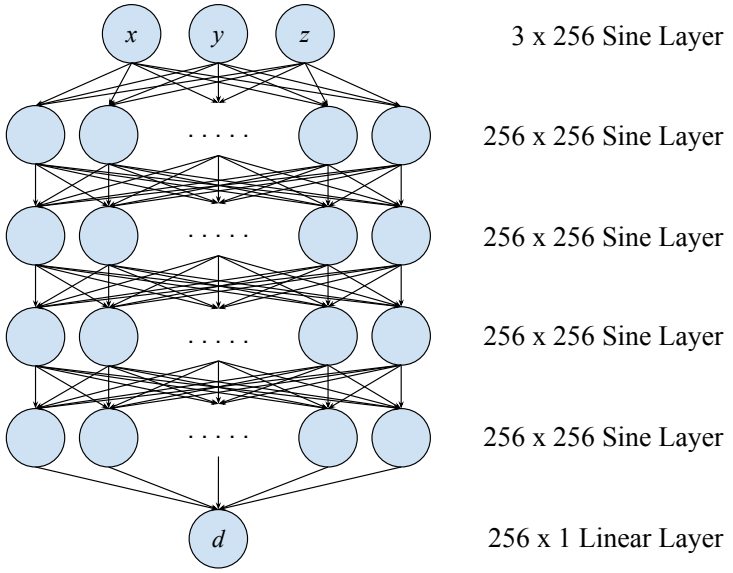

Figure 3.2: Architecture of each SIREN used to represent a region of voxels. There are 256 hidden features per 4 hidden layers with a final linear layer to output the cryo-EM density given an spatial coordinate.

time, and non-volatile storage space. We use the PyTorch[22] framework for neural network and many data manipulation operations.

The SIREN training process utilizes the observed periodic behavior of the network fit improving for many epochs and then briefly regressing, leading to an overall network fit improvement for each of these cycles. Using the mean squared error (MSE) as the loss function along with the Adam optimizer[13], the network is trained to its "natural fit point," which we define as the point at which the lowest MSE loss value, l_{min}, has not been improved for 25 epochs while $0.00001 \leq l_{min} < 0.0004$. If $l_{min} < 0.00001$, the training loop exits immediately. In practice, this reduces the noise ceiling for cryo-EM regions with relatively smoother contents, such as empty space. Since the goal of SIRENs is to fit the network to all the voxels in the map, there are no separate data splits for validation and testing. The separate nature of each SIREN leads to the ability to parallelize the training of a neural cryo-EM map over multiple GPUs.

3.1.3 Data Retrieval

When allocating spatial coordinates to sub-regions, we observed discontinuities in the output values when using hard boundaries between sub-regions. Therefore, we employ a strategy of weighing the output of the distinct networks by the coordinate's position along the axes of overlap and performing an average with the output of all networks that contain the coordinate. Once the SIRENs are trained, the sub-regions are patched together such that any spatial coordinate p in the cryo-EM map exists at a point where sub-regions overlap along n axes, and given that sub-regions were created to overlap strictly along the coordinate axes, the possible values of n are 0, 1, 2, and 3. The nominal case, $n = 0$, means only one neural network r_0 is used to produce the output d:

$$d = r_0(p) \tag{3.4}$$

In the off-nominal cases, $1 \leq n \leq 3$, for each axis overlap there are two regions r_{n1} and r_{n2} that produce output at the coordinate. The network outputs are weighted by the corresponding coordinate component's distance along the overlap and averaged with any other overlapping axes:

$$w_{n1} = \frac{r_{n1end} - p_n}{r_{n1end} - r_{n2start}}, w_{n2} = 1 - w_{n1}$$

$$d = \frac{\sum_{n=1}^{3} w_{n1} r_{n1}(p) + w_{n2} r_{n2}(p)}{n} \tag{3.5}$$

The density value is query-able at any point in the neural cryo-EM map. To allow for meaningful human inspection and rendering of SIREN output in common 3DEM tools such as Chimera[23] and Coot[5], we transform a query's SIREN output, d_{out}, in a post-processing step from the internal nominal range of [-1, 1] to the range of [0, 1] to give the final output d.

$$d = \frac{d_{out} + 1}{2} \tag{3.6}$$

The actual values, d_{out}, are not strictly constrained to the range of [0, 1] in order to allow the network to interpolate beyond the range of the initial training data.

While the neural cryo-EM data format provides the ability to sample density data from anywhere in the map, it is important to distinguish that it does not increase the data resolution. The data format provides a non-linear interpolation of density that is fully continuous and differentiable. This format may be employed and extended in different ways, but for this paper we present a novel graph-based format constructed using the neural cryo-EM map.

3.2 Cryo-EM Density Graphs

Our graph-based interpretation of cryo-EM maps is an extension of the neural cryo-EM map format. The nodes of the graph describe the locally dense spatial coordinates throughout the map. The context of the nodes in relation to the molecular structure depends on the resolution of the underlying cryo-EM map. In general, for high resolution maps, the intention is for nodes to correspond to amino acid residue locations. The lack of cryo-EM map annotation means that the resulting feature vector of each node is composed of four values: the three-dimensional spatial coordinate and the density at that location. Each edge of the graph may contain an optional feature vector, depending on the intended use of the graph in a future downstream implementation. The edge's feature vector is a single dimension with the value of the edge's length.

The method by which the graphs are created relies heavily on the fully continuous and differentiable nature of the neural cryo-EM representation. At any spatial coordinate contained in the cryo-EM map, the neural format may be queried for a density value and a gradient vector, which gives the magnitude and direction of density increase. The graph creation is the ensemble of summarily naive steps which are made possible by the neural cryo-EM format. We employ a global generic thresholding mechanism to filter out irrelevant regions of the cryo-EM map. When creating the neural cryo-EM map, the mean and standard deviation of the normalized SIREN training data are retained. The threshold value T is given as $T = \mu + 3\sigma$, and seed points are determined by sampling the entire map with a 0.5 Å step

Data: Seed points of the map
Result: All seed points either deleted or moved to a density peak.
while *seed points remain in pool* **do**
  for *each point in the pool* **do**
    if *point not in map* **then**
      remove point from pool;
    end
    calculate density and gradient vector;
    if *calculated density* $\leq$ *cached density* **then**
      add cached point location to results;
      remove point from pool;
    else
      cache current position;
      cache current density for point;
      modify point location by 0.05 Å in direction of gradient vector;
    end
  end
end
return results;

Algorithm 1: Gradient-walking algorithm used in detecting local density peaks using the neural cryo-EM map format

size in each axis, discarding points below the value of T.

Each seed point is iteratively moved along its gradient vector until a density peak is detected (Algorithm 1). The density peak is the point of highest detected density while traversing the gradient vector with a step size of 0.05 Å. For each step iteration, the point location and density are retained for reference against the next step in order to compare densities and provide the correct position in the results. It is possible that traversing in the

direction of the gradient results in points exiting the spatial domain of the map. In this case, the seed point is simply removed from the pool of seed points.

Once all density peaks have been reached, the DBSCAN[6] clustering algorithm is performed on the points. For our case, two points are considered part of the same cluster if they are within 0.2 Å of the other point, and points without a neighbor within that range are considered their own cluster. While this latter setting decreases the specificity of the node placement and makes it more susceptible to noise, we found that the tradeoff with an increase of overall sensitivity was worth it. The centroid is calculated for each cluster as a potential graph node.

While our graph evaluation is focused on the node placement and representation of the underlying deposited structures, graphs are not simply a set of nodes. We connect nodes with edges based on an arbitrary spatial adjacency threshold 2 times greater than the reported resolution of the underlying cryo-EM map, which we found to balance the resiliency against potential missed atomic locations with the presence of noise in the final graph. In order to create the edges, a pairwise adjacency matrix is computed over all nodes, and any indices whose value is below the threshold are used to create node pairs. Nodes without edges are removed from the graph. The remaining nodes and edges compose the output graph-based interpretation of a cryo-EM map.

3.3 Software Architecture

The software implementation of the neural cryo-EM format and density graph is encapsulated in three Python modules in order to isolate dependencies: one for each of the formats respectively and an additional one for the neural network abstraction. The neural network creation and operations are built upon the PyTorch framework[22]. PyTorch, along with NumPy[12], is also used throughout the SIREN and neural cryo-EM modules for many basic array, tensor, and data loading operations. The density graph's underlying data structure is a wrapper around a *Graph* class provided by the NetworkX[10] library. Each of the dependencies is publicly available with no restrictive licensing.

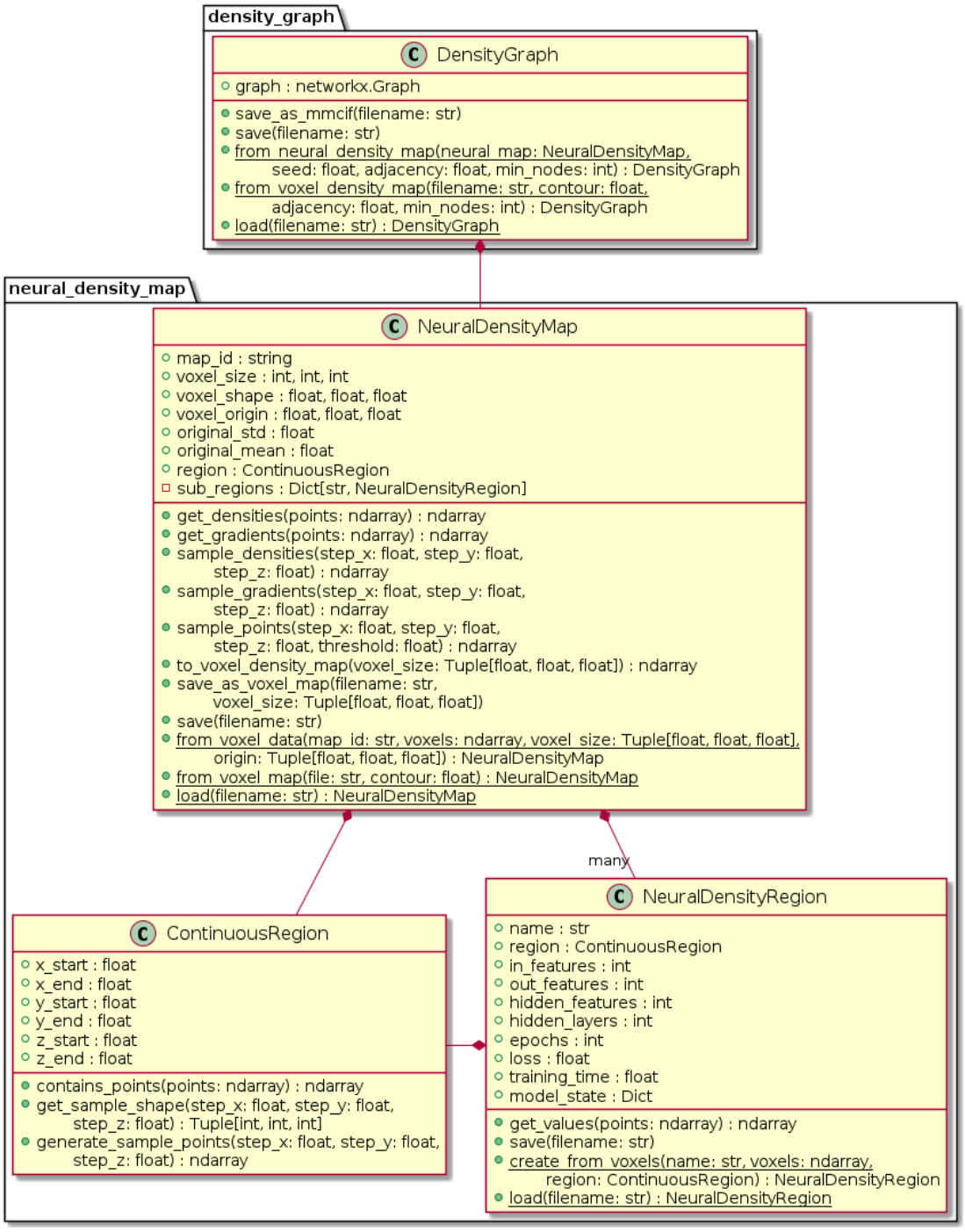

Figure 3.3: Modular software architecture and relationship of the neural cryo-EM map and density graph implementations. Excluded is the module encapsulating the SIREN implementation, which is an extension of PyTorch functionality.

As shown by the public methods of the *NeuralDensityMap* module, the focus is on two main creation methods and two types of data retrieval workflows. The neural cryo-EM map may be created directly from a voxel map file, or it may be created from arbitrary regions of voxel data. The experiments we detail in Chapter 4 create neural cryo-EM maps from the entire original voxel map, but tailored creation will result in more rapid operations. The class provides the ability to retrieve both the density and gradient vector of any points in the continuous region occupied by the map. Additionally, the density and gradient values may be sampled at fixed step sizes in each dimension of the map, which is how the density graph utilizes this class.

Density graphs rely on the functionality provided by the neural cryo-EM map implementation, but the *DensityGraph* module provides the ability to create a graph directly from a voxel map, effectively condensing the neural map and graph creation into a single step. The underlying structure of the density graph is a *Graph* from the NetworkX library[10], rather than a custom graph data structure. We made this choice based on three factors. The first is that we attempted our own data structure and found the task to be quite challenging to optimize and extend. This led to a search for third-party integration. The NetworkX library is both mature for our use case and currently active. The third factor is that the *Graph* structure integrates with numerous graph-related algorithms provided by the library, such as link analysis and node classification, which will be useful in future, downstream uses of the density graph.

Both the neural cryo-EM map and density graph have built-in saving and loading mechanisms, which significantly improve the usability of the data formats (See Chapter 4, Section 4.3). This functionality additionally brings the ability to visualize the data formats. Voxel cryo-EM maps may be created and saved by sampling the neural network at regular spatial intervals corresponding to the desired voxel size. Density graphs are visualized by taking advantage of the immense functionality of the CIF file format[11]. Using this format we are able to arbitrarily place graph nodes as atoms and graph edges as atomic bonds. The resulting structure is not a valid atomic structure, but it serves as a way to visualize the

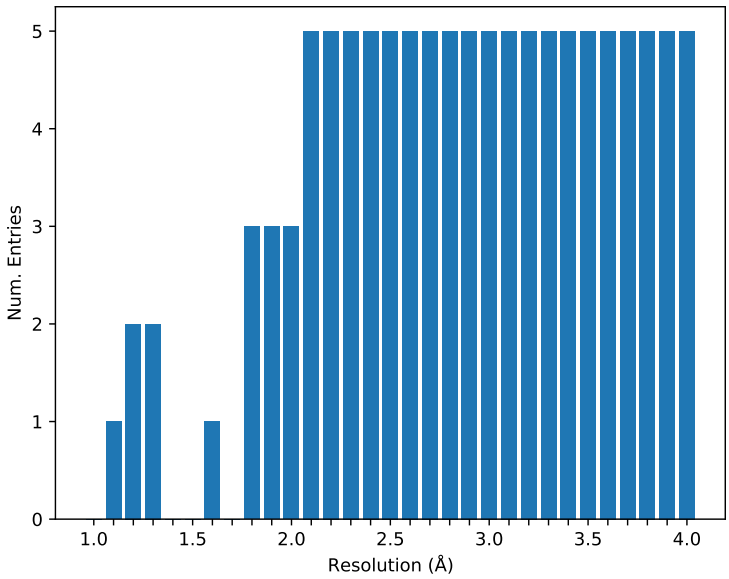

Figure 3.4: The distribution of the data set across resolution. There are unrepresented resolutions and a general lack of maps at the high end of the resolution range.

graph nodes and edges using available visualization tools, such as ChimeraX[24].

3.4 Data Sets

This paper derives two sets of cryo-EM data, one of simulated maps and the other of experimental maps, from the same source. The base for the datasets is all the high resolution ($\leq$ 4 Å) cryo-EM maps from the EM Data Resource[16] that have an associated PDB deposition. Each deposition is often one of many from a given publication, and we filter the base dataset for only the map and corresponding structure of highest resolution from each publication. This significantly reduces the number of duplicate and very similar entries in the data pool.

From this data pool, entries are grouped by resolution rounded to the nearest 0.1 Å. Up to five entries are randomly selected from each group to represent that slice of the resolution range. If five maps entries are not available in a group, then all the entries of the group are selected. Additionally, if a cryo-EM map of a selected entry contains more than 512^3

voxels, the candidate entry is ignored, and the random selection is retried with that map removed from the pool. This results in a total of 115 high resolution cyro-EM maps and corresponding deposited structures. Figure 3.4 shows the distribution of the data across the range of resolutions.

Chapter 4

RESULTS

A dataset of 115 experimental cryo-EM maps and corresponding PDB-deposited[9] structures across the range of high resolutions (≤ 4 Å) serves as the basis for the experiments presented in this paper. In this chapter we present the results of experiments in three main areas. By constructing simulated maps from the deposited structures at the reported experimental cryo-EM resolutions, the neural cryo-EM map's capability for interpolation is evaluated, comparing it against a tri-linear interpolation (Section 4.1). Experimental cryo-EM maps are used to present a novel use for this data format, a graph-based interpretation of cryo-EM maps (Section 4.2). We evaluate the graphs' coverage of the underlying structure along with the accuracy of node placement with respect to residue and atom locations. The results are also compared to the prediction output of the state-of-the-art cryo-EM modelling tool DeepTracer[25]. There is no standard benchmark for measuring the performance of our code implementation, and many optimizations may be made depending on the task. However, it is important to capture the technical performance of the method to serve as a reference (Section 4.3). We finish this chapter with a discussion about the results (Section 4.4).

4.1 Non-linear Interpolation

Simulated cryo-EM maps, created using Chimera's *molmap* tool[23], are constructed from the sum of resolution-dependent Gaussian functions centered on atomic locations, providing determinism throughout the spatial region of the map. This determinism acts as the control in the evaluation of the interpolation capability of the neural cryo-EM map format. For each deposited structure in the 115-map dataset, the reported resolution of the corresponding cryo-EM map is used as the target resolution for *molmap*. Two simulated maps are created

per entry, one with the tool's default voxel size of $(resolution/3)$, which is equivalent to the ratio seen in typical experimental maps, to serve as the input to experimental interpolators and the other with a voxel size of 0.2 Å to serve as the control for interpolated values. We chose the value of 0.2 Å to show the ability to interpolate across the entire range of resolutions in our dataset without exceeding the memory resources of our test system in the case of the voxel maps. All other arguments remain their default value.

We evaluate the interpolations against the control by the metric of mean absolute error (MAE) with the initial and control maps' voxel values normalized to the range of [0, 1]. The voxels of simulated maps have the value of zero in empty regions of the map. The neural cryo-EM map format, by its nature, contains a very low level of global background noise correlated to the amount of loss existing at the exit point of the training loop, which influences a global MAE calculation. This noise is absent in a tri-linear interpolation, and to reduce the influence of the known neural noise, only the voxels of the control that contain a non-zero value are used in the comparisons of the types of interpolations.

As shown in Figure 4.1, the interpolation performance of the neural cryo-EM map format is an order of magnitude more accurate to the true values of simulated map, consistently showing a MAE of < 0.01, and it does not appear to worsen with a decrease in resolution. The tri-linear interpolation not only worsens with decreasing resolution, it is unable to capture non-linearity of the underlying data, effectively performing a smoothing of density peaks in the map. The overall average MAE for the tri-linear interpolation is 0.066, but rises to 0.12 for the lowest resolutions in the range. In contrast, the learned neural representations of the maps are able to not only capture the non-linearity but also preserve the ability to interpolate density values higher or lower than the initial maps' inputs.

In very high resolution simulated maps, the interpolation performance is exceptionally high at about 0.0005 MAE, but the gap in performance between the two interpolators is not pronounced. This is likely due to the rapidly diminishing Gaussian function around atomic locations of the underlying control map, which means that the original simulated map has already captured much of the density data. The amount of interpolation performed is also

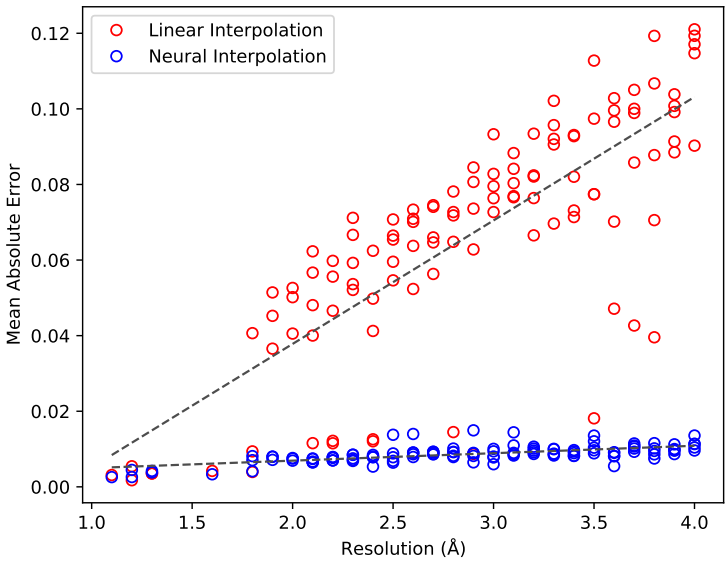

Figure 4.1: Plot comparing the mean absolute error of tri-linear and the neural interpolation against 115 simulated cryo-EM maps. The neural interpolation significantly outperforms the tri-linear interpolation and does not suffer worse performance as resolution increases.

less in these cases due to the initially small voxel sizes on those maps. For example, an atomic-level resolution map simulated at 1.2 Å has a voxel size of 0.4 Å, meaning only 2 voxels are interpolated for every voxel of the original source.

As shown, the neural cryo-EM map format contains the ability to capture the non-linearity of underlying data, preserve density peaks, and provide spatially continuous and differentiable data. This leads to many extensions of the data format beyond what the conventional voxel representation provides. One such format is a graph, which we present in the next section.

4.2 Graph-based Interpretation

A density graph is created from a neural cryo-EM map by seeding the map with points, incrementally adjusting the points' position in the direction of their gradient vectors and stopping when a peak is detected, clustering the points using the DBSCAN[6] algorithm, and calculating the centroid of each cluster. Candidate nodes are placed at the centroid locations

and connected with edges based on an adjacency threshold and sub-graph constraints using the NetworkX[10] library.

From our dataset of 115 experimental cryo-EM maps, there are six maps at or below 1.6 Å resolution, which we consider "atomic resolution" to be inclusive of the upper bound for typical peptide bond lengths[32]. The dense points in these maps are largely correlated to individual atoms, as opposed to the general location of the amino acid residues. Table 4.1 shows the evaluation the cryo-EM density graphs from atomic resolution maps against all atoms documented in their respective deposited structures. The constraint in the sensitivity and specificity calculations is a 1 Å radius around atoms and nodes. Considering just the Cα atoms, typically the most prominent atoms in amino acid residues, the density graphs provide a coverage of 99.4%. For all atoms documented in the PDB-deposited structure, the density graphs provide an average 85.2% coverage as well as an average 88.4% specificity rating of nodes to atoms. The RMSD of the matching nodes to their respective atomic locations is very low with a mean value of 0.19 Å across the atomic resolution density graph set. Figure 4.6 depicts an atomic resolution density graph compared with the deposited structure and the voxel grid data.

Due to the nature of the method as essentially a "dense point detector," the context of what a node might represent differs between cryo-EM resolutions. In the < 1.6Å range, dense points largely represent individual atoms of the protein residues, including the atoms in the both the backbone and side chains. With other high resolution maps ≤ 4Å, the dense points are more indicative of amino acid residue locations, but are not precisely atomic locations. Despite this, the Cα atom locations of the deposited structure serve as the basis for evaluating the graph's of near-atomic resolution maps because every amino acid residue contains one, and they are relatively centrally located in a given residue. At the near-atomic resolution, nodes and residues are considered matched if a node is within 3 Å of the Cα atom.

Our results (Figure 4.2) show a clear delineation along the boundary of atomic and near-atomic resolutions. The near-atomic resolution graphs (> 1.6Å) have a mean RMSD of 1.13 Å and 84.5% match. As resolution decreases, the graphs' performance in these metrics

Atomic Resolution Density Graphs					
EMDB ID	PDB ID	Resolution (Å)	RMSD (Å)	Sensitivity (1 Å)	Specificity (1 Å)
11668	7a6a	1.15	0.138	87.98	96.21
11638	7a4m	1.22	0.192	87.00	88.81
11103	6z6u	1.25	0.147	88.73	95.34
11669	7a6b	1.33	0.151	84.25	94.32
22657	7k3v	1.34	0.226	78.82	89.44
11121	6z9e	1.55	0.291	84.44	66.28

Table 4.1: Evaluation of cryo-EM density graph node locations against all atomic locations contained in the PDB-deposited structure file. The sensitivity is the percentage of atoms that match exclusively with a node within 1 Å. The specificity is the percentage of nodes that match exclusively with an atom within 1 Å.

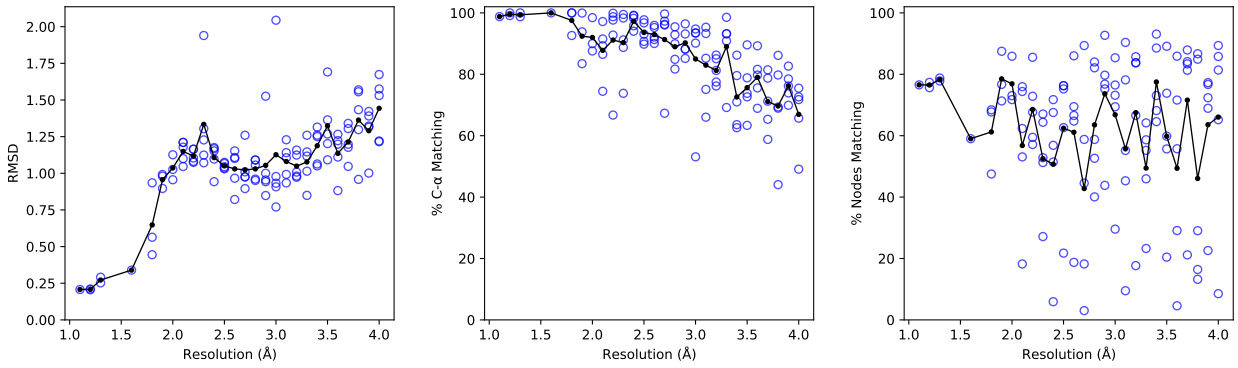

Figure 4.2: For all experimental maps in the 115-map dataset: the root mean square deviation (RMSD) of the closest node to Cα atoms (left), the percentage of Cα atoms of the deposited structure that have a node within 3 Å (middle), and the percentage of total nodes that are within 3 Å of a Cα atom in the deposited structure (right). Superimposed on the scatter graphs is the average value across all points at the given resolution in 0.1 Å increments.

decreases, which is expected. Interestingly, a jump in RMSD values appears between the atomic and near-atomic resolution maps, suggesting the possibility that in the latter maps the most dense point in a local area does not correspond exactly to the Cα location.

The results of node sensitivity to residue locations are competitive when comparing against the Cα predictions of the tool DeepTracer (Figure 4.3). DeepTracer[25] is a method for *de novo* protein structure prediction from high resolution cryo-EM maps that uses a U-Net[26] deep convolutional network as the basis for amino acid residue location and type annotations. While the methods are not exactly similar in output, the comparison provides context to the sensitivity metrics of the density graphs. By assuming every node is a potential Cα atom, the overall matching percentage of output to Cα locations is 85.3% and 85.0% for the graphs and DeepTracer predictions respectively. Though the RMSD values of the density graph nodes are worse than the Cα predictions of DeepTracer overall, an average of 1.09 Å against 0.72 Å, the graphs in the atomic resolution range outperform the output of DeepTracer.

4.3 Performance Metrics

Over the course of our experimentation, we gathered two main technical performance metrics of the neural cryo-EM implementation: training time and non-volatile storage size. The training time of the neural cryo-EM format is important when considering implementing it in a pipeline of other tasks. The storage size is likewise important due to the benefits of caching pre-trained neural cryo-EM maps.

The software implementation of the neural cryo-EM map incorporates the use of multiple GPU devices for training, if available. Parallelizing the training drastically reduces the time spent training, but our metrics presented are cumulative across all devices in order to capture a common use case of with single GPU available. Across our dataset consisting of 115 experimental cryo-EM maps, we see the training time of a given map is proportional to the number of neural cryo-EM regions in the map (Figure 4.4). The average training time per region, which is 64x64x64 voxels in all but the smallest maps, is about 100 seconds, and 65 out of 115 maps in our dataset cumulatively spent under an hour training. The maximum

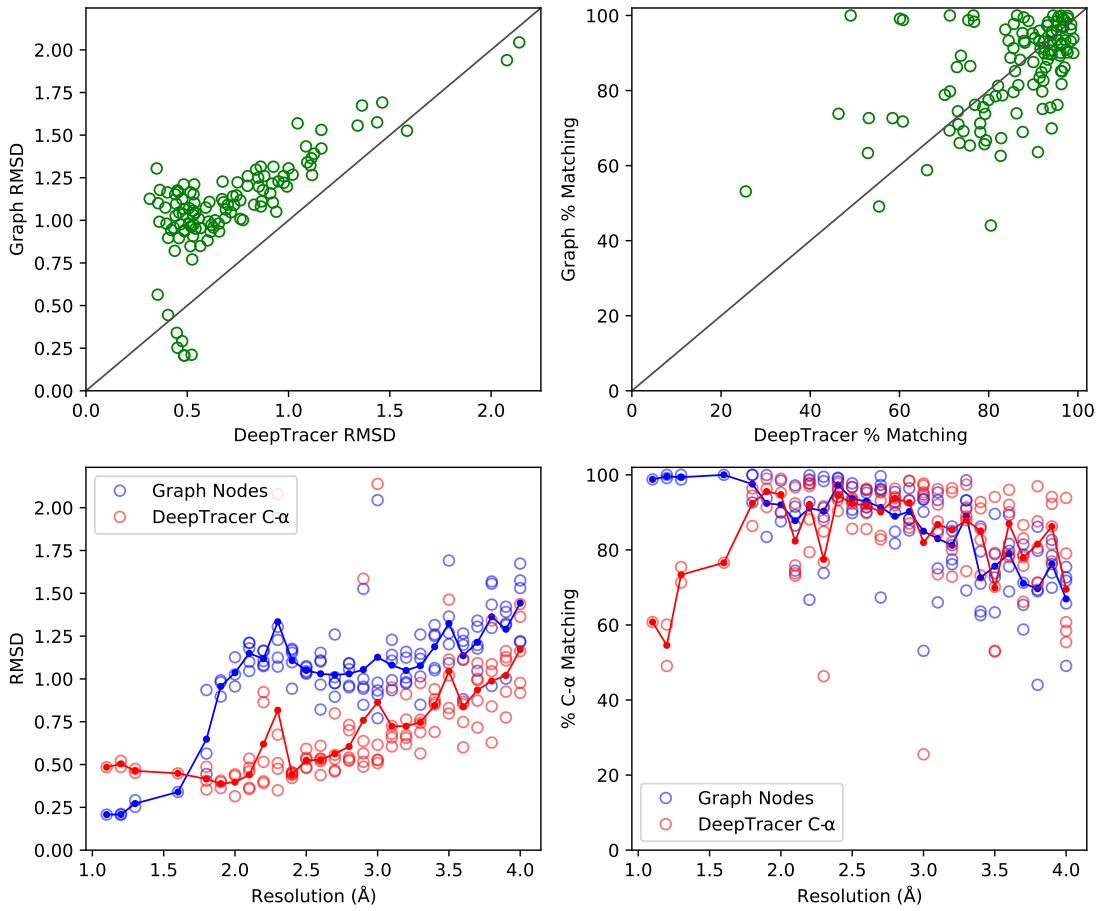

Figure 4.3: Performance of graph construction compared to predictive output of DeepTracer. The map-wise comparison of the root mean square deviation (RMSD) of nodes and Cα predictions (upper-left), map-wise comparison of the percent of the deposited structure's Cα atoms that have a node or prediction within 3 Å (upper-right), and the same values but plotted against the corresponding resolutions (lower-left and lower-right respectively).

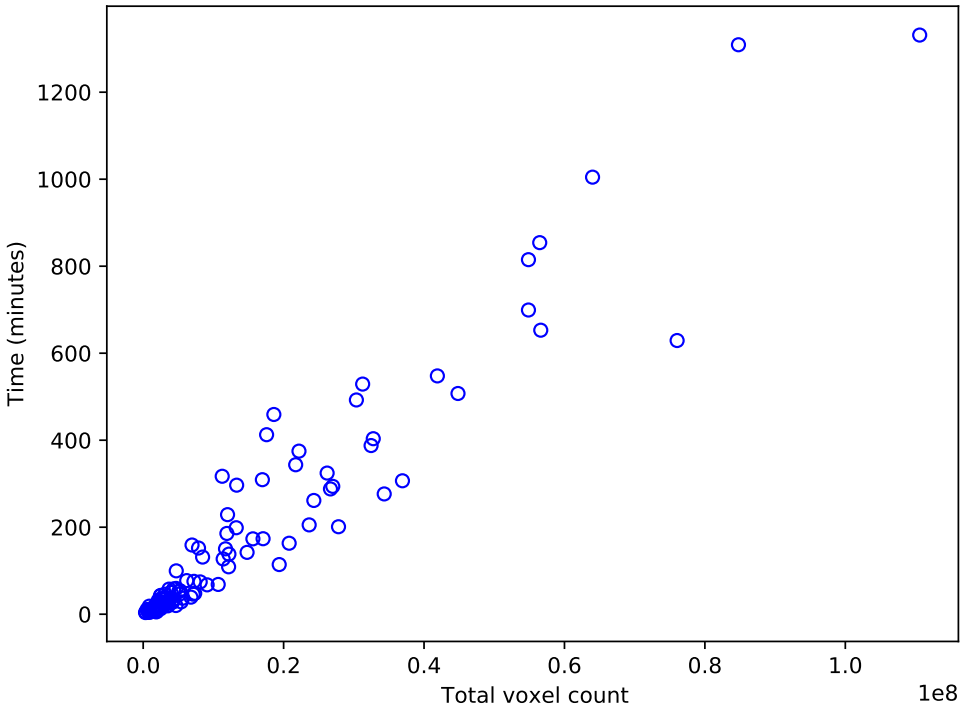

Figure 4.4: Plot of the cumulative training time of the neural cryo-EM maps against their respective map size, based on the total number of voxels. Most of the maps (65 out of 115) cumulatively train in less than an hour, but very large maps take a long time to train.

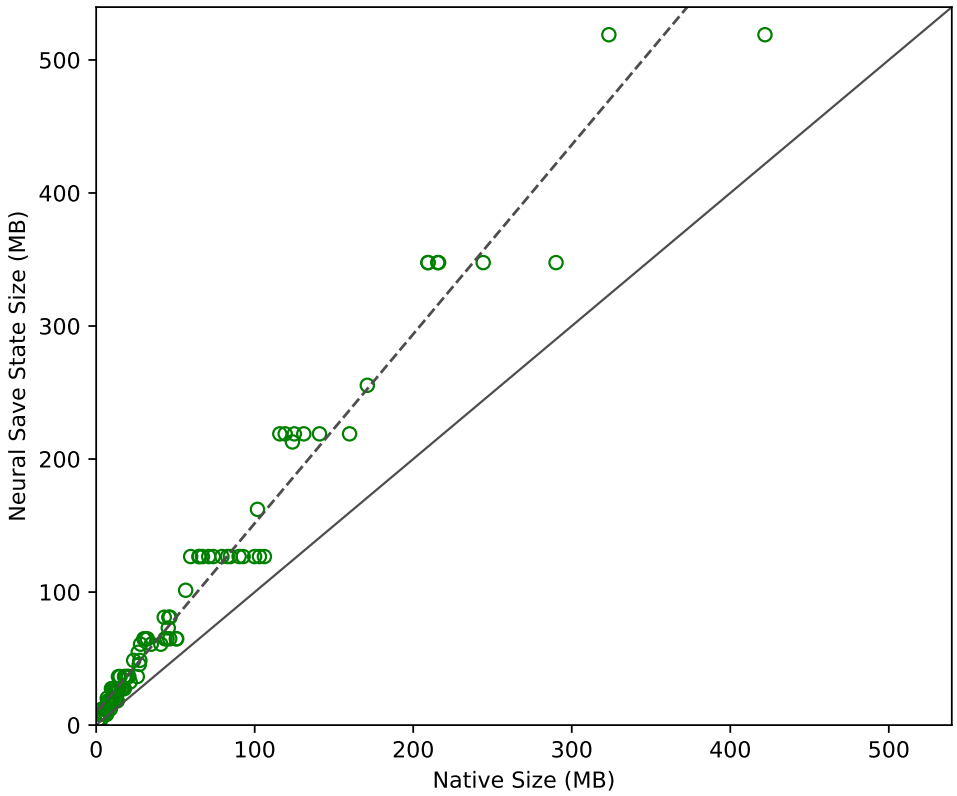

Figure 4.5: The uncompressed storage size of the neural cryo-EM maps plotted directly against those of their voxel map counterparts. The neural format trends higher in all cases, but the average increase is less than double the original size.

cumulative time was about 22.2 hours for EMD-9829[1], which has a native size of 442 MB and contains 512 distinct neural regions. Variance between maps that have the same number of trained regions is likely due to the differences in data smoothness of the voxel map, which is disrupted by isolated voxels of density much higher or lower than the surrounding voxels, i.e. noise.

To mitigate large training times for repeated uses of the same maps, the neural cryo-EM maps may be stored in a non-volatile format. This format consists of the weights of the neural networks that compose the neural map. Figure 4.5 shows that the stored size of neural cryo-EM maps are larger than their native counterparts. Though our implementation

supports saving and loading compressed versions of the neural cryo-EM format, we compare the un-compressed size against the original map size. We see a wide spread in the amount of size increase with an average of 90%, and a standard deviation of 48%. This is a modest increase but well within the bounds for general use cases. We argue the benefit of spatially continuous data with a minimally lossy (< 0.0004 MSE) representation of the original map is worth the additional storage space.

4.4 Discussion

The increasing deposition of maps at high resolutions is a promising sign for the use of our data format and graph-based interpretation in a system of producing atomic structure and type. The density graphs best characterize atomic locations from cryo-EM maps at atomic resolution, and we are unaware of an automated method that performs as well for finding atoms in maps of this resolution. The main significance of this method over simpler methods, such as a combination of a simple high-pass filter and clustering method, is that it handles variations of density values present in experimental cryo-EM maps. Visually inspecting the maps shows that the threshold for a high-pass filter that is high enough to capture individual atoms is too high to capture atoms in side chains and on the periphery of the protein (Figure 2.2). Our method is able to find density peaks due to the superior interpolation performance of the underlying neural cryo-EM format (Figure 4.6).

With the cryo-EM density graph creation method applied to 115 experimental cryo-EM maps, it has the additional challenge of dealing with the noise and artifacts present in experimental maps. In order to support dense point detection for both atomic and near-atomic resolution cryo-EM maps, our method does not discriminate between dense points that are relatively close together. Additionally, with our generic initial threshold value calculation, the seed points of some maps may either correspond to noisy regions or may not cover the entire imaged structure. While the specificity of density graphs created from atomic resolution cryo-EM maps is high, over the full range of high resolution maps, the specificity nodes correlated to amino acid residues is relatively low. The rightmost plot in Figure 4.2 shows

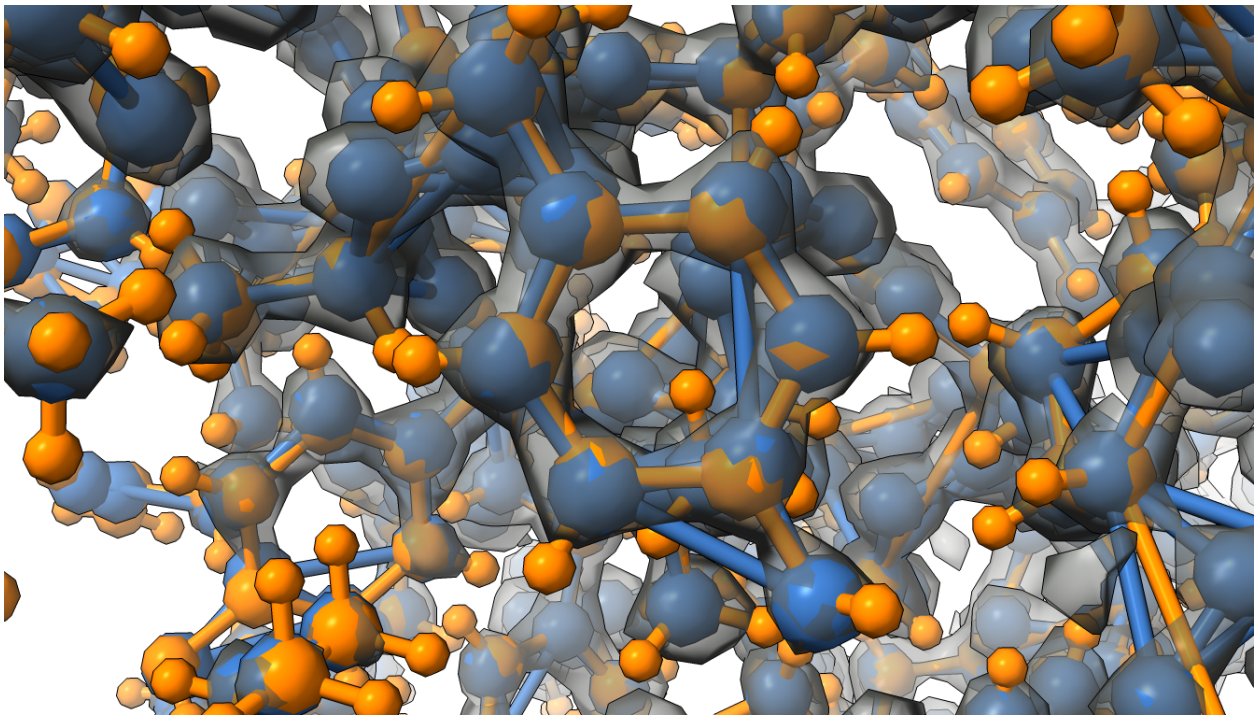

Figure 4.6: Localized view of graph nodes (blue) created by our method for the atomic resolution map EMD-11103 human apoferritin (transparent gray) and the atom locations of the corresponding deposited structure PDB-6z6u (orange).

this with an average of 61.9% of density graph nodes within 3 Å of any Cα atom, and the variance of this specificity metric is high for density graphs based on near-atomic resolution cryo-EM maps.

As previously stated, experimental noise is a contributor to poorer specificity, and it may be mitigated by a more dynamic or user-influenced initial map threshold. We replaced the generic seed-point threshold with the author-recommended contour value, normalized to the scale of the neural cryo-EM data, as the seed point threshold for a density graph created from EMD-10815. This graph had a high sensitivity with 99% of Cα atoms matching with a node, but was the worst scorer in specificity, showing only 3% of nodes within 3 Å of a Cα location. This suggested the presence of extensive experimental noise at the initial contour, and, indeed, the re-creation of the graph with the author-recommended threshold resulted in an increase of specificity measurement to 56% while maintaining a 99% sensitivity to Cα locations. Density graphs with low sensitivity were, however, not improved by manually adjusting the seed-point threshold to author-recommended values.

It is also important to highlight that the basis for the cryo-EM density graphs is density peaks, whose meaning changes depending on the resolution of the original cryo-EM map, and this fluidity of context affects the calculation of sensitivity and specificity to concrete locations. Our methods are applicable to maps of any resolution, but the application of our methods toward atomic location prediction is diminished as the resolution decreases beyond high resolution. Cryo-EM density graphs are inherently descriptive of molecular structure but lack the ability to discern atomic types. We argue that our graph-based interpretation of cryo-EM maps, based on the neural cryo-EM format, is adequate for inclusion in workflows that allow for tailored user interaction in order to provide a boost in node specificity toward molecular structures and type annotation. Fully automated molecular description pipelines may still benefit from the graph format. Given the demonstrated sensitivity, the cryo-EM density graph is suitable as a pre-processing or initial step in an ensemble of other predictive and refinement methods that operate on graph data structures that may further derive macromolecular context.

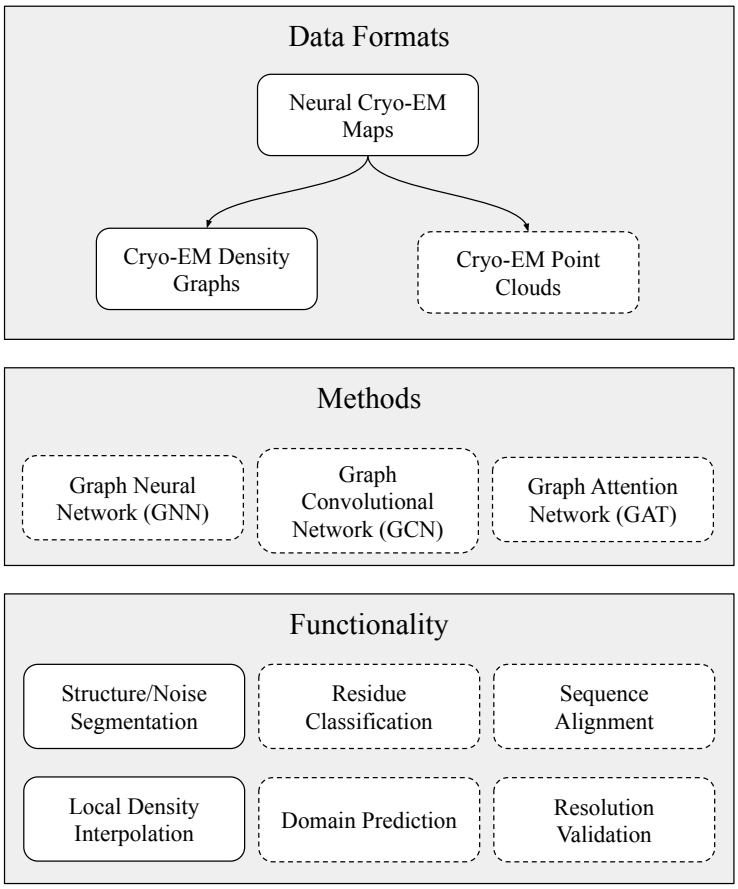

Figure 4.7: Potential downstream applications of the neural cryo-EM format with items shown in this paper having solid edges. The format itself may be extended to create density graphs, as we have in this paper, or other structures such as a point cloud. Neural cryo-EM maps may also be used directly for tasks such as local density interpolation. Using graph-based machine and deep learning, the graphs created from the neural cryo-EM format may be used to predict individual amino acid type, using maps of atomic resolutions, or other predictive functions depending on the resolution of the original map.

Figure 4.7 illustrates the potential downstream uses of the neural cryo-EM map and density graph by researchers. Given our goal of applying deep geometric learning techniques to cryo-EM maps, we created a graph data structure from the neural representation. Deep geometric learning applications exist beyond atomic location prediction, which was the focus of this paper. For example, portraying the cryo-EM data as a density graph is necessary to apply geometric learning techniques for noise segmentation or resolution validation. The graph data structure closely resembles atomic bonds, which leads application of geometric learning in tracing the protein backbone polypeptide chain or aligning the primary structure onto a predicted tertiary structure. The availability of density and gradient values from the neural format lead to extensions toward several other types of data structures for additional applications. For example, rather than density peaks, a point cloud based on density may be constructed from the neural cryo-EM map for the purpose of detecting continuous density regions or domain-level predictions. Utilization of the gradient features of the cryo-EM map, given by the neural representation, may also be useful for detecting the surface of proteins and segmenting multiple protein chains in a multi-domain structure. With an accurate, natively continuous and fully differentiable cyro-EM representation, the neural cryo-EM map enables the application of continuous mathematics, rather than discrete, toward creating alternative representations of the cryo-EM data.

Chapter 5

CONCLUSION

The native voxel format of cryo-EM maps does not provide density information in a spatially continuous manner, rather it discretizes the density data to a grid whose intervals are not of sufficient size to accurately label atomic locations. The format is additionally unsuitable as input to deep geometric learning methods for the purpose of atomic location segmentation. Current methods for atomic location prediction serve well in creating a compatible format, but involve a redundant prediction task which supplants the involvement of deep geometric learning. This belies the need for an effective global interpolator to extract density features in high resolution maps to create nodes of a graph that are representative of atomic locations.

In this paper, we presented a novel data format of cryo-EM maps called the neural cryo-EM map as well as a graph extension of the format that, when applied to atomic and near-atomic resolution cryo-EM maps, form high-coverage, accurate representations of the underlying molecular structure. The neural cryo-EM format offers superior interpolation performance compared to conventional global tri-linear interpolation, effectively capturing non-linearity of the optimal cryo-EM map imagery across the range of high resolutions. The ability to interpolate cryo-EM map data is preserved in experimentally produced maps as well, validated by using the neural cryo-EM format to detect locations of density peaks, which correlate to atomic locations. Despite no additional predictive or refinement methods, graphs created from the neural cryo-EM map and detected dense points show similar residue sensitivity to DeepTracer and greatly surpass it for maps of atomic resolution. Density graphs created for cryo-EM maps of atomic resolution cover over 85% of all atoms in the structure, with the specificity of the nodes equally as high.

The graph data format is especially intriguing with regards to protein representation and

structure determination. While our graph node and edge feature vectors are simply spatial data and density data, the high sensitivity of the node placement shows that they may serve as the basis for further predictive methods that use graphs, such as graph convolutional networks. The creation of graphs based on cryo-EM data may facilitate the combination with and integration of methods generally used in other areas of protein structure prediction, such as domain prediction and sequence alignment. As shown by the relatively low specificity and highly contextual nature of our cryo-EM density graph nodes, opportunities to improve our graph format exist, largely in the area of handling noise present in the experimental maps.

As the resolution of cryo-EM maps continues to be driven higher, the demonstrated ability of this format to capture and detect dense locations becomes important to driving future automated methods for determining structure. We chose to implement a graph interpretation of cryo-EM maps, however, due to the nature of the underlying neural cryo-EM format, graphs are not the only possible extension. The accurate interpolation and ability to sample a continuous spatial region may also be used to create other interpretations, such as point clouds. Additionally, as the cryo-EM data is not constrained by voxels and instead represented by neural networks, the format opens the possibility of being integrated into advanced machine and deep learning systems.

BIBLIOGRAPHY

[1] Lei Cao, Pi Liu, Pan Yang, Qiang Gao, Hong Li, Yao Sun, Ling Zhu, Jianping Lin, Dan Su, Zihe Rao, and Xiangxi Wang. Structural basis for neutralization of hepatitis A virus informs a rational design of highly potent inhibitors. *PLoS biology*, 17(4):e3000229, April 2019.

[2] M.A. Carreira-Perpinan. Acceleration Strategies for Gaussian Mean-Shift Image Segmentation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1160–1167, June 2006.

[3] Anchi Cheng, Richard Henderson, David Mastronarde, Steven J. Ludtke, Remco H.M. Schoenmakers, Judith Short, Roberto Marabini, Sargis Dallakyan, David Agard, and Martyn Winn. MRC2014: Extensions to the MRC format header for electron cryo-microscopy and tomography. *Journal of Structural Biology*, 192(2):146–150, November 2015.

[4] Tomasz Danel, Przemysław Spurek, Jacek Tabor, Marek Śmieja, Łukasz Struski, Agnieszka Słowik, and Łukasz Maziarka. Spatial Graph Convolutional Networks. *arXiv:1909.05310 [cs, stat]*, July 2020.

[5] P. Emsley, B. Lohkamp, W. G. Scott, and K. Cowtan. Features and development of Coot. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):486–501, April 2010.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, Palo Alto, CA, 1996. AAAI Press.

[7] Brandon Frenz, Alexandra C. Walls, Edward H. Egelman, David Veesler, and Frank DiMaio. RosettaES: A sampling strategy enabling automated interpretation of difficult cryo-EM maps. *Nature Methods*, 14(8):797–800, August 2017.

[8] Vladimir Gligorijevic, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-Based Protein Function Prediction using Graph Convolutional Networks. *bioRxiv*, page 786236, June 2020.

[9] David S. Goodsell, Christine Zardecki, Luigi Di Costanzo, Jose M. Duarte, Brian P. Hudson, Irina Persikova, Joan Segura, Chenghua Shao, Maria Voigt, John D. Westbrook, Jasmine Y. Young, and Stephen K. Burley. RCSB Protein Data Bank: Enabling biomedical research and drug discovery. *Protein Science*, 29(1):52–65, 2020.

[10] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.

[11] S. R. Hall, F. H. Allen, and I. D. Brown. The crystallographic information file (CIF): A new standard archive file for crystallography. *Acta Crystallographica Section A: Foundations of Crystallography*, 47(6):655–685, November 1991.

[12] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017.

[14] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, February 2017.

[15] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (CASP)—Round XIII. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1011–1020, 2019.

[16] Catherine L. Lawson, Matthew L. Baker, Christoph Best, Chunxiao Bi, Matthew Dougherty, Powei Feng, Glen van Ginkel, Batsal Devkota, Ingvar Lagerstedt, Steven J. Ludtke, Richard H. Newman, Tom J. Oldfield, Ian Rees, Gaurav Sahni, Raul Sala, Sameer Velankar, Joe Warren, John D. Westbrook, Kim Henrick, Gerard J. Kleywegt, Helen M. Berman, and Wah Chiu. EMDataBank.org: Unified data resource for CryoEM. *Nucleic Acids Research*, 39(Database issue):D456–464, January 2011.

[17] Po-Nan Li, Saulo H. P. de Oliveira, Soichi Wakatsuki, and Henry van den Bedem. Sequence-guided protein structure determination using graph convolutional and recurrent networks. *arXiv:2007.06847 [cs, q-bio, stat]*, September 2020.

[18] Dmitry Lyumkis. Challenges and opportunities in cryo-EM single-particle analysis. *The Journal of Biological Chemistry*, 294(13):5181–5197, March 2019.

[19] David N. Mastronarde. Automated electron microscope tomography using robust prediction of specimen movements. *Journal of Structural Biology*, 152(1):36–51, October 2005.

[20] Takanori Nakane, Abhay Kotecha, Andrija Sente, Greg McMullan, Simonas Masiulis, Patricia M. G. E. Brown, Ioana T. Grigoras, Lina Malinauskaite, Tomas Malinauskas, Jonas Miehling, Tomasz Uchański, Lingbo Yu, Dimple Karia, Evgeniya V. Pechnikova, Erwin de Jong, Jeroen Keizer, Maarten Bischoff, Jamie McCormack, Peter Tiemeijer, Steven W. Hardwick, Dimitri Y. Chirgadze, Garib Murshudov, A. Radu Aricescu, and Sjors H. W. Scheres. Single-particle cryo-EM at atomic resolution. *Nature*, 587(7832):152–156, November 2020.

[21] Eva Nogales. The development of cryo-EM into a mainstream structural biology technique. *Nature methods*, 13(1):24–27, January 2016.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., NY, 2019.

[23] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. UCSF Chimera–a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, October 2004.

[24] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Elaine C. Meng, Gregory S. Couch, Tristan I. Croll, John H. Morris, and Thomas E. Ferrin. UCSF ChimeraX: Structure visualization for researchers, educators, and developers. *Protein Science: A Publication of the Protein Society*, 30(1):70–82, January 2021.

[25] Jonas Pfab, Nhut Minh Phan, and Dong Si. DeepTracer for fast de novo cryo-EM protein structure modeling and special studies on CoV-related complexes. *Proceedings of the National Academy of Sciences*, 118(2), January 2021.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, May 2015.

[27] W. H. Roos, I. L. Ivanovska, A. Evilevitch, and G. J. L. Wuite. Viral capsids: Mechanical characteristics, genome packaging and delivery mechanisms. *Cellular and Molecular Life Sciences*, 64(12):1484–1497, June 2007.

[28] Sjors H. W. Scheres and Shaoxia Chen. Prevention of overfitting in cryo-EM structure determination. *Nature Methods*, 9(9):853–854, September 2012.

[29] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, January 2020.

[30] Dong Si, Spencer A. Moritz, Jonas Pfab, Jie Hou, Renzhi Cao, Liguo Wang, Tianqi Wu, and Jianlin Cheng. Deep Learning to Predict Protein Backbone Structure from High-Resolution Cryo-EM Density Maps. *Scientific Reports*, 10(1):4282, March 2020.

[31] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. *arXiv:2006.09661 [cs, eess]*, June 2020.

[32] Gregory Stella. Protein subsets with correlated atomic bond lengths. *Journal of Structural Biology*, 143(2):164–170, August 2003.

[33] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M. Kim. Fast and Flexible Protein Design Using Deep Graph Neural Networks. *Cell Systems*, 11(4):402–411.e4, October 2020.

[34] Christian Suloway, James Pulokas, Denis Fellmann, Anchi Cheng, Francisco Guerra, Joel Quispe, Scott Stagg, Clinton S. Potter, and Bridget Carragher. Automated molecular microscopy: The new Leginon system. *Journal of Structural Biology*, 151(1):41–60, July 2005.

[35] Genki Terashi and Daisuke Kihara. De novo main-chain modeling for EM maps using MAINMAST. *Nature Communications*, 9(1):1618, April 2018.

[36] Thomas C. Terwilliger, Paul D. Adams, Pavel V. Afonine, and Oleg V. Sobolev. A fully automatic method yielding initial models from high-resolution cryo-electron microscopy maps. *Nature Methods*, 15(11):905–908, November 2018.

[37] Rebecca F. Thompson, Matt Walker, C. Alistair Siebert, Stephen P. Muench, and Neil A. Ranson. An introduction to sample preparation and imaging by cryo-electron microscopy for structural biology. *Methods (San Diego, Calif.)*, 100:3–15, May 2016.

[38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *arXiv:1710.10903 [cs, stat]*, February 2018.

[39] Ze Xiao and Yue Deng. Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network. *PLOS ONE*, 15(9):e0238915, September 2020.

[40] Fang Yang, Kunjie Fan, Dandan Song, and Huakang Lin. Graph-based prediction of Protein-protein interactions with attributed signed graph embedding. *BMC Bioinformatics*, 21(1):323, July 2020.

[41] Rafael Zamora-Resendiz and Silvia Crivelli. Structural Learning of Proteins Using Graph Convolutional Neural Networks. *bioRxiv*, page 610444, April 2019.

[42] Wei Zheng, Yang Li, Chengxin Zhang, Robin Pearce, S. M. Mortuza, and Yang Zhang. Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins*, 87(12):1149–1164, December 2019.

[43] Jasenko Zivanov, Takanori Nakane, Björn O Forsberg, Dari Kimanius, Wim JH Hagen, Erik Lindahl, and Sjors HW Scheres. New tools for automated high-resolution cryo-EM structure determination in RELION-3. *eLife*, 7:e42166, November 2018.