Simulating the cellular context in synthetic datasets for cryo-electron tomography

Antonio Martinez-Sanchez[1,*], Marion Jasnin[2,3], Harold Phelippeau[4], Lorenz Lamm[2,5]

May 26, 2023

[1]Department of Information and Communications Engineering, Faculty of Computers Sciences, University of Murcia, Campus de Espinardo 30100 Murcia, Spain
[2]Helmholtz Pioneer Campus, Helmholtz Munich, 85764 Neuherberg, Germany
[3]Department of Chemistry, Technical University of Munich, 85748, Garching, Germany
[4]Thermo Fisher Scientific, Bordeaux, France
[5]Helmholtz AI, Helmholtz Munich, 85764 Neuherberg, Germany
[*]E-mail: anmartinezs@um.es

Abstract

Cryo-electron tomography (cryo-ET) allows to visualize the cellular context at macromolecular level. To date, the impossibility of obtaining a reliable ground truth is limiting the application of deep learning-based image processing algorithms in this field. As a consequence, there is a growing demand of realistic synthetic datasets for training deep learning algorithms. In addition, besides assisting the acquisition and interpretation of experimental data, synthetic tomograms are used as reference models for cellular organization analysis from cellular tomograms. Current simulators in cryo-ET focus on reproducing distortions from image acquisition and tomogram reconstruction, however they cannot generate many of the low order features present in cellular tomograms.

Here we propose several geometric and organization models to simulate low order cellular structures imaged by cryo-ET. Specifically, clusters of any known cytosolic or membrane-bound macromolecules, membranes with different geometries as well as different filamentous structures such as microtubules or actin networks. Moreover, we use parametrizable stochastic models to generate a high diversity of geometries and organizations to simulate representative and generalized datasets, including very crowded environments like those observed in native cells.

These models have been implemented in a multiplatform open-source Python package, including scripts to generate cryo-tomograms with adjustable sizes and resolutions. In addition, these scripts also provide distortion free density maps besides the ground truth in different file formats for an efficient access and advanced visualization. We show that such a realistic synthetic dataset can be used to train generalizable deep learning algorithms.

Keywords: Cryo-Electron Tomography, Deep Learning, Image processing, Scientific Computation, Synthetic data generation

1 Introduction

Cryo-electron tomography (cryo-ET) has emerged as a unique technique to generate tridimensional (3D) representations of the cellular context at sub-nanometric resolution [1]. However, volumetric images obtained by cryo-ET, called tomograms, suffer from high levels of noise and anisotropic distortions, mainly due to the missing wedge [2], which greatly complicate their analysis.

Due to its success in image processing and computer vision in recent years, deep learning (DL) algorithms, and especially convolutional neuronal networks (CNNs) [3], are being adapted to process cryo-ET tomograms

at different steps of the workflow: reconstruction [4], denoising [5, 6], segmentation [7] or macromolecular localization [8]. CNNs enable efficient analysis of local features in images because they take advantage of the inherent regular grid structure of digital images. When the training set is representative and large enough, CNN-based methods outperform classical approaches like template matching (TM) [9]. Nevertheless, the major bottleneck to apply DL in cryo-ET is the lack of ground truth. In most cases it is not possible to generate a reliable labeled dataset for a proper supervised training.

Regardless the method used for segmenting cellular compartments and localizing macromolecules in cryo-tomograms, the next step is usually a quantitative analysis. In cryo-ET, statistical approaches are used to carry out macromolecular censuses [10, 11], analyze the distance between two proteins [12], study protein-membrane distance distribution [13] or prove the co-localization between two macromolecules [14]. The second order statistics are more sophisticated [15], they allow to analyze macromolecular organization on different scales, as well as merge and compare information coming from different tomograms. Second order statistics has recently been used in cryo-ET to evaluate the clustering organization of some protein complexes under different functional states [16], show nanodomains defined by the co-localization of proteins bounded to two aligned membranes [17], describe the liquid-like organization of densely packed proteins by adjusting a statistical mechanics model [18]. In addition, rotations of macromolecules are typically analyzed to determine the rigidity of a bounding, e.g. between two macromolecules [19] or proteins and membranes [20]. Additionally, ad-hoc analysis can be defined to study filament networks [21] and membranes organization [22, 23]. In general, these statistical analyses require a synthetic null-model working as reference [15], which also allows to provide an statistical confidence.

Cryo-ET data acquisition is costly as it requires high-end equipment and a lot of human expertise. Unfortunately, determining the applicability of a dataset for studying a specific structure is only possible after tomogram reconstruction, the last step in the workflow. Synthetic data can be used to streamline imaging parameters as well as to ensure sample appropriateness for solving a specific biological question. Simulations have already been used for particle picking and subtomogram averaging assessment [24, 25]. Recently, a tool specifically designed for membrane-embedded proteins has been proposed [26], however it is limited for a single spherical membrane with uniformly distributed isolated proteins. Simulated annealing and molecular dynamics are used afterwards to improve the packing and avoiding volume overlapping by approximating molecules to spheres [25].

As shown above, simulating synthetic data containing lower order structures present in the cellular environment will contribute greatly to the successful application of DL, to analyze quantitatively the organization of structures in cryo-ET, and assist the acquisition and interpretation of experimental data. There are already computational solutions for generating artificial data in transmission electron microscopy (TEM), nevertheless they focus on modelling the physics of image formation from atomic representations of macromolecules [27, 28, 29]. Methods for simulating cellular environments represent either isolated structures or a mixture of purely random distribution of macromolecules with some spherical vesicles [25, 30, 31]. In [32] some membrane-bound proteins are modelled but just by simple blobs in 2D. A concurrent pre-print [33] generates a more diverse scenario than previous approaches, which includes vesicles and membrane-proteins. However, this approach cannot generate networks of curved filaments and does not allow to control the concentration of structures. More importantly, this pre-print shows that coarse synthetic data already allow to train usable CNN models.

Here, we focus on producing richer contextual information, and directly generate synthetic 3D density maps, also known as phantoms, containing membranes with different combinations of local curvatures, clusters of macromolecules (either cytosolic or membrane-bound), as well as polymers such as microtubules or actin networks. Together with these clean 3D images, motif lists in different formats are provided as ground truth, allowing the generation of micrographs using any external TEM simulator to simulate the distortions introduced by the microscope and 3D reconstruction methods. Moreover, the introduction of generative neural networks holds the promise to produce datasets with more realistic noise and distortion [34], they can reproduce noise and distortions directly from the synthetic micrographs [35], but this is outside the scope of this study. In addition, techniques such as data augmentation [36], knowledge transfer [37] or contrastive learning [31] may allow to train DL-based algorithms without having a perfect representation of the experimental data. For statistical organization analysis, null-models do not require simulating data noise and distortions.

The simulation of high order cellular structures such as organelles (mitochondrion, centriole, ...) is beyond

the scope of this paper. Here we focus on low order structural features present in almost all cellular contexts. Low order features can be generated from parametric mathematical models to control their geometry and organization. In addition, we use parametrizable stochastic models to generate a high diversity of different combinations of geometries and organization to simulate representative and generalized datasets.

The rest of this paper is organized as follows. First, the developed methods are described, we detail how to model the geometry of each structure and their organization, and we explain how these models were implemented. Then, we show examples of synthetic data generated for replicating the cellular context in cryo-ET and use these data to train a generalized deep learning algorithm for semantic segmentation. We also evaluate certain properties of the proposed algorithms. Finally, the proposed methods and the obtained results are discussed. A glossary of acronyms is available in Supplementary Material Table 1.

2 Methods

In this section, we first explain the mathematical models used to parameterize the cellular structures. Then, we describe the algorithms used to generate the synthetic data. Finally, some implementation aspects are detailed because of their relevance. A glossary of the mathematical symbols used in this section is available in Supplementary Material Table 2.

2.1 Structural Models

The cellular context is represented by a density map $D(\mathrm{x}) : \mathcal{V} \subset \mathbb{R}^3 \mapsto \mathbb{R}$ being $\mathrm{x} \in \mathbb{R}^3$ a point and $\mathcal{V}$ the volume of interest (VOI), typically a cuboid whose dimensions are defined by the output tomogram, but in practice it can have any arbitrary shape.

2.1.1 Membranes

Membranes are modelled as parametric surfaces with double Gaussian profile along its local normal. The purpose of this profile is to simulate the lipid bilayer, which in the TEM looks like two parallel and very close electron dense sheets. Although all biological membranes share the same structure, small visual differences may appear due to their distinct molecular composition or simply due to dissimilar imaging conditions. To account for these variations, the membrane profile is controlled by two parameters (see Figure 1): membrane thickness or distance between both layers, t, and the layer variance, σ_l, which controls the thickness of the layers. Below, we describe the parametric surfaces used to model the membrane geometries and how the double-layered profile is incorporated into them.

Sphere: the outer, L_o, and inner, L_i, layers are defined by

$$L^S = \left\{ \mathrm{x} \in \left[B^S(\mathrm{c}) \geq (l^S - 1) \wedge B^S(\mathrm{c}) \leq (l^S + 1) \right] \right\}. \tag{1}$$

Where $B^S(\mathrm{c}) = \|\mathrm{x} - \mathrm{c}\|$, $\mathrm{c} = (c_x, c_y, c_z)$ is the sphere center, $l_o^S = r + t$ and $l_i^S = r - t$ are the values of l for each layer, and r as the sphere radius.

Ellipsoid: As above, the ellipsoidal layers are defined as

$$L^E = \left\{ \mathrm{x} \in \left[B^E(\mathrm{a}) \geq 1 \wedge B^E(\mathrm{a}) \leq 1 \right] \right\}. \tag{2}$$

Where $B^E(\mathrm{a}) = \frac{x - c_x}{a_x} + \frac{y - c_y}{a_y} + \frac{z - c_z}{a_z}$, with $\mathrm{a} = (a_x, a_y, a_z)$ the ellipsoid semi-axis lengths, for each layer the actual values for these lengths are $\mathrm{a}_o = \mathrm{a} + t$ and $\mathrm{a}_i = \mathrm{a} - t$.

Torus: Given a point $\mathrm{x} = (x, y, z) \in \mathbb{R}^3$ the two toroid layers are defined by

$$L^T = \left\{ \mathrm{x} \in \left[B^T(d, e) \geq 1 \wedge B^T(d, e) \leq 1 \right] \right\}. \tag{3}$$

Where $B^T(d, e) = \left(d - \sqrt{(x - \mathrm{c_x})^2 + (y - \mathrm{c_y})^2} \right)^2 + (z - c_z)^2 - e$, with d the major torus radius and e the minor torus radius, which have different values for each layer: $e_o = e + t$ and $e_i = e - t$.

The layers L described above have a non-smooth profile (hard edge). A 3D Gaussian filter $\mathcal{G}$ is thus applied to obtain a smoothed version $L^{\mathcal{G}} \in C^2$ using the layer thickness σ_l defined above:

$$L^{\mathcal{G}} = \mathcal{G}(L, \sigma_l) \tag{4}$$

In addition to representing membranes as scalar fields, which is useful to generate the output density map, it is convenient to also represent them also using 2-manifolds (surfaces):

$$\mathcal{L} = \left\{ \mathrm{x} \in \mathbb{R}^3 \mid \mathrm{x} \text{ is a critical point and } \lambda(\mathrm{x}) > 0 \right\}. \tag{5}$$

Being λ the largest eigenvalue of the Hessian Matrix of the scalar field $L_i^{\mathcal{G}} + L_o^{\mathcal{G}}$ at point x. Since $\mathcal{L}$ is actually the surface in the middle of both the inner and outer layers of the membrane, it can be used as the reference to localize membrane-bound macromolecules and to measure certain geometric properties of the membrane, such areas or local curvatures.

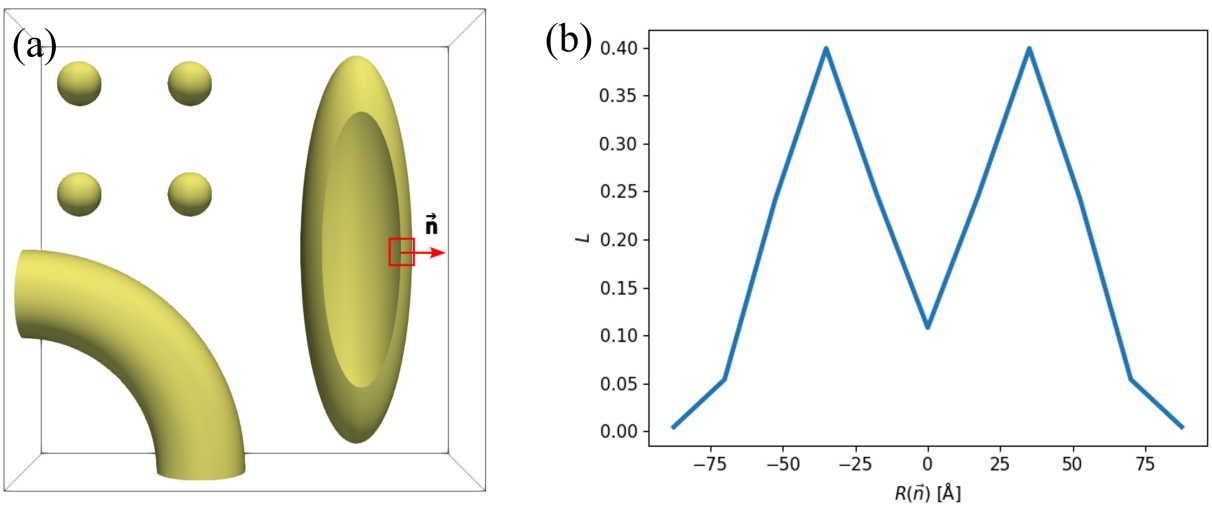

Figure 1: Structural models for membranes. (a) Examples of the three proposed membrane geometries: 4 spherical (top-left), one ellipsoid (right) and one toroid (bottom-left) membrane surfaces. The ellipsoid surface has been cropped on top and a red vector $\vec{n}$ shows the normal direction on a point. (b) $L^{\mathcal{G}}$ density profile on a point on the membrane surface, $\mathrm{x} \in \mathcal{L}$, along its normal, with $t = 35$Å and $\sigma_l = 1$Å.

2.1.2 Filaments

Filaments are modelled as flexible helical polymers. We consider a polymer as a chain of structural unit. A structural units is defined as a density function in 3D $M(\mathrm{x}) : \mathbb{R}^3 \mapsto \mathbb{R}$.

The normalized tangent vector, $\vec{\tau}_i$, at position $\mathrm{x_i}$ in the filament is used to determine the position of the next structural unit in the chain as

$$\mathrm{x_{i+1}} = d_M \cdot \vec{\tau}_i + \mathrm{x_i}. \tag{6}$$

With d_M the distance between two consecutive structural units. Filament centerlines are modelled as helical curves aligned along the Z-axis so the tangent at a point is computed as

$$\vec{\tau}(s) = q\left[g, -f\sin(sq), f\cos(sq)\right]. \tag{7}$$

Where s is the length of the curve from its origin to the position of the point and $q = (f^2 + g^2)^{-1}$. The parameters of the helix curve are the radius f and the slope f/g. These parameters can be assigned from a

given persistence length p, the physical parameter used to characterize polymer rigidity [38], using the following expressions

$$\kappa = \frac{\arccos e^{-d_M/p}}{d_M}, \quad f = \frac{1}{\kappa}, \quad g = h_z \cdot f. \tag{8}$$

Where κ is the curve curvature in the XY-plane and $h_z \in [0, 1]$ determines the curve elevation along the Z-axis as a fraction of the curvature, this giving the curve also a torsion.

After being placed on the filament curve, each structural unit can be additionally rotated around the tangent of the curve by an angle of

$$\alpha = \alpha_0 + i\frac{2\pi h_l^I}{d_M}. \tag{9}$$

With α_0 an arbitrary starting angle to allow randomization, i the position of the structural unit on the filament, and h_l^I the length on the curve required to complete a turn. Consequently, we have two helices (see Figure 2), the curve, or centerline, where the structural units are placed to give curvature and torsion, and the helix formed by the angle between two consecutive structural units, which model a possible helical inner structure of the filament as for actin.

After defining the curve used to model the filament centerline from a given persistence length, the different structural units are inserted on the centerline. Below we present the models for the structural units for the two types of filaments used so far:

- **Microtubules (MT):** the structural unit is modelled as a set of spheres with a radius of 40Å. Structural units are composed of 13 smooth-edged 3D solid spheres placed in a circle of radius 125Å on the XY-plane every $2\pi/13$ radians, thus forming a complete circle as described in [39]. The distance between two consecutive structural units is set to 60Å, which is 1.2 times the unit length in the Z-axis (sphere diameter). MTs do not have a helical inner structure so the turn angle is $\alpha = 0$.

- **Actin:** the structural unit is modelled as a pair of spheres with a radius of 25Å aligned with the X-axis and separated by 50Å. The distance between two consecutive units is set to $d_M = 60$Å, which is 1.2 times the sphere diameter. The turn angle is $\alpha = 2\pi/12.5$ because $h_l^I = 750$Å. This model approximates and averaged actin filament as described in [40].

2.1.3 Macromolecules

Cellular macromolecules can float in the cytoplasmic matrix or be bound to membranes. They can also associate to with filaments, but this case is not yet considered here. In addition, macromolecules can be randomly distributed or aggregated, creating clusters. Here we use the self-avoiding worm like chain (SAWLC) model for polymers [41] to create these clusters. The SAWLC model also allows for random distribution of macromolecules if the polymer size is equal to that of a single macromolecule. Unlike the filamentous polymers described above, we assume here that the polymers have no rigidity and thus the clusters have no specific shape.

Cytosolic: in the SAWLC model each structural unit (or macromolecule), M_i, is placed at a random point of the next set

$$\mathcal{C}_i = \mathcal{S}(\mathrm{x}_{i-1}, d_i) \cap \mathcal{V}. \tag{10}$$

Where d_i is a variable distance from the previous macromolecule M_{i-i} in the sequence, and $\mathcal{S}$ a sphere centered at position, x_{i-1}, and radius d_i.

Membrane-bound: in the SAWLC model each macromolecule, M_i, is placed in random point of the next in the next set

$$\mathcal{M}_i = \mathcal{S}(\mathrm{x}_{i-1}, d_i) \cap \mathcal{V} \cap \mathcal{L}. \tag{11}$$

In comparison with cytosolic macromolecules, membrane-bound proteins must also be embedded in a membrane set (see Section 2.1.1 Equation 5) $\mathcal{L}$.

Heterogeneous clusters: a SAWLC cluster, either cytosolic or membrane-bound, can contain J different types of macromolecules, M^j, with $j \in [1, J] \subset \mathbb{N}$.

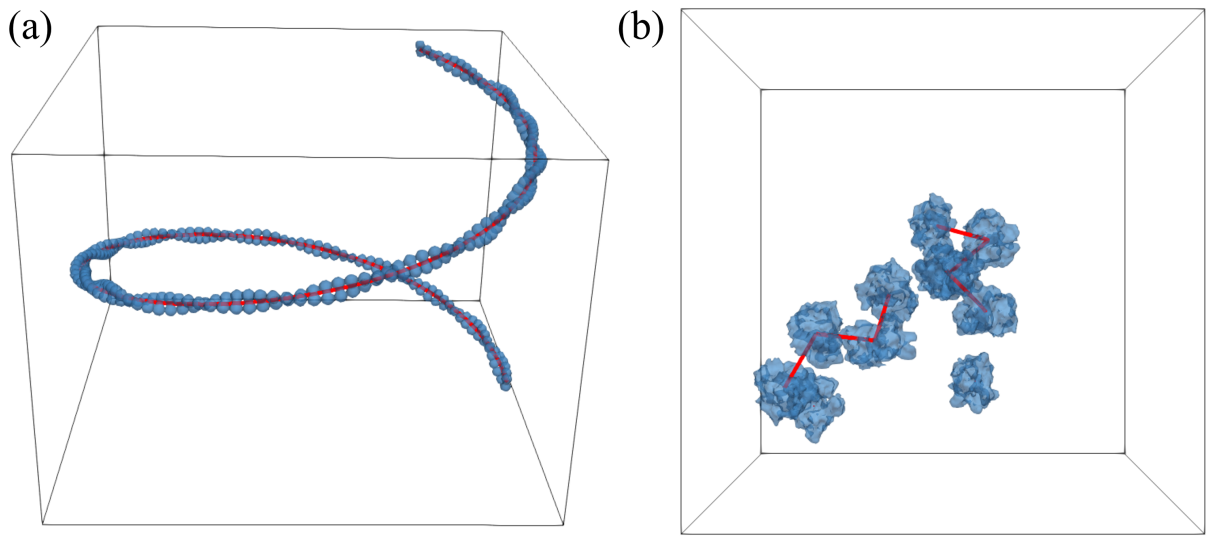

Figure 2: Polymeric models. (a) Helical model used for actin filaments. The first order helix defining the filament centerline is in red. Here curvature and torsion are set to $p = 1000$Å and $h_z = 0.2$ just for visualization purposes, the Z-axis corresponds to the vertical dimension of the cube in the 3D view. In transparent blue the structural units (M_i, two spheres); the turning angle between two consecutive structural units, which is computed from $h_l^i = 750$Å, defines the twist of the inner helix around the centerline. (b). Volume with randomly distributed homogeneous SAWLC clusters as well as isolated macromolecules, red lines represent the sequence of macromolecules forming each SAWLC cluster, transparent blue shows the iso-surface of each macromolecule density (M_i, Protein Data Bank (PDB) code 5mrc).

2.2 Transformation Models

Each new inserted structural unit has a specific rotation, R, with respect to the reference structural model and a specific translation, T, to its final position in D. Depending on the structure type, the rigid transformations R and T are generated according to different stochastic models (transformation generators G^T), thus generating a representative variety of structures with different positions and rotations. Similarly, the geometric parameters for membranes and filaments are also generated using stochastic models (structural generators G^S).

2.2.1 Membranes

Translations and rotations are generated in the same way for all membrane varieties. Translations are expressed with a coordinates vector generated according to the complete spatial randomness (CSR) model [42, 43]. Rotations are expressed as unit quaternions and are generated using the algorithm described in [44] to approximate a uniform random distribution.

The membrane thickness t and layer thickness σ_l parameters are obtained from a random uniform distribution in a specified range.

Sphere: the radius r is generated from a random uniform distribution within the following range: the minimum corresponds approximately with the smallest vesicles in cells (typically 75Å) and the maximum with the diameter of $\mathcal{V}$.

Ellipsoid: the lengths of the semi-axis $\mathrm{a} = (a_x, a_y, a_z)$ are taken from random uniform distributions within the same range as the spheres. In addition, the eccentricities are defined as

$$\epsilon_{xz} = \sqrt{1 - \left(\frac{a_z}{a_x}\right)^2}, \quad \epsilon_{yz} = \sqrt{1 - \left(\frac{a_z}{a_y}\right)^2}, \tag{12}$$

and must be smaller than a given value.

Toroid: the two radii are obtained using a uniform random distributions similarly to sphere radius, then the greater value is assigned to the major radius d and the lower to the minor radius e.

2.2.2 Filaments

Unlike membranes, which are independent structures, filaments consists of a sequence of several structural units. Consequently, the rigid transformations for the structural unit, M_i, depend on those applied to the previous unit, M_{i-1}. Therefore a random generator for rigid transformations is only required for the first structural unit, M_0, using the same distributions used for membranes.

For the filament geometry expressed as a helical curve, the only free parameters are the persistence length p, which controls the circular radius (and curvature) in the XY-plane, and the elevation in the Z-axis as a fraction of the circular radius h_z, which controls the torsion. The values of p are derived from a random exponential unit distribution added to a minimum value of persistence length, this value sets a maximum to the filaments flexibility. The values of h_z are obtained from a random uniform distribution between a given value (≤ 1) and 1.

In the case of actin-like filaments, it is possible to create a branched network, instead of generating isolated filaments. In a branched network, each new filament may start from any location in the VOI, like regular filaments thus starting a new network, but with a certain probability, $P_b \in [0, 1]$, it starts from a random point of an already inserted filament, creating a branch within an already existing network. These parameters thus control the average number of networks to be generated for a given maximum occupancy and the branching density (number of branches per network length).

2.3 Algorithms

This section contains the description of the different algorithms used to insert the structures into the VOI, $\mathcal{V}$. First, the order chosen to process each structural type seen above is listed. Then, the algorithms for inserting each specific structure are described.

2.3.1 Structure type selection

Algorithm 1 details the procedure for processing the structural models. Only after all the structures of one type have been inserted, then we move to the next type. In each type, the order of each entity, e.g. membrane geometry (spherical, ellipsoidal or toroidal), is an input that can be controlled by the user. The criterion used to establish the order specified in Algorithm 1 is to place potentially larger and more rigid structures first. We assume that a cluster of small macromolecules following the flexible SAWLC model will be inserted more easily than a membrane or a filament in a highly fragmented volume.

Algorithm 1 Order of insertion for the different types of structures.

Require: VOI; lists of structural models (one per type)
Ensure: Density map; list of inserted structures; output VOI (empty space after inserting of all structures)
  Insert membranes using Algorithm 2
  Insert filaments using Algorithm 3
  Insert cytosolic macromolecules using Algorithm 3
  Insert membrane-bound macromolecules using Algorithm 3

2.3.2 Structure insertion

The procedure for inserting membranes is described in Algorithm 2, for filaments and macromolecules this procedure is identical and explained in Algorithm 3. Both algorithms are based on the trial-and-error principle: one tries to insert a new instance of a structure according to some geometrical constraints defined by the structural generator G^S into a space defined by the transformations generator G^T, if this is not possible due to overlap with already inserted structures or because a fraction of the structure is outside of the VOI, then the algorithms try again by generating new geometric and transformation parameters. An additional stopping criterion is added to ensure the end of the process after a certain number of failed tries. Information about each structure is entered in the form of an input list Q including: the structural unit M, the target occupancy O and the statistical models for generating the structural parameters G^S and the rigid transformations G^T. Occupancy is the percentage of a volume occupied by one or more structures.

In Algorithm 2, for each membrane model i in the list Q, an instance M with parameters generated by G^S is inserted into the updated VOI, $\mathcal{V}_o$, according to the rigid body transformations R and T generated by the transformation generator G_T. The insertion loop for a membrane model i ends when the target occupancy O_i is reached, and then the algorithm proceeds to the next membrane model in the list $i+1$ until all models have been processed. The input $\mathcal{V}$ is updated for each inserted membrane to exclude the already occupied volume. Therefore, $\mathcal{V}_o$ will define the final empty space and corresponds to the VOI input for the next algorithm, which will process the next structure type.

Algorithm 2 Insertion of membranes in the VOI.

Require: VOI, $\mathcal{V}$; list of membrane models, Q^M, each entry i is a tuple (O_i, G_i^S, G_i^T); N_t, maximum number of trials for the insertion of a membrane instance.
Ensure: D, density map; Q_o, list of inserted structures; $\mathcal{V}_o$, output VOI.

```
V_o ← V
for i ← 1 to length(Q^M) do                          ▷ Loop for membrane models
    n_f ← 0
    (O_i, G_i^S, G_i^T) ← Q_i^M
    O = 0
    while (O < O_i) ∧ (n_f < N_t) do                 ▷ Add membranes with model G_i^S until O_i is reached
        Generate membrane M from G_i^S
        Get T and R from G_i^T
        M' ← M(R(x) + T)                             ▷ Rigid transformation
        if V_o ∩ M̌' = M̌' then                        ▷ M̌ is the volume occupied by M
            V_o ← V_o ∪ M̌'
            O ← O + 100 · (volume(M)/volume(V_o))
            Add M' to D
            Add M' to Q_o
            n_f ← 0
        else
            n_f ← n_f + 1                            ▷ Update the number of consecutive fails
        end if
    end while
end for
```

Regardless of overlapping, membrane instances are independent of each other. However, for filaments and macromolecules the insertion of the next structural unit depends on the previous one, except for the first one. Algorithm 3 is an extension of Algorithm 2 to handle this type of situation. It thus includes three nested loops. The first one iterates over all the introduced structural models. The second one attempts to reach the requested occupancy by inserting structural instances of a filament or cluster. The third one grows the filament or cluster by adding structural units.

Algorithm 3 Insertion of filaments or macromolecules in the VOI.

Require: VOI, $\mathcal{V}$; list of models, Q^S, each entry i is a tuple (O_i, G_i^S, G_i^T); N_t, maximum number of trials.
Ensure: D, density map; Q_o, list of inserted structures; $\mathcal{V}_o$, output VOI.

```
V_o ← V
for i ← 1 to length(Q^M) do                                   ▷ Loop for models
    n_f ← 0
    (O_i, G_i^S, G_i^T) ← Q_i^S
    O = 0
    while (O < O_i) ∧ (n_f < N_t) do                          ▷ Loop for filaments or clusters
        S_i ← ∅                                               ▷ Empty filament or cluster
        Generate seed structure unit S_i^0 from G_i^S(∅)
        j = 0
        Get T and R from G_i^T
        (S_i^j)' ← S_i^0(R(x) + T)
        while V_o ∩ (Š_i^j)' = (Š_i^j)' do                    ▷ Loop for structural units
            V_o ← V_o ∪ (Š_i^j)'
            O ← O + 100 · (volume(S^j)/volume(V_o))
            Add (S_i^j)' to S_i
            Add (S_i^j)' to D
            j ← j + 1                                         ▷ move to next structural unit
            Generate next structure unit S_i^j from G_i^S(S_i^{j-1})
            Ŝ_i^j ← S_i^j(R(x) + T)
        end while
        if S_i = ∅ then
            n_f ← n_f + 1
        else
            Add S_i to Q_o
            n_f ← 0
        end if
    end while
end for
```

2.4 Implementation

In this section we explain how the above models and algorithms are implemented. First, we explain the numerical approximations needed to handle discrete data such as digital images or polygonal representations. Next, we describe the format of the input and output data. Finally, we list the software implemented and used.

2.4.1 Numerical aspects

The VOI, $\mathcal{V}$, is in practice represented as a binary discrete 3D image, where the value 1 represents the available empty space and the value 0 the forbidden space, either already occupied or discarded by the user as input. This representation allows the introduction of VOIs representing spaces with irregular boundaries such as the Golgi apparatus, vesicles, endoplasmic reticulum, etc. The voxel size of the VOI images is defined as the resolution of the output data. Since this image is binary its impact in terms of memory space is low.

Macromolecules are represented by 3D images encoding the electron densities of a single instance. They can be generated from a deposited atomic model, directly from the electron densities obtained by cryo-EM/ET, the public databases are PDB [45] and EMDB [46]. All input reference macromolecules are represented by a 3D grayscale image with the same voxel size. In general, the reference images of macromolecules are relatively small, so applying rigid transformations on these images is not computationally demanding. In contrast, membranes are represented by 3D images generated directly from mathematical models. Therefore, no reference image of the electron density is needed, but they can cover the entire VOI, so applying rigid transformations on these images is computationally expensive. On the bright side, the requested occupancy is usually achieved with a few instances. Filaments are an intermediate case, they are constructed from multiple instances of the same structural unit. At the same time, each structural unit consists of copies of the same monomer (solid spheres with smooth edges). Although our current implementation uses only spherical densities to represent individual monomers of microtubules and actin filaments, we believe that this is a good approximation for nanometer resolution and represents a generic filamentous structure well. If subnanometer resolution is required, atomic models for actin and tubulin proteins should be used instead.

The most common and computationally intensive operations in the above algorithms are the rigid transformations $S(R(\mathrm{x}) + T)$ and the evaluation of the overlap volume $\mathcal{V} \cap S$. To reduce the computational cost of applying rigid transformations on the 3D images, a polygonal representation of the surface is extracted for each structural unit. Isosurface algorithm [47] is used to obtain a closed surface from the 3D density map. Rigid transformations on surfaces are more efficient than on 3D images. The evaluation of volume overlap ensures the exclusion of volumes between different structures. This is done by sampling the surface mesh and evaluating the fraction of these samples outside of the VOI, $\in \bar{\mathcal{V}}$. For more flexibility, a threshold can be adjusted to define the fraction of the area outside the VOI that should be considered overlapping. This approach is particularly effective when a simulation has a high concentration of structures (the cumulative input occupancy is high). The insertion procedure is based on a trial-and-error algorithm, so the probability of failure is very high in fragmented volumes with high occupancy (see Section 3.2).

The membranes comprise a single structural unit determined entirely by the model parameters and rigid transformations. In filaments each new structural unit is placed in a sequence determined by model parameters for an specific filament. However, in macromolecules, each new structural unit (macromolecule) in the sequence can have any position around a sphere of a certain radius centered on the previous structural unit. For uniform random sampling of the sphere, we use the algorithm in [15] (Algorithm S2).

The electron density of the macromolecules is determined from their atomic models, here we used Chimera and EMAN2 software [48, 49]. Because we generate membranes and filaments solely from geometrical models, we added multiplicative coefficients to the stochastic generators to adjust the contrast of the membranes and filaments based on the macromolecules and to add some variability in the contrast.

2.4.2 Input and output data

By default, the input VOI is the whole tomogram, but it is also possible to specify a smaller centered bounding box. For more sophisticated VOIs, an arbitrary shape is introduced via a binary mask, which must have the same size as the tomograms to generate.

The structures to be simulated are specified by four different types of input files: membranes, *.mbs*; filaments, *.hns*; cytosolic macromolecules, *.pbs*; and membrane-bound macromolecules, *.pms*. Of course several files of the same type are possible, e.g. spherical and toroidal membranes, actin-like and microtubule-like filaments, different classes of macromolecules (ribosomes, proteasome, etc). In general, these files determine the target occupancy, the structural unit models (filaments and macromolecules) and the parameters for the generators. Supplementary Material Table 3 contains a description of the parameters included in each file type.

The output files provided by the simulator are the following:

- The density map D, or phantom, a clean version of the 3D tomogram (without noise and missing wedge) at the input resolution in MRC format (see Figure 3.a).
- A motif list as a CSV file with the position, rotation and identity for each structural element in the tomogram. This is usually a large file because filaments are curves and membranes are surfaces. For macromolecules, we have one entry per particle, but for filaments and surfaces, we have as many entries as there are points in the corresponding curve or surface.
- A ground truth tomogram with labeled voxels, so-called label field, which corresponds to the label values in the CSV files (see Figure 3.b).
- Ground truth in VTK format for polygonal data, one for each simulated density map. VTK is a more sophisticated format than MRC, but it allows for more efficient processing and visualization (see Figure 3.c).
- Another ground truth VTK file but containing the skeletons instead of the 3D shapes (see Figure 3.d).
- The micrographs in MRC format, 2D projections of the simulated tomogram at the specified tilt angles. Noise and misalignment offsets are added to these micrographs.
- The 3D reconstructed tomograms at the input resolution. In addition to noise and misalignment, these data include missing wedge distortions (see Figure 4).

Although this paper is dedicated to simulating cellular structures rather than reproducing the 3D image acquisition process in TEM, the implementation presented here is capable of reconstructing 3D tomograms containing some of the basic distortions inherent to cryo-ET like noise, micrograph misalignment and missing wedge. The micrographs are generated as 2D projections at specified tilted angles using IMOD [50]. The noise is modelled by a Gaussian distribution, which is added to the micrographs, in order to approximate the tomogram contrast for a certain signal-to-noise ratio (SNR). Since the ground truth is known, noise level is adjusted to achieve certain SNR. Target SNRs are randomly generated from a uniform distribution within a specified range. The misalignment of the micrographs is modeled by applying random offsets on the X and Y axes of the micrographs in a sinusoidal model to penalize high tilt angles. Finally, the 3D tomogram is reconstructed from the micrographs using IMOD.

There are more accurate tools than IMOD for reconstructing cryo-tomograms from synthetic data [28, 26]. They have more complex models for noise, which also take into account solvent contribution. Since simulating the TEM process is still an active research field, here we generate enough output information (shape, position and rotation for each structural unit) to feed a TEM simulator. For example, very recently a software based on generative neuronal networks has been published to reconstruct tomograms from the synthetic micrographs [35] having distortions with realistic appearance.

2.4.3 Code

The software implementation of the models and algorithms described here is in a Python 3 package called 'PolNet' open-source available in the public repository https://github.com/anmartinezs/polnet.git.

This code relies on the 'NumPy'/'SciPy' libraries [51, 52] for 3D image manipulation and processing, except for rotations which are performed by the 'JAX' library [53] when GPU hardware is available to further accelerate data generation. PolNet uses the 'mrcfile' [54] library for input/output tomograms, and IMOD [50]

for their reconstruction from micrographs, VTK for polygonal data manipulation [55] and 'Paraview' [56] for their visualization. All the graphs in this paper were generated using the 'matplotlib' library [57].

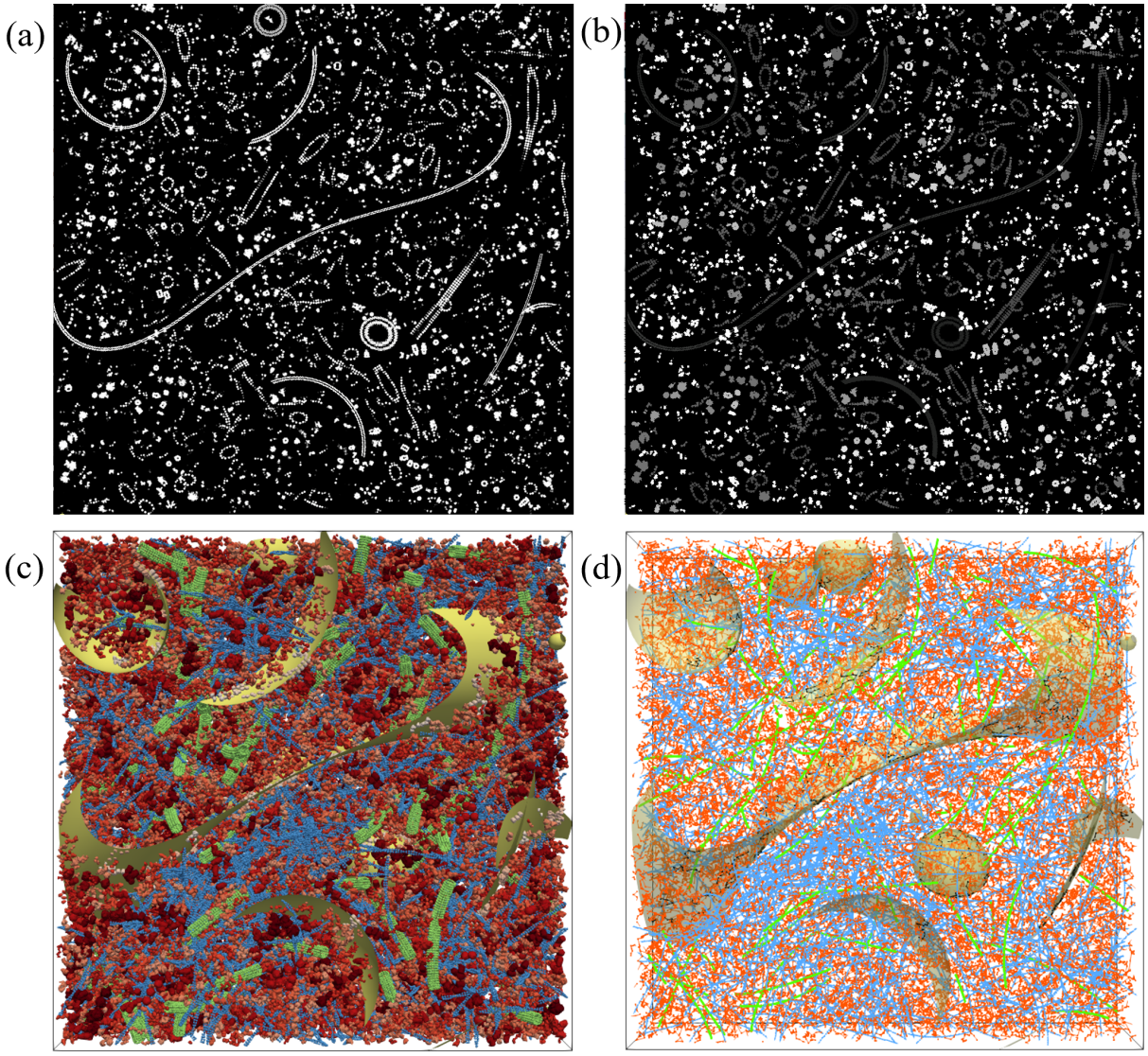

Figure 3: Simulation of the cellular context. (a) A 2D slice in the XY-plane of the generated density map. (b) Labeled (or segmented) ground truth map, each type of structure has a different label, background is 0. (c) Polygonal ground truth data, membranes in yellow, actin network in blue, microtubules in green, macromolecules (included membrane-bound) in different shades of red. (d) Ground truth skeleton data, filamentous structures (actin networks and microtubules) are represented by their centerlines, macromolecules by points interconnected by lines defining the clusters, isolated points represent non-clustered macromolecules, here membrane-bound clusters are depicted by black lines.

3 Results

Here we present synthetic data generated to show the potential of the proposed approach. All data were generated using a workstation running Ubuntu 22.04 LTS with an Intel^{R} Core^{TM} i7-11700 @ 2.50GHz processor, 128GB RAM and a GeForce RTX^{TM} 3090 Ti GPU.

3.1 Cellular cryo-ET

The goal of this experiment is to simulate an exemplary tomograms with all currently available types of structures: membranes (spherical, ellipsoidal and toroidal), cytoskeleton (actin-like networks and microtubules), cytosolic and membrane-bound macromolecules. Generators settings and a complete list of the used structures, including macromolecular models from the PDB, are available in Supplementary Material Tables 3 and 4 respectively. The size of the generated tomograms is 1000x1000x250 voxels, with a voxel size of 10Å, which is the typical configuration used today for tomogram visualization, molecular localization, and segmentation in cryo-ET.

Figure 3.a shows an example of a generated clean density, free of noise and missing wedge. Figure 3.b shows the associated ground truth segmentation, which labels all voxels in each entity with the integers in Supplementary Material Table 4. The equivalent information is also stored in VTK format (see Figure 3.c), which is more suitable for 3D visualization. In addition, the skeletons of the structures are also stored in VTK format (see Figure 3.d), with filaments represented by their centerlines and clusters of macromolecules represented by a sequence of points connected by lines, with single points consequently corresponding to single macromolecules.

Figure 4.a shows a 3D reconstructed tomogram from 2D projected micrographs ($\pm 60°$) that were randomly misaligned using a sinusoidal random model to penalize high tilt micrographs, with a mean value of 1 and a maximum misalignment of 1.5 pixels. As a result, the reconstructed tomogram suffers from distortions due to angular sampling and missing wedge. Gaussian noise was added to the micrographs to simulate the low contrast of the cryo-ET data by setting a target SNR randomly in the range $[1, 2]$. Figure 4.b shows the missing wedge effect by comparing the reconstructed tomogram with the ground truth. The missing wedge effect is particularly visible in the YZ-slice where the membrane disappears and the macromolecules are elongated on the Z-axis. The reconstructed and ground truth tomograms can be used as input information to train/test machine learning algorithms for image segmentation.

This example shows how Algorithm 1 can merge the different types of structures to generate a complex environment, which resembles the low order structures present in the cellular context.

3.2 Simulation of the crowded environments

The cell cytoplasm and membranes usually have a high concentration of structures. Since structure insertion is based on a trial-and-error algorithm, the execution time increases with the concentration or occupancy. To evaluate structural insertion algorithms, here we run experiments to evaluate the insertion of cytosolic macromolecules, membrane-bound proteins and filaments separately. The membrane insertion algorithm is not evaluated because the total membrane occupancy generally does not exceed 1%. In these experiments, all tomograms had a size of 500x500x200 voxels, with a voxel size 10Å.

Regardless of the type of structure, the input occupancy is a target value. A trial-and-error algorithm cannot guarantee convergence within a certain time, which is especially difficult when the occupancy is very high. For macromolecules, Algorithm 3 terminates after failing N_t consecutive times to initialize a new structure. Increasing N_t therefore allows to obtain tomograms with higher densities at the cost of higher running times due to a greater number of structure insertion failures (see Figure 5.a-c). An $N_t = 100$ achieves a target occupancy of 25% for cytosolic macromolecules (see Figure 5.d-f), which already corresponds to a very crowded tomogram. Increasing $N_t = 1000$ allows to reach nearly 30% at the cost of approximately doubling the time.

In contrast to cytosolic macromolecules, for which occupancy is calculated as the volume fraction (in percent) of the occupied VOI, the target occupancy for membrane-bound macromolecules represents the maximum

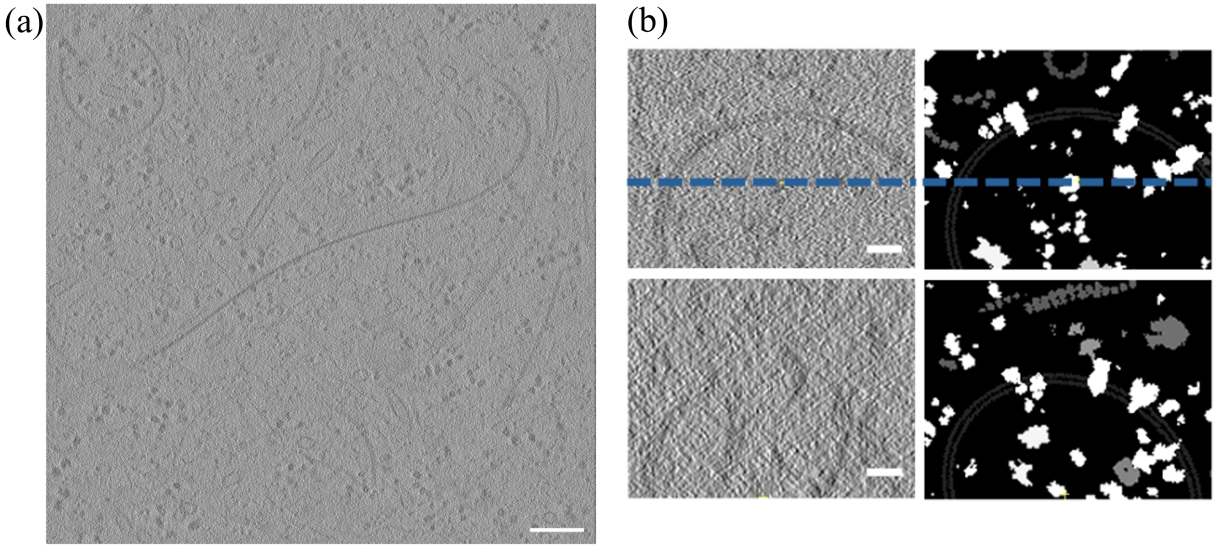

Figure 4: Reconstruction of tomograms. (a) A 2D slice in the XY-plane of a reconstructed tomogram corresponding to the density shown in Figure 3 with an $SNR = 1.84$. Scale bar 200 nm. (b) Zoom of the XY-plane showing membrane-bound proteins in the upper left, the label map is in the upper right, the dashed lines show the intersection with the YZ-plane slices shown in the bottom row. In the reconstructed YZ-slice (bottom-left) the membrane disappears at the apex and the macromolecules are elongated in the Z-axis relative to the ground truth (bottom-right) due to the missing wedge distortion introduced during reconstruction. The bilayer profile of the membrane can be appreciated in the magnified images. Scale bars 20 nm.

area fraction (in percent) of the membrane surface occupied, as membrane-bound macromolecules can only be inserted on the membrane surface. Therefore, the input occupancy does not match the measured volume occupancy (see Figure 6.a), although their dependence is approximately linear. Nanodomains, or clusters, of membrane-bound proteins are evident at low occupancies (Figure 6.b). It is also possible to generate membranes with a high density of membrane-associated proteins, as in synaptic vesicles, viruses or endoplasmic reticulum (see Figure 6.c).

As for cytosolic macromolecules, the occupancy is computed from the filament volume. Occupancy values greater than 10% are not necessary because, in general, they do not correspond to realistic conditions in the cell. Figure 7.c shows an occupancy of 7% which already results in a very dense actin network. An important parameter to control the shape of the filament networks is the branching probability for the filament P_b (see Figure 7.a). The contrast between Figures 7.c and 7.d shows that a high P_b value generates a densely concentrated filament network in specific regions, while conversely, a low P_b value generates a more evenly distributed network along the VOI.

3.3 Training a deep learning model

To assess the usability of our synthetic data for training deep learning algorithms, we use a specific implementation of the U-Net architecture known as nnU-Net [58]. This framework is particularly known for its good performance without the need of manual hyper parameter tuning in the field of semantic segmentation for biomedical images. We trained nnU-Net to segment cryo-electron tomograms into the classes of membranes, microtubules, actin filaments, ribosomes, and background (i.e., non of the other classes). To this end, we utilized the dataset presented in Subsection 3.1, and used ten reconstructed tomograms (see an example in Figure 4.a) for training. The tomograms were presented to the network in patches of size 160^3, extracted from all ten tomograms without overlap. These patches were split into 80% for the training set and 20% for the validation set. For the model design, we chose nnU-Net 3D full resolution option, and the default *nnUNet-*

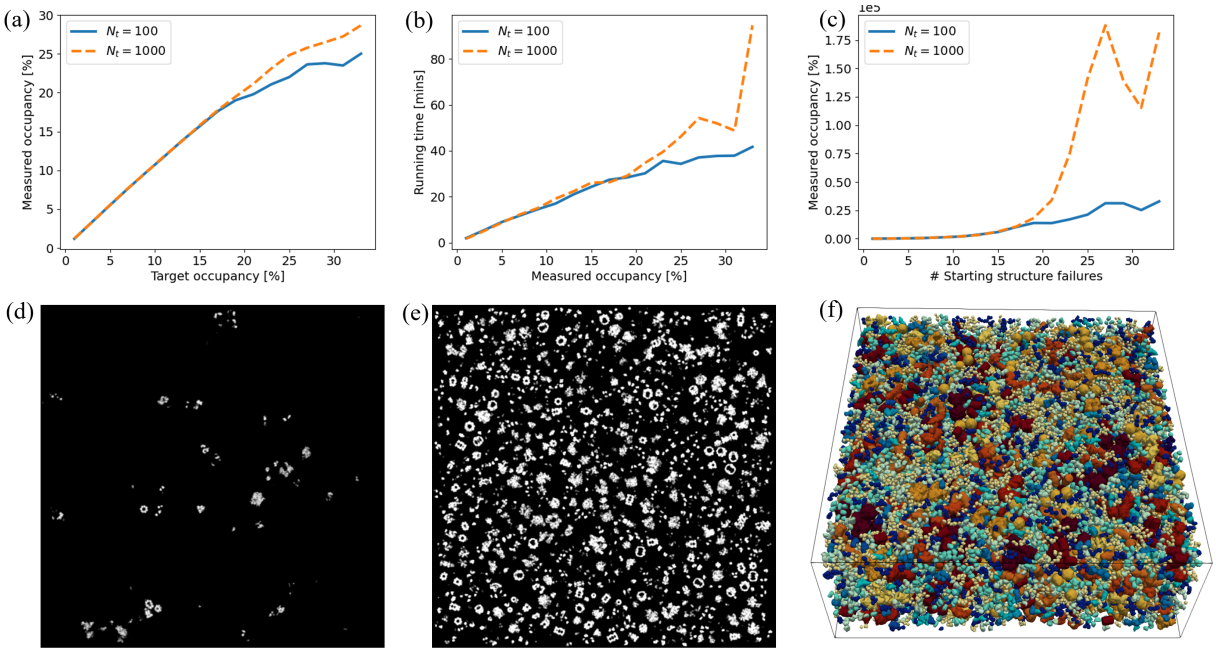

Figure 5: Simulation of cytosolic macromolecules. (a-c) Macromolecule insertion performance for different target occupancies where two N_t values are evaluated. (a) Occupancy measured as the fraction of voxels belonging to macromolecules relative to the VOI volume. (b) Execution times. (c) Number of unsuccessful attempts to start a new macromolecule cluster. (d) 2D slice of a tomogram with 1% occupancy. (e) 2D slice of a tomogram with a 25% occupancy. (f) 3D view of the tomogram with a 25% occupancy.

TrainerV2 trainer class for performing the optimization. As depicted in Figure 8, the resulting prediction from our network validates our approach. Despite being trained solely on synthetic data, the network was capable of coarsely, but effectively, segmenting the main features within two *in situ* tomogram from different samples.

4 Discussion

Although higher order structures such as organelles (mitochondrion, centriole, ...) or the nucleolus require specific and very sophisticated models, the simulation of low-order structures proposed here can be used to study many interesting cellular processes such as membrane morphology [22], filament interaction under pathological conditions [21], macromolecule clusters [16], macromolecules in the liquid-like state [18] and membrane-bound protein nanodomains [60, 17]. In addition, local feature simulation has been shown a path to overcome the current limitations of deep learning approaches for processing (segmentation, macromolecular localization, ...) cryo-tomograms [31, 33].

We proposed different structural models for the most frequent features in the cell visualized by cryo-ET. For the membranes, we propose a uniform parametric model for their double-layer profile, and three different geometries for their surfaces, the aim of reproducing the different density profiles of the membrane and all possible combinations for the local curvatures in the cell. Regarding filamentous structures, we propose two models for the most abundant structures of the cellular cytoskeleton: one for microtubules and the other for actin networks which can include branches connecting filaments. These models are based on a helical curve that can be parameterized from the physical parameter persistence length to generate any possible curvature and torsion. Although they do not reproduce actin and microtubules at subnanometer resolution, they should be good enough to train or validate generalized algorithms for tracing these structures. They can be extended in the future to reproduce high resolution detail using the atomic model of actin and tubulin. Macromolecules

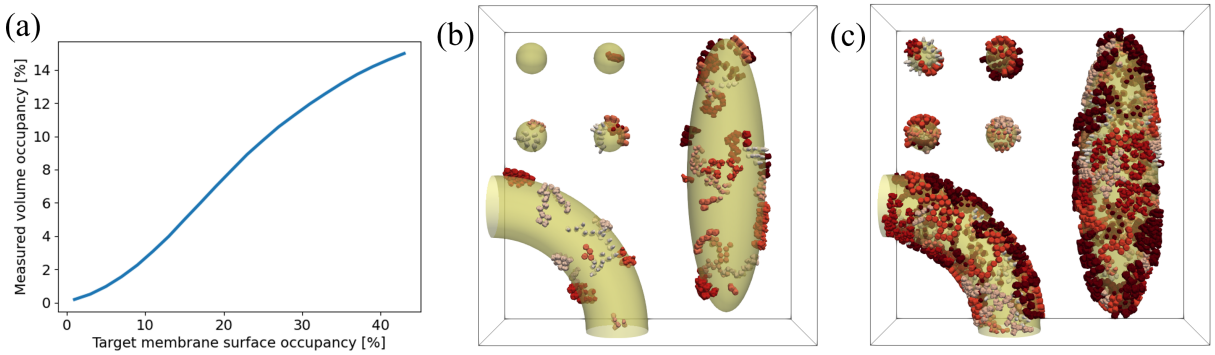

Figure 6: Simulation of membrane-bound proteins. (a) Measured output volume occupancy compared to the input target occupancy on membrane surfaces. (b) 3D visualization of a tomogram with a 7% target occupancy for membrane-bound proteins. (c) Target occupancy of 39%. (b-c) Membrane surfaces in transparent yellow, membrane proteins in different levels of red. Membrane geometries are taken from Figure 1.

can be represented with arbitrary accuracy from their atomic models; they may be cytosolic or membrane-bound. Furthermore, we use the SAWLC model to form homogeneous or heterogeneous clusters, as well as isolated macromolecules, of densely packed macromolecules such as in the cellular cytoplasm and its membrane organelles.

In addition to structural models, stochastic transformation models allow: (1) to obtain a wide variety of shapes for membranes and filaments by randomly sampling their geometric parameters, (2) to generate rigid transformations, rotations and translations, in order to randomly place the structures in their final location.

Reproducing the densely packed cellular context is not a trivial task. Here we propose trial-and-error based algorithms to insert randomly distributed structures. The insertion of structures is ordered by type, with the criterion being to insert the smallest and most flexible structures at the beginning, as it is easier to find a slot for them in a highly fragmented volume. The maximal occupancy that can be achieved depends strongly on the structure to be inserted but has been shown sufficient to represent the cellular context.

Users can control the input parameters of stochastic models to modulate certain output properties such as the occupancy of each structure or the ranges of their geometric properties. The output data includes different formats to efficiently perform tasks such as visualization, statistical analysis, or training machine learning algorithms. Although the software proposed here contains minimal functionality to reproduce the most important distortions introduced during tomographic reconstruction for cryo-ET, it produces enough output data to use more advanced TEM simulators. The implemented software is an open-source Python package, thus allowing for further extensions for structural models; it also contains the scripts necessary to generate the results presented here. 3D image rotation is probably the most computationally demanding routine, and it is widely used by the proposed algorithms. Two strategies have been applied to reduce the computational cost of structure rotation: (1) if possible, we replace image rotations with surface rotations, which is a faster operation, (2) otherwise we use GPU hardware acceleration.

When a dataset consisting of many tomograms is required, the target occupancy of each structure and the parameters of the geometric models are randomly taken from manually adjusted input values to produce a representative dataset with high variability. Several experiments have been conducted to evaluate the performance of the insertion algorithms to achieve a high concentration of different structures. With the inclusion of a limit for the number of trials, it is possible to generate tomograms with occupancy rates close to 30% for macromolecules in an affordable amount of time (less than 1 hour per tomogram). In the case of actin networks, the branching probability controls how the network is distributed along the tomogram.

Finally, we demonstrate that our synthetic data approximate well many of the features present in experimental *in situ* tomograms. Specifically, the information contained by the synthetic data is sufficient to train a generalized deep learning model to coarsely, but effectively, segment most of the features present in real cellular tomograms.

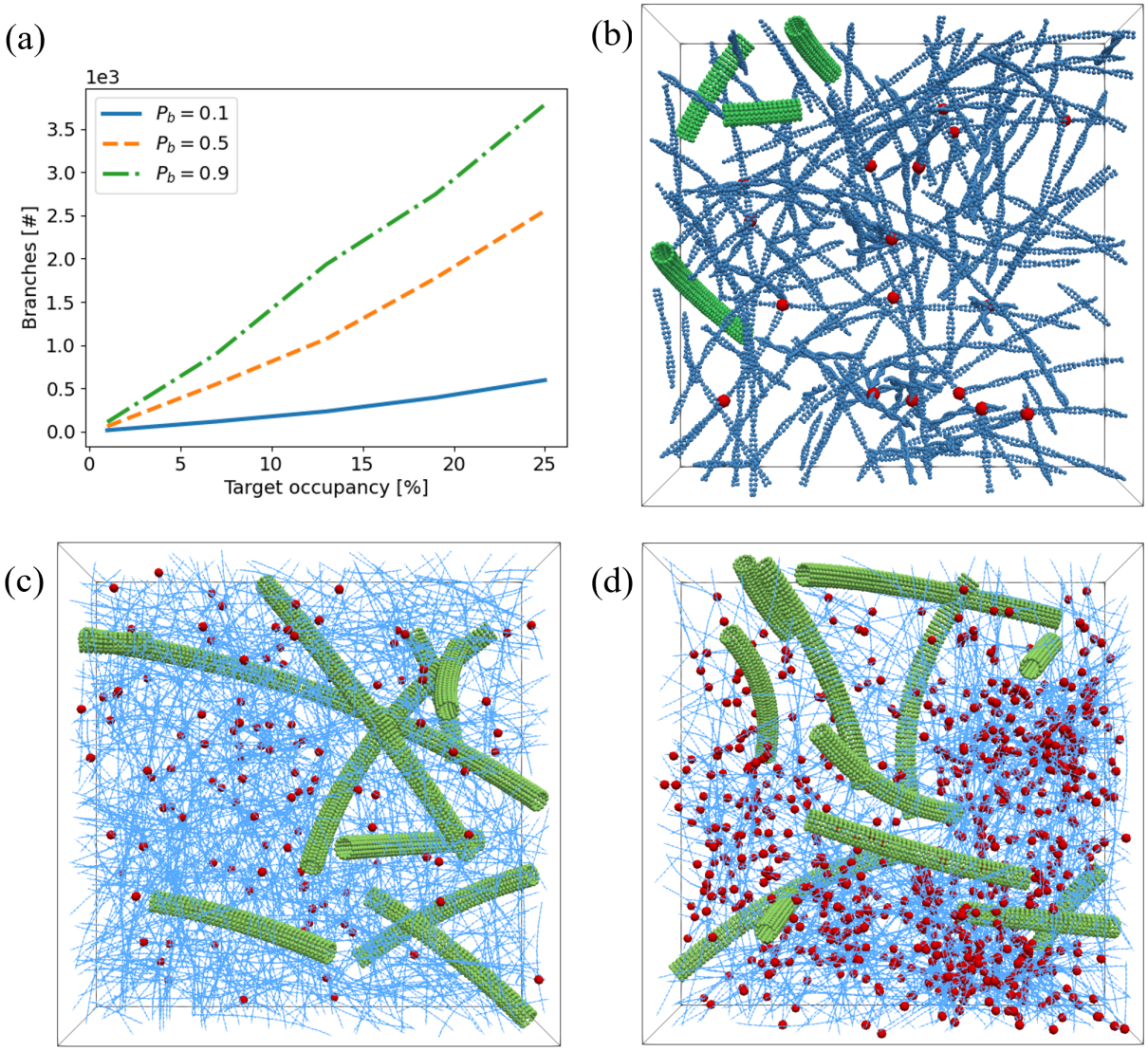

Figure 7: Simulation of filament-like structures. Target occupancy of 10% for microtubules and 90% for the actin filament network. (a) Number of actin filament branches for different occupancies and P_b. (b) 3D visualization of a tomogram with 1% occupancy and $P_b = 0.1$. (c) 7% occupancy and $P_b = 0.1$. (d) 7% occupancy and $P_b = 0.9$. In (c-d) only the centerlines are shown for actin filaments, instead of the filaments shape as in (b), to allow a better visualization of the generated crowded network. (b-d) Actin filaments in blue, actin filament branches as red spheres and microtubules in green.

Acknowledgments

We thank Prof Dr Juergen Plitzko at the Max Planck Institute of Biochemistry for sharing computational resources needed for software development. We thank Tingying Peng from Helmholtz AI for mentoring L.L. We thank Jonathan Schneider for providing tomogram in Fig 8.b. This work was supported by Ramon y Cajal program, RYC2021-032626-I, of the Spanish State Research Agency (AEI) and Attract-RYC 2023 program of the University of Murcia. L.L. was funded from the Munich School for Data Science (MUDS) and a fellowship from the Boehringer Ingelheim Fonds.

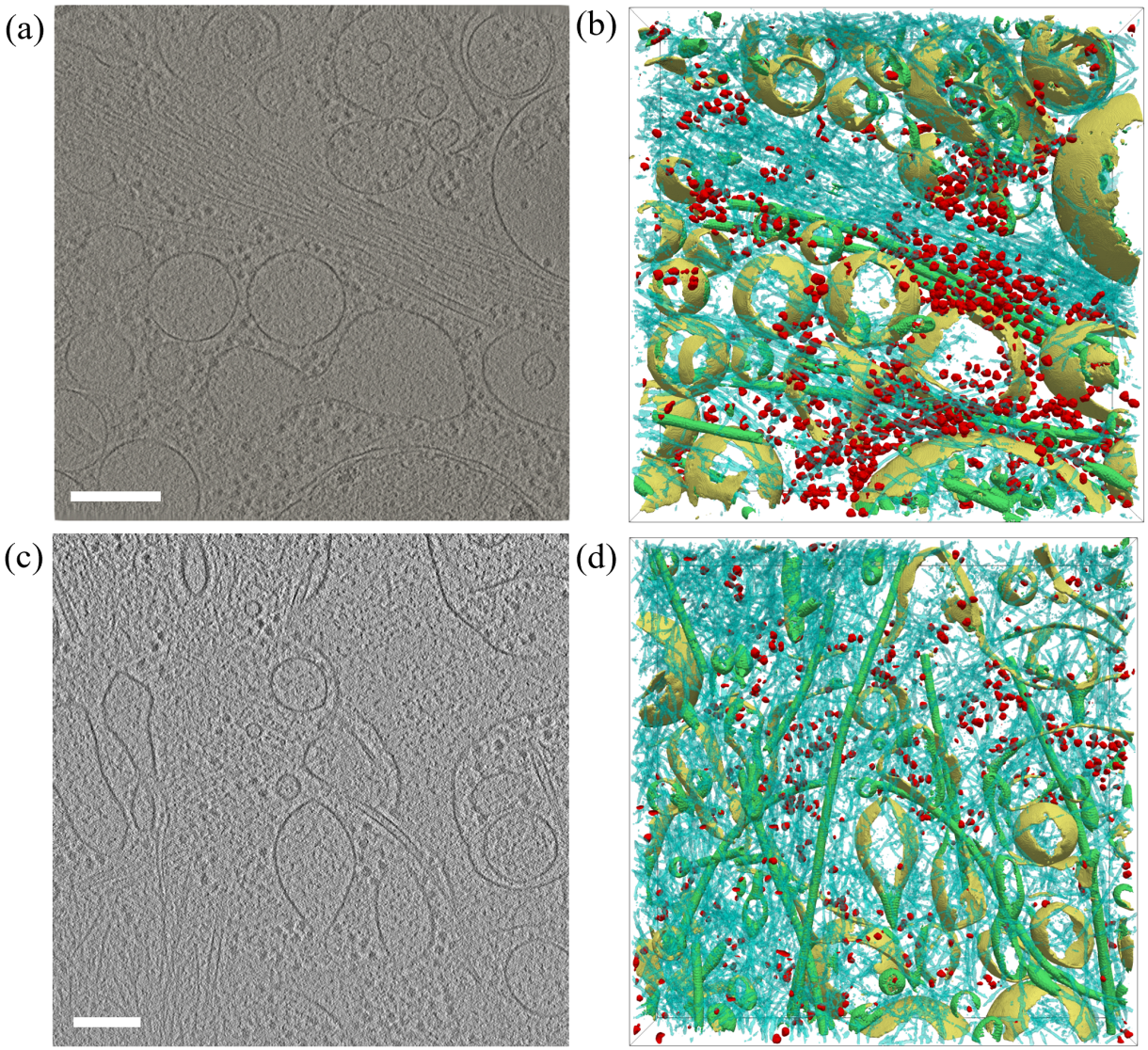

Figure 8: Training a deep learning model. (a) A 2D slice in the XY-plane of an *in situ* tomogram taken from EMD-10439. (b) Segmentation performed by nnU-Net trained solely from the synthetic data, membranes are in yellow, microtubules in green, actin filaments in transparent blue and ribosomes in red. (c-d) Similar to (a-b) but the tomogram was provided by [59] authors. Scale bars 200 nm.

Contributions

Conceptualization, algorithm development and software implementation was done by A.M-S., conceptualization was assisted by M.J. and H.P., L.L. implemented and trained the deep learning model. A.M-S. wrote the manuscript and all authors reviewed and edited the paper.

References

[1] Martin Turk and Wolfgang Baumeister. The promise and the challenges of cryo-electron tomography. *FEBS Letters*, 594(20):3243–3261, 2020.

[2] V. Lučić, A. Rigort, and W. Baumeister. Cryo-electron tomography: The challenge of doing structural biology in situ. *The Journal of cell biology*, 202(3):407–419, 2013.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[4] Yun-Tao Liu, Heng Zhang, Hui Wang, Chang-Lu Tao, Guo-Qiang Bi, and Z. Hong Zhou. Isotropic reconstruction for electron tomography with deep learning. *Nature Communications*, 13(1):6482–, 2022.

[5] Tim-Oliver Buchholz, Mareike Jordan, Gaia Pigino, and Florian Jug. Cryo-care: Content-aware image restoration for cryo-transmission electron microscopy data. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 502–506, 2019.

[6] Tristan Bepler, Kotaro Kelley, Alex J. Noble, and Bonnie Berger. Topaz-denoise: general deep denoising models for cryoem and cryoet. *Nature Communications*, 11(1):5208–, 2020.

[7] Muyuan Chen, Wei Dai, Stella Y Sun, Darius Jonasch, Cynthia Y He, Michael F Schmid, Wah Chiu, and Steven J Ludtke. Convolutional neural networks for automated annotation of cellular cryo-electron tomograms. *Nature Methods*, 14(10):983–985, 2017.

[8] Emmanuel Moebel, Antonio Martinez-Sanchez, Lorenz Lamm, Ricardo D. Righetto, Wojciech Wietrzynski, Sahradha Albert, Damien Lariviere, Eric Fourmentin, Stefan Pfeffer, Julio Ortiz, Wolfgang Baumeister, Tingying Peng, Benjamin D. Engel, and Charles Kervrann. Deep learning improves macromolecule identification in 3d cellular cryo-electron tomograms. *Nature Methods*, 18(11):1386–1394, 2021.

[9] Thomas Hrabe, Yuxiang Chen, Stefan Pfeffer, Luis Kuhn Cuellar, Ann-Victoria Mangold, and Friedrich Forster. Pytom: A python-based toolbox for localization of macromolecules in cryo-electron tomograms and subtomogram analysis. *Journal of Structural Biology*, 178(2):177–188, 2012. Special Issue: Electron Tomography.

[10] Shoh Asano, Yoshiyuki Fukuda, Florian Beck, Antje Aufderheide, Friedrich Förster, Radostin Danev, and Wolfgang Baumeister. A molecular census of 26s proteasomes in intact neurons. *Science*, 347(6220):439–442, 2015.

[11] Qiang Guo, Carina Lehmer, Antonio Martínez-Sánchez, Till Rudack, Florian Beck, Hannelore Hartmann, Manuela Pérez-Berlanga, Frédéric Frottin, Mark S Hipp, F Ulrich Hartl, et al. In situ structure of neuronal c9orf72 poly-ga aggregates reveals proteasome recruitment. *Cell*, 172(4):696–705, 2018.

[12] Wei Dai, Muyuan Chen, Christopher Myers, Steven J. Ludtke, B. Montgomery Pettitt, Jonathan A. King, Michael F. Schmid, and Wah Chiu. Visualizing individual rubisco and its assembly into carboxysomes in marine cyanobacteria by cryo-electron tomography. *Journal of Molecular Biology*, 430(21):4156–4167, 2018.

[13] Steffen Klein, Mirko Cortese, Sophie L. Winter, Moritz Wachsmuth-Melm, Christopher J. Neufeldt, Berati Cerikan, Megan L. Stanifer, Steeve Boulant, Ralf Bartenschlager, and Petr Chlanda. Sars-cov-2 structure and replication characterized by in situ cryo-electron tomography. *Nature Communications*, 11(1):5885–, 2020.

[14] Yoshiyuki Fukuda, Florian Beck, Jürgen M. Plitzko, and Wolfgang Baumeister. In situ structural studies of tripeptidyl peptidase ii (tppii) reveal spatial association with proteasomes. *Proceedings of the National Academy of Sciences*, 114(17):4412–4417, 2017.

[15] Antonio Martinez-Sanchez, Wolfgang Baumeister, and Vladan Lučić. Statistical spatial analysis for cryo-electron tomography. *Computer Methods and Programs in Biomedicine*, 218:106693, 2022.

[16] M. Jasnin, F. Beck, M. Ecke, Y. Fukuda, A. Martinez-Sanchez, W. Baumeister, and G. Gerisch. The architecture of traveling actin waves revealed by cryo-electron tomography. *Structure*, 27(8):1187–1189, 2019.

[17] Antonio Martinez-Sanchez, Ulrike Laugks, Zdravko Kochovski, Christos Papantoniou, Luca Zinzula, Wolfgang Baumeister, and Vladan Lučić. Trans-synaptic assemblies link synaptic vesicles and neuroreceptors. *Science Advances*, 7(10):eabe6204, 2021.

[18] E.S.F. Rosenzweig, B. Xu, L.K. Cuellar, A. Martinez-Sanchez, M. Schaffer, M. Strauss, H.N. Cartwright, P. Ronceray, J.M. Plitzko, F. Förster, et al. The eukaryotic co 2-concentrating organelle is liquid-like and exhibits dynamic reorganization. *Cell*, 171(1):148–162, 2017.

[19] Florian Brandt, Lars-Anders Carlson, F. Ulrich Hartl, Wolfgang Baumeister, and Kay Grünewald. The three-dimensional organization of polyribosomes in intact human cells. *Molecular Cell*, 39(4):560–569, 2010.

[20] Beata Turoňová, Mateusz Sikora, Christoph Schürmann, Wim J. H. Hagen, Sonja Welsch, Florian E. C. Blanc, Sören von Bülow, Michael Gecht, Katrin Bagola, Cindy Hörner, Ger van Zandbergen, Jonathan Landry, Nayara Trevisan Doimo de Azevedo, Shyamal Mosalaganti, Andre Schwarz, Roberto Covino, Michael D. Mühlebach, Gerhard Hummer, Jacomine Krijnse Locker, and Martin Beck. In situ structural analysis of sars-cov-2 spike reveals flexibility mediated by three hinges. *Science*, 370(6513):203–208, 2020.

[21] Victoria A. Trinkaus, Irene Riera-Tur, Antonio Martinez-Sanchez, Felix J. B. Bauerlein, Qiang Guo, Thomas Arzberger, Wolfgang Baumeister, Irina Dudanova, Mark S. Hipp, F. Ulrich Hartl, and Ruben Fernandez-Busnadiego. In situ architecture of neuronal α-synuclein inclusions. *Nature Communications*, 12(1):2110–20, 2021.

[22] Maria Salfer, Javier F. Collado, Wolfgang Baumeister, Rubén Fernández-Busnadiego, and Antonio Martínez-Sánchez. Reliable estimation of membrane curvature for cryo-electron tomography. *PLOS Computational Biology*, 16(8):1–29, 08 2020.

[23] Benjamin A. Barad, Michaela Medina, Daniel Fuentes, R. Luke Wiseman, and Danielle A Grotjahn. A surface morphometrics toolkit to quantify organellar membrane ultrastructure using cryo-electron tomography. *bioRxiv*, 2022.

[24] Friedrich Förster, Sabine Pruggnaller, Anja Seybert, and Achilleas S Frangakis. Classification of cryo-electron sub-tomograms using constrained correlation. *Journal of structural biology*, 161(3):276–286, 2008.

[25] Long Pei, Min Xu, Zachary Frazier, and Frank Albert. Simulating cryo electron tomograms of crowded cell cytoplasm for assessment of automated particle picking. *BMC Bioinformatics*, 17(1):405, 2016.

[26] Borja Rodríguez de Francisco, Armel Bezault, Xiao-Ping Xu, Dorit Hanein, and Niels Volkmann. Mepsi: A tool for simulating tomograms of membrane-embedded proteins. *Journal of Structural Biology*, 214(4):107921, 2022.

[27] Milov̌ Vulović, Raimond B.G. Ravelli, Lucas J. van Vliet, Abraham J. Koster, Ivan Lazić, Uwe Lücken, Hans Rullgård, Ozan Öktem, and Bernd Rieger. Image formation modeling in cryo-electron microscopy. *Journal of Structural Biology*, 183(1):19–32, 2013.

[28] H. Rullgard, L.-G. Ofverstedt, S. Masich, B. Daneholt, and O. Oktem. Simulation of transmission electron microscope images of biological specimens. *Journal of Microscopy*, 243(3):234–256, 2011.

[29] Benjamin Himes and Nikolaus Grigorieff. Cryo-TEM simulations of amorphous radiation-sensitive samples using multislice wave propagation. *IUCrJ*, 8(6):943–953, Nov 2021.

[30] Ilja Gubins, Marten L. Chaillet, Gijs van der Schot, M. Cristina Trueba, Remco C. Veltkamp, Friedrich Förster, Xiao Wang, Daisuke Kihara, Emmanuel Moebel, Nguyen P. Nguyen, Tommi White, Filiz Bunyak, Giorgos Papoulias, Stavros Gerolymatos, Evangelia I. Zacharaki, Konstantinos Moustakas, Xiangrui Zeng, Sinuo Liu, Min Xu, Yaoyu Wang, Cheng Chen, Xuefeng Cui, and Fa Zhang. SHREC 2021: Classification in Cryo-electron Tomograms. In Silvia Biasotti, Roberto M. Dyke, Yukun Lai, Paul L. Rosin, and Remco C. Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2021.

[31] Gavin Rice, Thorsten Wagner, Markus Stabrin, Oleg Sitsel, Daniel Prumbaum, and Steffan Raunser. Tomotwin: generalized 3d localization of macromolecules in cryo-electron tomograms with structural data mining. *Nature Methods*, 2023.

[32] Li Zhou, Chao Yang, Weiguo Gao, Talita Perciano, Karen M. Davies, and Nicholas K. Sauter. A machine learning pipeline for membrane segmentation of cryo-electron tomograms. *Journal of Computational Science*, 66:101904, 2023.

[33] Carson Purnell, Jessica Heebner, Michael T. Swulius, Ryan Hylton, Seth Kabonick, Michael Grillo, Sergei Grigoryev, Fred Heberle, M. Neal Waxham, and Matthew T. Swulius. Rapid synthesis of cryo-et data for training deep learning models. *bioRxiv*, 2023.

[34] Xindi Wu, Chengkun Li, Xiangrui Zeng, Haocheng Wei, Hong-Wen Deng, Jing Zhang, and Min Xu. Cryo-oetgan: Cryo-electron tomography image synthesis via unpaired image translation. *Frontiers in Physiology*, 13, 2022.

[35] Pavol Harar, Lukas Herrmann, Philipp Grohs, and David Haselbach. Faket: Simulating cryo-electron tomograms with neural style transfer, 2023.

[36] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.

[37] E. Moebel and C. Kervrann. Towards unsupervised classification of macromolecular complexes in cryo electron tomography: Challenges and opportunities. *Computer Methods and Programs in Biomedicine*, 225:107017, 2022.

[38] Hajime Nagashima and Sho Asakura. Dark-field light microscopic study of the flexibility of f-actin complexes. *Journal of Molecular Biology*, 136(2):169–182, 1980.

[39] Hanqing Jiang, Liying Jiang, Jonathan D Posner, and Bryan D Vogt. Atomistic-based continuum constitutive relation for microtubules: elastic modulus prediction. *Computational Mechanics*, 42:607–618, 2008.

[40] Antoine Jegou and Guillaume Romet-Lemonne. The many implications of actin filament helicity. *Seminars in Cell & Developmental Biology*, 102:65–72, 2020.

[41] Yaroslav Pollak, Sarah Goldberg, and Roee Amit. Self-avoiding wormlike chain model for double-stranded-dna loop formation. *Phys. Rev. E*, 90:052602, Nov 2014.

[42] T. Wiegand and K.A. Moloney. Rings, circles, and null-models for point pattern analysis in ecology. *Oikos*, 104(2):209–229, 2004.

[43] P. Andrey, K. Kieu, C. Kress, G. Lehmann, L. Tirichine, Z. Liu, E. Biot, P. G. Adenot, C. Hue-Beauvais, N. Houba-Herin, V. Duranthon, E. Devinoy, N. Beaujean, V. Gaudin, Y. Maurin, and P. Debey. Statistical analysis of 3D images detects regular spatial distributions of centromeres and chromocenters in animal and plant nuclei. *PLOS Computational Biology*, 6(7):e1000853, 2010.

[44] J.J. Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 4, pages 3993–3998 Vol.4, 2004.

[45] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[46] Catherine L. Lawson, Ardan Patwardhan, Matthew L. Baker, Corey Hryc, Eduardo Sanz Garcia, Brian P. Hudson, Ingvar Lagerstedt, Steven J. Ludtke, Grigore Pintilie, Raul Sala, John D. Westbrook, Helen M. Berman, Gerard J. Kleywegt, and Wah Chiu. EMDataBank unified data resource for 3DEM. *Nucleic Acids Research*, 44(D1):D396–D403, 11 2015.

[47] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[48] 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, 2004.

[49] Guang Tang, Liwei Peng, Philip R. Baldwin, Deepinder S. Mann, Wen Jiang, Ian Rees, and Steven J. Ludtke. Eman2: An extensible image processing suite for electron microscopy. *Journal of Structural Biology*, 157(1):38–46, 2007. Software tools for macromolecular microscopy.

[50] Kremer James R., David N. Mastronarde, and J. Richard McIntosh. Computer visualization of three-dimensional image data using imod. *Journal of structural biology*, 116(1):71–76, 1996.

[51] 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.

[52] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[53] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[54] Tom Burnley, Colin M. Palmer, and Martyn Winn. Recent developments in the *CCP-EM* software suite. *Acta Crystallographica Section D*, 73(6):469–477, Jun 2017.

[55] Will Schroeder, Ken Martin, and Bill Loresen. *The Visualization Toolkit (4th ed.)*. Kitware, 2006.

[56] James Ahrens, Berk Geveci, and Charles Law. ParaView: An end-user tool for large data visualization. In *Visualization Handbook*. Elesvier, 2005. ISBN 978-0123875822.

[57] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[58] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[59] Jonathan Schneider and Marion Jasnin. Capturing actin assemblies in cells using in situ cryo-electron tomography. *European Journal of Cell Biology*, 101(3):151224, 2022.

[60] A. Martinez-Sanchez, Z. Kochovski, U. Laugks, J. Meyer zum Alten Borgloh, S. Pfeffer, W. Baumeister, and V. Lučić. Template-free detection and classification of membrane-bound complexes in cryo-electron tomograms. *Nature Methods*, 17(2):209–216, 2020.