Contents lists available at ScienceDirect

Journal of Structural Biology

journal homepage: www.elsevier.com/locate/jsb

Research Article

Membrane and vesicle structure detection in cryo-electron tomography based on deep learning

Alain Morales-Martínez [a], Edgar Garduño [d,*], José María Carazo [c], Carlos Oscar S. Sorzano [b,c], José Luis Vilas [c]

[a] Posgrado en Ingeniería Eléctrica, Universidad Nacional Autónoma de México, Cd. Universitaria, C.P. 04510, Mexico City, Mexico
[b] Univ. San Pablo CEU, Campus Urb. Montepríncipe s/n, 28668, Boadilla del Monte, Madrid, Spain
[c] National Center for Biotechnology, CSIC, Campus Univ. Autónoma de Madrid, 28049, Cantoblanco, Madrid, Spain
[d] Department of Computer Science, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City, 04510, Mexico

ARTICLE INFO

Edited by Bauke W. Dijkstra

Dataset link: https://github.com/AlainMm/DLCryoTomo

Keywords:
Cryo-electron tomography
Convolutional neural networks
Semantic segmentation
Synthetic data generation

ABSTRACT

Cryo-electron tomography (cryo-ET) is a microscopy technique that enables the acquisition of 3D images of biological samples. Research in cell biology has shown that cellular processes are carried out by groups of macromolecules that interact in a crowded environment. In such an environment, where multiple biological macromolecules coexist and intertwine, semantic segmentation becomes even more challenging but crucial to understanding the structure and function of macromolecular complexes. However, manual semantic segmentation can be time-consuming, highly subjective, and prone to variability, which poses significant obstacles in studies dealing with large volumes of data. In contrast, automated algorithms such as Convolutional Neural Networks (CNNs) can process large-scale datasets with minimal human resources, thereby reducing the subjectivity associated with manual segmentation. In this work, we propose a convolutional neural network architecture that combines the features of U-Net, DeepLab, SegNet, Gated-SCNN, LSTM (Long Short-Term Memory), RNN (Recurrent Neural Network), and GAN (Generative Adversarial Network) architectures. This hybrid architecture effectively learns to identify different types of membranes and can replicate the behavior of a skilled human annotator. This system demonstrates a strong ability to segment various cellular membranes and vesicle structures.

1. Introduction

Cryo-electron tomography (cryo-ET) is an imaging modality where a biological sample is rapidly frozen and imaged multiple times by a transmission electron microscope (TEM); in this scheme, the images are acquired at different tilt angles of the sample. The collection of images, known as a tilt series, represents the projection data used to create a 3D representation of the sample, that is, a tomogram (Young and Villa, 2023).

A preliminary step before detailed structural analysis can be segmentation, which consists in assigning labels to the elements of a tomogram to associate them to biological structures such as ribosomes, macromolecular complexes, cell membranes, filaments, or microtubules. Because a tomogram contains significant information and noise, an appropriate segmentation simplifies several post-processing tasks such as the interpretation, analysis, quantification, or visualization. It is important to notice that segmentation is distinct from particle picking or particle selection, which refers to the identification of discrete positions (and possibly orientations) of individual particles within the tomogram for downstream subtomogram averaging or classification.

Manual segmentation has long been considered a *gold standard* for delineating structures. However, this process is unreproducible, imprecise, and labor-intensive. More importantly, operators must have a high degree of experience to obtain good results; unfortunately, observer bias and fatigue can negatively influence the results of manual segmentation (Hecksel et al., 2016).

In contrast, current automatic neural network-based segmentation methods can provide quick and consistent data analysis. These methods have shown excellent agreement with manual segmentation, significantly reducing the analysis time and offering a more objective approach with less variability (Zhou et al., 2020). Recently, methods based on the so-called Convolutional Neural Networks (CNN) (Chen

[*] Corresponding author.
E-mail addresses: alain_mm1@comunidad.unam.mx (A. Morales-Martínez), edgargar@ieee.org (E. Garduño), carazo@cnb.csic.es (J.M. Carazo), coss@cnb.csic.es (C.O.S. Sorzano), jlvilas@cnb.csic.es (J.L. Vilas).

https://doi.org/10.1016/j.jsb.2025.108258
Received 23 June 2025; Received in revised form 21 October 2025; Accepted 21 October 2025
Available online 30 October 2025

et al., 2018) have achieved great success in segmenting images from various imaging modalities (Minaee et al., 2022). These methods are highly popular because they can detect intricate non-linear relationships. They are also able to learn hierarchical feature representations. These properties make them ideal for complex tasks, such as identifying and delineating multiple structures within an image (Shelhamer et al., 2014). Therefore, these methods are particularly effective for segmenting subcellular structures such as organelles and cellular compartments, which have diverse textures, shapes, sizes, and densities in cryo-ET tomograms (Zhou et al., 2023).

There have been some efforts in cryo-ET to incorporate deep learning into the segmentation of tomograms and facilitate the analysis of biological structures with higher resolution. The authors of Dai et al. (2017) proposed using CNNs to automatically annotate cryo-ET data, thereby overcoming noise, missing wedge artifacts, and the complexity of cellular environments. Alternatively, Liu et al. (2018) utilized a Subtomogram Segmentation Network (SSN3D), a network that combines a Fully Convolutional Network (FCN) with an encoder–decoder structure, for the supervised segmentation of macromolecules in tomograms. Subsequently, the Fully Residual U-Net (FRU-Net), a modified U-Net with added residual layers, was used in de Mariscal et al. (2019) to segment small extracellular vesicles (sEVs) in images.

Recently, the DeepFinder network was proposed to enhance the localization and identification of macromolecules in tomograms (Moebel et al., 2020). Also, Zhou et al. (2021) proposed the Cryo-ET One-Shot Network (COS-Net) to classify and segment macromolecules in tomograms. Additionally, Dragonfly (Heebner et al., 2022) was proposed as a comprehensive tool for segmenting, visualizing, and analyzing tomograms. It provides the capability to train various versions of U-Net neural networks to identify specific structures, such as membranes, ribosomes, and microtubules, within a tomogram.

More recently, the authors of Zeng et al. (2023) proposed the Deep Iterative Subtomogram Clustering Approach (DISCA) to automatically identify structurally similar groups of macromolecular complexes in tomograms. Another related work, presented in de Teresa et al. (2022), proposes the network DeePiCt, an architecture designed to detect low-abundance and low-density complexes, with a specific focus on ribosomes.

On the other hand, MemBrain Lamm et al. (2022) is a deep-learning-based pipeline designed for automatically detecting membrane proteins in tomograms by sampling 3D subimages from segmented membranes, scoring them with a CNN, and then extracting the positions of proteins through clustering. In addition, an alternative approach is ColabSeg, a Python-based tool designed to edit, process, and visualize membrane segmentations from noisy tomograms (Siggel et al., 2024); specifically addressing the challenges associated with segmenting lipid membranes.

Ais (Last et al., 2024) is described as a machine learning-based software that is designed for multi-feature segmentation and particle selection for subtomogram averaging in cryo-ET data. The software includes a library of adaptations of various single-model architectures, including InceptionNet, ResNet, several UNet variations, VGGNet, and the generative adversarial network Pix2Pix. Finally, TARDIS (Transformer-based Rapid Dimensionless Instance Segmentation) (Kiewisz et al., 2024), a machine learning framework for annotating micrographs and tomograms, integrates deep learning for semantic segmentation with a geometric model for instance segmentation of various macromolecules.

Some proposed methodologies can work directly with 3D data without needing extensive filtering or retraining. However, this direct segmentation approach demands significant computing power. This requirement becomes particularly expensive when handling large volumes of data and can be inefficient without powerful GPUs or high-performance clusters.

Moreover, previous studies have utilized versions of U-Net architectures, which restrict their ability to adapt to different scales and noise levels. Traditional U-Nets are designed primarily to capture local contexts and are highly dependent on the quality and quantity of manually annotated training data. As a result, they may struggle to accurately segment complex, diffuse, or poorly defined structures, such as vesicles in cryo-ET.

In this work, we demonstrate that it is possible to extend semantic segmentation methods (approaches associating a label or category with every pixel or voxel) to automate the detection of membranes and vesicles in tomograms. This is achieved through a hybrid neural network architecture that combines the features of several architectures, such as U-Net, DeepLab, SegNet, Gated-SCNN, Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Generative Adversarial Network (GAN). The integration of U-Net enables our model to combine low-level information with high-level information, resulting in precise segmentation of border-like structures even in the presence of noise. DeepLab permits our model to capture wider contexts while maintaining spatial resolution (this enhances the segmentation of vesicles and membranes of varying sizes and shapes, accommodating multiple scales). SegNet enhances the accuracy of a segmentation without increasing the number of parameters, which in turn reduces computational power requirements; this helps preserve the precise location of thin structures (e.g., membranes). The Gated-SCNN enhances the refinement of contours for membranes and vesicles, crucial when boundaries appear blurred; this is achieved by its dedicated branch for boundary detection working in conjunction with its main branch through gated modules. An important model is the LSTM because it can keep track of arbitrary long-term dependencies in consecutive 2D images and it allows our model to learn the contextual, spatial, and temporal dependencies between adjacent 2D images. As a result, it helps preserve the structural continuity of vesicles and membranes throughout the 2D images making up the tomogram. The GAN model promotes the creation of additional synthetic data that allows more variability in the training process. This addition is particularly beneficial when there is a limited amount of high-quality annotated data available for training. Finally, we also incorporate spatial and channel attention mechanisms, dilated convolutions, edge attention, and specific loss functions, such as Dice and Focal loss, to enhance fine segmentation details.

This work is organized as follows: the next section provides a detailed description of our proposed methodology. Subsequently, in Section 3, we explain the development of the experiments carried out. Then, in Section 4, we present results to evaluate our approach as a tool for validation. Finally, in Section 5, we discuss and analyze the results reported in the previous section.

2. Methodology

In this work, we are interested in what we refer to as Semantic Convolutional Neural Networks (S-CNN) (Teuwen and Moriakov, 2020). Instead of relying on a single architecture to segment tomograms, we propose using a combination of recurrent and convolutional neural networks, which we refer to as a *hybrid architecture*. See Fig. 1 for a schematic of the fundamental architecture. For an easy flow of the presentation, we briefly describe this architecture next, but will provide the details in Appendix A.

2.1. Hybrid network architecture

Similar to the U-Net network, which effectively captures fine details through its encoding–decoding structure with skip connections (Wang et al., 2023), our hybrid architecture also features two essential components: a compression path, referred to as the encoder, and an expansion path, known as the decoder, as shown in Fig. 1. The incorporation of skip connections facilitates the incorporation of features extracted in the earlier layers of the network. This process is crucial for preserving fine spatial details.

The encoder blocks are fully described in Appendix A. At this point, it is crucial to note two of its features:

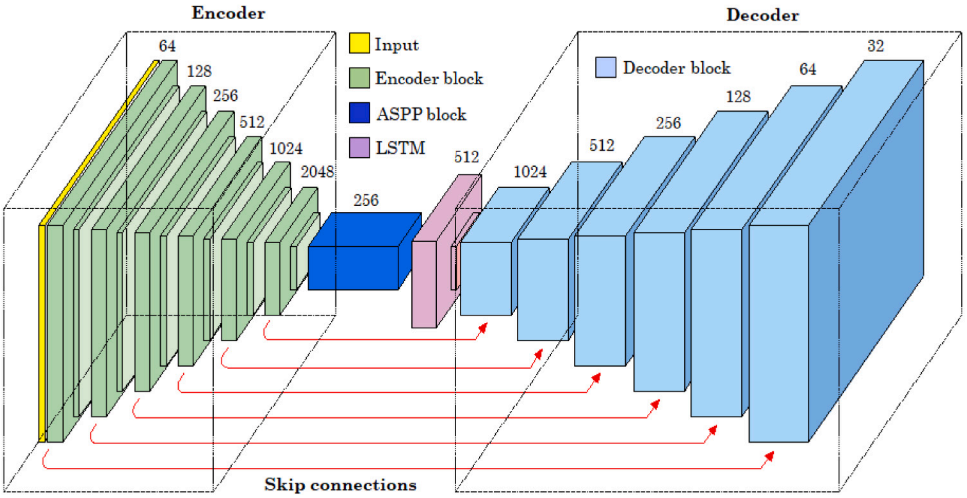


Fig. 1. Hybrid neural network model combining an encoder–decoder architecture. This design enables multi-scale segmentation, capturing fine details. Additionally, the model's capacity to sequentially process extracted features allows it to identify temporal patterns and relationships.

- We utilize the Gated-SCNN architecture (Takikawa et al., 2019) and its gating mechanism to maximize the benefits of its attention procedure, which dynamically determines relevant information and prioritizes its flow through the various layers of the network. By incorporating this architecture, the hybrid architecture can efficiently process information, enabling it to focus on edges and subtle details, such as intensity or shape, of objects in tomograms. Following this approach, we incorporate an Attention block, in conjunction with a Convolutional block, which includes both spatial and channel attention mechanisms; these mechanisms enhance the robustness of the hybrid architecture against variations in image quality and noise, which are common challenges in tomograms.
- The SegNet architecture (Badrinarayanan et al., 2016) is renowned for its success in segmenting images; for this reason, we incorporate some of its components into the Encoder block, as they enable the precise re-creation of segmented structures (a feature particularly beneficial for edge highlighting in tomograms). In this manner, the compression path comprises six encoder blocks with distinct filters, as illustrated in Fig. 1. To prevent overfitting, each of these blocks is followed by a Dropout layer.

Importantly, we incorporate elements of the DeepLab architecture, specifically its Atrous Spatial Pyramid Pooling (ASPP) block (Chen et al., 2018), into our hybrid architecture. This block utilizes multiple convolutions with different dilation rates. These are used to expand the receptive field and capture features at various scales.

All the aforementioned components of our hybrid architecture operate on 2D images. For this reason, we consider a tomogram (a 3D image) as a stack of 2D images, with the values of neighboring pixels in contiguous images being related. Hence, it is crucial to maintain such consistencies when segmenting a 3D image into a stack of 2D images. Therefore, we incorporate parts of the Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) architectures because they are commonly used for sequential tasks (Shi et al., 2015). This feature ensures consistency in the semantic segmentation of biological structures that can vary in shape, size, and orientation.

For our hybrid architecture, we implemented an expansion path consisting of six Decoder blocks that reverse the encoding process to assemble a segmented image from the compressed features. At each stage, a Convolution block is applied to enhance the image resolution. Finally, our hybrid architecture includes a Conv2D layer to generate the final output.

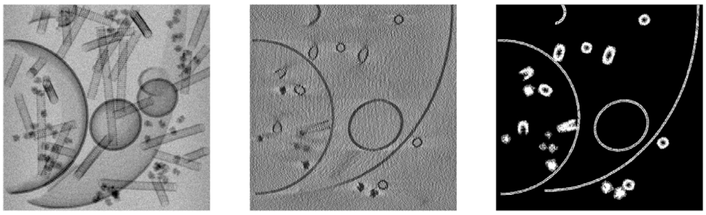
2.2. Datasets

We trained our hybrid neural network model using a combination of real and synthetic data. The datasets from real acquisitions were obtained from the Electron Microscopy Data Bank (EMDB) (wwPDB Consortium, T., 2023) and the Electron Microscopy Public Image Archive (EMPIAR) (Iudin et al., 2022). These datasets are:

EMD-8594	Automated tomogram annotation of PC12 cell.
EMPIAR-10236	Cryo-ET of cis-mutated mouse protocadherin gamma B6 on membranes.
EMPIAR-10368	Human delta protocadherin 1 full ectodomains on membranes.
EMD-10439	Tomogram used for *in situ* template-free membrane bound complexes determination
EMPIAR-10498	Arrangements of proteins at reconstituted synaptic vesicle fusion sites depend on the separation of membranes.
EMD-15394	Cell–cell contact between two PTK-1 cells.
EMPIAR-11751	Micrographs of vesicles derived from the endoplasmic reticulum in HEK293F cells. Scipion software (Conesa et al., 2023) is utilized to perform the tomogram reconstruction process.
EMPIAR-12038	Cryo-ET dataset of purified SARS-CoV-2 double membrane vesicles formed by nsp3-4.
EMD-16084	Munc13-SNAP25 cryo-ET dataset, synapse tomo Munc13 DHet 115.
EMD-24604	Tomogram of SARS-CoV-2 spike-bearing virus-like particles (VLPs) interacting with hACE2-bearing extracellular vesicles (tEVs), showing various intermediate states of the SARS-CoV-2 spike protein.

This work does not provide a detailed analysis of specific biological use cases; however, it is important to note that EMPIAR and EMD data are derived from actual cryo-ET experiments. These experiments capture cellular structures in their natural context, incorporating instrumental noise, acquisition artifacts, and biological variability.

We incorporate limited data acquisition range, the missing wedge effect, low contrast, low signal-to-noise ratio, and aberrations caused by a TEM while generating synthetic data to address the inherent challenges in cryo-ET. In this way, we can effectively manage various aspects of the imaging process, including image aberrations, resolution, contrast, and noise sensitivity. Although the physical interactions of electrons with the sample are not directly simulated, they can be



(a) (b) (c)

Fig. 2. (a) Projection of the tomogram along the z-direction using synthetic data (*phantoms*) shows membranes exhibiting various combinations of local curvatures, clusters of macromolecules (either cytosolic or membrane-bound), and polymers such as microtubules and actin networks. (b) A 2D slice in the xy-plane of the tomogram. (c) Labeled (or segmented) fundamental truth map, each type of structure has a different label with the background's equal to 0.

approximated reliably enough to produce synthetic images that mimic the visual and structural results of a real microscope and cryo-ET.

The synthetic data are created using PolNet (Martinez-Sanchez et al., 2024). This software enables the simulation of tomograms comprising a diverse range of structures, including membranes, groups of macromolecules, and polymers, such as microtubules or actin networks. Their structures are simulated using geometrical shapes such as ellipsoids and tori. In particular, membranes are modeled as parametric surfaces with double Gaussian profiles parallel to each other, forming double membranes. This model aims to replicate a lipid bilayer, which appears as two closely spaced walls.

Double Gaussian profiles effectively simulate the density distribution observed in cryo-ET for membranes, providing a synthetic representation that honors its actual physical structure. Biological membranes are continuous and smooth, lacking abrupt edges. Gaussian functions share these properties: they are both continuous and infinitely differentiable. Utilizing double Gaussian profiles allows for precise control over the geometry of synthetic membranes, enabling the creation of shapes with varying curvatures without discontinuities. Thus, it is possible to model the spatial continuity of a membrane while avoiding introducing artifacts. Furthermore, the ability to adjust the curvature during synthetic data generation equips the hybrid model to be pretrained, enhancing its robustness against various real morphological configurations.

To ensure a good coverage of different cellular environments, we employ parameterized stochastic models to generate a wide variety of geometries and organizational combinations. The low-order features are created using parametric mathematical models, allowing for the control of their geometry and organization. Additionally, we employ parameterized stochastic models to generate a wide variety of geometries and organizational combinations, enabling the simulation of representative synthetic datasets. From each of these phantoms, we generated a series of several 2D projections by tilting them in the range $[-60°, 60°]$ using IMOD (Kremer et al., 1996). Noise with a Gaussian distribution and zero mean is added to every projection to meet an SNR in the range of $[1, 2]$ (Martinez-Sanchez et al., 2024). The misalignment of the tilt series is simulated by applying random offsets in the horizontal and vertical directions, employing a sinusoidal approach to penalize high tilt angles. The mean value of these offsets is 1 pixel, with a maximum misalignment of 1.5 pixels. The goal is to simulate imperfect alignment of the tilt series, which is essential for assessing the robustness of the hybrid model in near-real conditions. The simulation process results in 3D images with dimensions of 500 × 500 × 250 voxels, where each voxel has a size of 10 Å. It is worth noting that these images display distortions caused by angular sampling and the missing wedge effect, as expected for real tomograms in cryo-ET (see Fig. 2).

Generating and processing synthetic data at a resolution of 10 Å significantly reduces computational costs and pre-training time. Using larger pixel sizes helps to smooth out fine details, which serves as a form of regularization during pre-training. This approach prevents the hybrid model from overfitting to overly precise or unrealistic patterns that are often found in synthetic data. The hybrid architecture initially focuses on learning global shapes and coarse structures, which is adequate for pretraining the model for tasks such as contour detection, general shape recognition, and orientation analysis.

On the other hand, a Generative Adversarial Network (GAN) architecture is employed to generate additional synthetic data. To achieve this, we implement a Pix2Pix model that consists of a U-Net and a PatchGAN, as detailed in Appendix B. This process results in a new dataset comprised of pairs of synthetic images and known masks, enabling the training of the hybrid model to be supplemented without the need for extra manual annotations; in this way, the training of the hybrid model is enhanced by introducing greater variability.

2.3. Evaluation metrics

To evaluate the performance of our hybrid architecture, we first introduce the concept of pixels correctly classified as part of the region of interest (true positives), correctly classified as not in the region of interest (true negatives), misclassified as positives (false positives), and misclassified as negatives (false negatives). Based on these pixel classifications, we utilize the following metrics:

- Accuracy $= \frac{\text{True Positives}+\text{True Negatives}}{\text{Total Pixels}}$.
- Precision $= \frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}$.
- Recall $= \frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$.
- F1 score $= \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$.
- Dice coefficient $= \frac{2\times\text{True Positives}}{2\times\text{True Positives}+\text{False Positives}+\text{False Negatives}}$.
- Intersection over Union (IoU) $= \frac{\left|O_i \cap O_j\right|}{\left|O_i \cup O_j\right|}$, where O_i and O_j, for $i \neq j$, represent two different segmentations. This is also known as the Jaccard similarity coefficient in the fields of statistics or set theory.

The last three metrics assess the pixel overlap between the predicted mask generated by the hybrid architecture and a reference mask, using values in the range of $[0, 1]$.

2.4. Implementation details

We utilize the Google Colab platform (Google LLC, 2017), a free cloud-based platform for collaborative development, which integrates Jupyter Notebook (Kluyver et al., 2016) (an open-source, interactive development environment for Python). This platform enables access to libraries such as TensorFlow, which is essential for image processing and for building and training our hybrid architecture. Notably, the platform has access to GPU and TPU accelerators. The code generated during the current study can be accessed at https://github.com/AlainMm/DLCryoTomo.

On the website there is a file called “requirements.txt” whose content describes how to install the necessary environment locally. Additionally, the following link to Google Colab: https://colab.research.google.com/github/AlainMm/DLCryoTomo/blob/main/model/Hybrid-model.ipynbpermits accessing a Jupyter Notebook with the necessary environment pre-installed for cloud execution.

3. Development of experiments

Before segmenting the tomograms, the dataset was processed in ScipionTomo (Jiménez de la Morena et al., 2022; Conesa et al., 2023). In particular, the tomograms were imported, preprocessed, and denoised. The preprocessing was carried out with IMOD and involved a binning operation and gray scaling. Thus, IMOD produced a tomogram with dimensions corresponding to the datasets in Section 2.2 and a zero mean and standard deviation of 1. The denoising was performed using Tomo3D via Edge Enhancing Diffusion (EED) (Agulleiro and Fern andez, 2011).

Although there exists recent deep learning-based approaches to remove noise in cryo-ET, we have chosen Edge Enhancing Diffusion (EED) because of its ability to preserve relevant edges without introducing artifacts, its reproducibility, and its direct integration into the Tomo3D pipeline, which has been widely validated in other studies. Furthermore, this choice also enhances traceability and facilitates comparisons with other studies.

After, we apply *contrast-limited adaptive histogram equalization* (CLAHE), a variant of adaptive histogram equalization, as a final step in the preprocessing of a 2D image (Stimper et al., 2019). This is done using the z-slices of the tomogram. However, CLAHE can behave inconsistently between tomograms due to the high variability in intensity distribution caused by experimental factors, acquisition conditions, and sampling differences. Thus, it is necessary to apply a pre-intensity normalization (i.e., rescaling the values of each volume to a standard range), something that contributes to the statistical consistency of the dataset in subsequent stages, particularly during the training of the hybrid architecture, which benefits from inputs with stable ranges and similar statistical properties. When using CLAHE, the block size is equal to 32 pixels, a size that improves local contrast while minimizing noise amplification. The local contrast threshold (whose value regulates the degree of contrast enhancement, where a lower value results in a more subtle effect with reduced noise, while a higher value boosts local contrast but may also amplify noise) is set at 3.0, providing a practical compromise between enhancing contrast and controlling noise.

For any Deep Learning model, it is necessary to set its parameters and hyperparameters before carrying out experiments. Hyperparameters are essential because they provide the tools required to measure performance, introduce nonlinearities, and optimize the model’s parameters. In this work, we utilize binary cross-entropy (BCE) as the loss function for binary segmentation tasks and categorical cross-entropy (CCE) for multiclass segmentation tasks (Qin et al., 2021). For activation functions, we apply the Sigmoid function in binary segmentation scenarios and the Softmax function for multiclass problems (Li et al., 2019). For optimization, we have selected the Adaptive Moment Estimation (Adam) algorithm, which is a variant of gradient descent (Fatima, 2020).

Furthermore, we introduce two callbacks to enhance the performance of the proposed hybrid model during training, aiming to prevent overfitting and poor convergence. First, we use an early stopping criterion to terminate training when the validation loss stops improving (i.e., it starts to minimize). This approach ensures that parameters are restored to their optimal values, as observed during the training process. We also use a scheme to reduce the learning rate (through the Reduce-LR callback) when the performance on the validation dataset stops improving. This reduction can help achieve better convergence during the later stages of training.

3.1. Network training

Before training the hybrid architecture, images are standardized to a fixed pixel size of 5 Å. Normalizing by sampling rate ensures that all images have a consistent physical scale. This process does not alter the actual physical size of the object; instead, it changes how the object is represented in pixels. After normalizing to the fixed pixel size, we make further adjustments to the image size through padding or cropping to maintain the appropriate scale.

Notably, cryo-EM techniques can generate tomograms with dimensions larger than $2000 \times 2000 \times 500$ voxels, which might occupy more than 16 GB of storage (Iudin et al., 2022). Using such large images for training or inference poses challenges for modern computers. Taking these facts into consideration, we opted to process a tomogram as several 2D images, each composed of several patches of dimensions 224 × 224 pixels. This size strikes a good balance between image resolution, with dimensions large enough to contain significant regions of interest, and computational performance, allowing the processing of several large tiles simultaneously (Mishkin et al., 2017). We also consider that current methods offer great flexibility in augmenting and preprocessing 2D images, which enhances the robustness of the neural network model (Ridnik et al., 2022).

Moreover, using pre-trained models on large 2D datasets and fine-tuning them for specific cryo-ET data can facilitate a quick adaptation to 3D structure segmentation (andler et al., 2019). The morphological features that need to be segmented are usually spread across multiple 2D images or sections of a tomogram. Training models with these 2D images can be beneficial for learning the relevant features.

In our hybrid architecture, the use of an LSTM network enhances the ability to capture contextual dependencies between consecutive slices, which improves 2D image segmentation by incorporating information from both previous and subsequent slices in the tomogram. This approach reduces inconsistent segmentation errors between different slices, helping to maintain structural continuity. For instance, a membrane that curves smoothly throughout the volume should appear consistent across slices. In this manner, LSTM functions as a memory mechanism, ensuring coherence and structure in the segmentation.

By incorporating the LSTM architecture, the hybrid model essentially operates as a 2.5D framework. This design enables the simultaneous processing of multiple 2D images, allowing it to keep track of contextual 3D information; in this way, we avoid working with the entire volume. An extra advantage of this approach is that allows for using pre-trained models from a large number of existing 2D neural networks. Furthermore, the 2.5D approach offers several advantages such as making it easier to implement with existing and optimized 2D-based models (just adding channels), consuming less memory than 3D-based models while providing greater spatial context than purely 2D networks, and allowing for the use of pre-trained models from 2D neural networks.

In cryo-ET, where noise levels are high and structures like membranes or vesicles can be continuous but very thin in the Z-axis, a 2.5D approach is beneficial. This method effectively captures the continuity of membranes across multiple planes, requires less training data and computational resources than a fully 3D network, and helps avoiding the overfitting challenges associated with complete 3D models.

On the other hand, we adjusted the parameters of our hybrid architecture by training it with experimental and synthetic tomograms. We initiated the training process for our hybrid model using the synthetically generated images described in Section 2.2. For the first stage of the training process, we used a batch size of 32 images with 50 epochs (each lasting approximately 2 min), and some of the predictions are shown in Fig. 5. This first stage allows our hybrid model to adjust parameters for valuable features such as edges, textures, and patterns. In this way, the process of transfer learning allows the hybrid model to generalize effectively to experimental data.

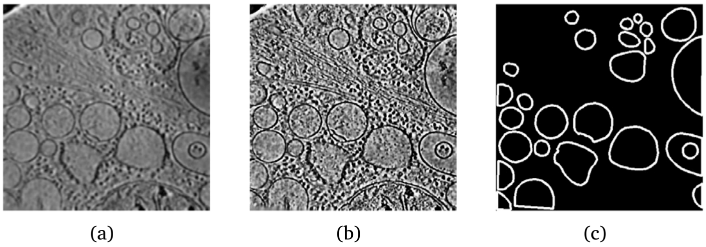


Fig. 3. Real membrane cell context. (a) A 2D slice in the xy-plane of an *in situ* tomogram taken from EMD-10439. The aim is to increase contrast and improve the signal-to-noise ratio (SNR). (b) Normalization process, denoising, and diffusion along the edges. (c) Manually designed binary segmentation mask.

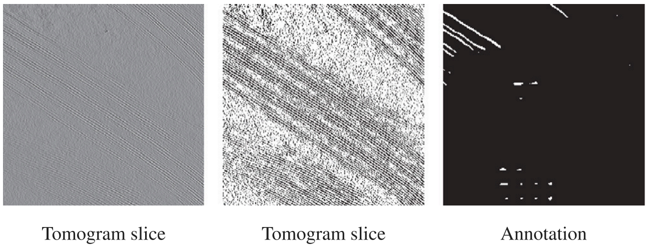


Fig. 4. CryoET Data Portal: In situ cryo-ET dataset of Chlamydomonas reinhardtii prepared using cryo-plasmaFIB milling. The axial section of the tomogram, along with its annotation, shows that the database used is not accurately processed. There are visible artifacts in the central region of the image, likely caused by movement or acquisition errors. Additionally, several structures are not identified in the annotation, which compromises the quality of the labeling.

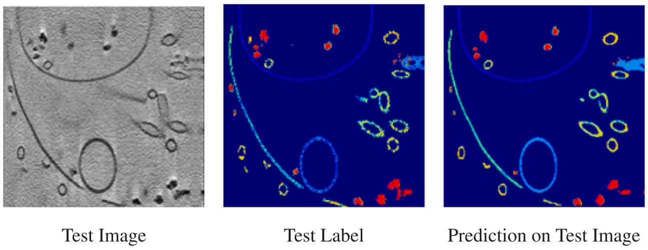


Fig. 5. Predictions of the hybrid neural network model with intersection-over-union (IoU) validation metric that evaluates model performance for the generated synthetic dataset. A 2D slice of the generated synthetic tomogram, the input segmentation mask, and the mask predicted by the convolutional neural network model are shown.

Additionally, to use experimental tomograms, it is necessary to know the ground truth, and we obtained such information by manually segmenting masks of a tomogram's regions of interest, see Fig. 3. We utilized two programs for manual segmentation: a version of the Segment Anything Model (SAM) (Kirillov et al., 2023), modified explicitly for cell membranes, and the open-source VGG Image Annotator (VIA) (Dutta and Zisserman, 2019). SAM enables automatic generation of object segmentation masks by simply specifying a point or a bounding box, eliminating the need for manual drawing. This feature makes it ideal for workflows that involve large volumes of data. In contrast, VIA is better suited for situations where manual precision is essential or when annotating objects that require multiple, precise geometric shapes.

The CryoET Data Portal (Ermel et al., 2024) offers tomograms along with annotations for various subcellular structures, including ribosomes, microtubules, and protein complexes. However, we chose not to use these annotations in this work due to several important limitations. First, the available segmentations are neither exhaustive nor consistent for structures like membranes or vesicles, which often have fuzzy edges or complex shapes. Additionally, the dataset does not adequately represent certain specific types of vesicles and membranes, which restricts its usefulness for segmentation tasks focused on these structures.

An important consideration is that annotations are usually presented as points of interest or instance masks, which do not meet the continuous semantic granularity level required for this study. Additionally, some semi-automated annotations exhibit limited accuracy and lack spatial consistency, particularly in the central regions of the tomograms where artifacts and unidentified structures are prevalent, as illustrated in Fig. 4. Many volumes in the CryoET Data Portal require further pre-processing to standardize formats, scales, or resolutions, which imposes a significant technical burden without any guarantee of improving annotation quality.

Therefore, our approach involves a two-step process: first, we utilize a pre-training strategy using synthetic data, and then we fine-tune the hybrid model with experimental data that features consistent and compatible annotations. To maintain consistency between the synthetic data and the experimental data, we manually generated the labels using the same semantic criteria employed in modeling the synthetic geometries. This consistency is crucial for the effective training of the proposed hybrid model.

Since our pipeline requires optimized annotations for direct integration, it is reasonable to use custom annotations generated by modern tools like the Segment Anything Model (SAM) for quick initial masks. These can then be refined further. Additionally, we use the VGG Image Annotator (VIA) to create high-quality dense annotations. Our integration tools, which are based on models such as U-Net, Gated-SCNN, and LSTM, necessitate denser and more specific spatial annotation structures. This is especially important for accurately capturing structures with fuzzy boundaries, such as highly deformed vesicles and membranes.

The inherent variability in human labeling, along with the potential for systematic errors, necessitates quality control. In this context, using quantitative metrics such as the Dice coefficient, intersection-over-union (IoU), accuracy, and sensitivity is crucial for validating the consistency of annotations and establishing a benchmark based on human performance. The performance of the hybrid model is compared to these human annotations. Additionally, these metrics help identify inconsistencies in the masks and assess whether the predictions made by the hybrid model are comparable to expert judgments.

It is also important to note that utilizing synthetic data allows for the creation of perfectly annotated masks for pre-training, which accelerates the learning process of the hybrid architecture. The final training phase with experimental data ensures that the model adapts to real conditions, including noise, resolution, and cryo-ET artifacts. This approach ensures that the masks used as ground truth accurately represent the biological structures of interest, providing a robust basis for an objective evaluation of the hybrid segmentation model.

Conventional 3D segmentation methods process volumetric data by treating all dimensions as if they have the same resolution, known as isotropic resolution. This approach means that three-dimensional convolutional kernels apply filters symmetrically across the X, Y, and Z axes. However, this can introduce structural noise along the Z axis, which typically has lower resolution and higher uncertainty. As a result, the model may overfit artifacts in the Z dimension, leading to decreased generalization. Additionally, the physical characteristics of the acquisition system are often overlooked, which negatively impacts the structural fidelity of the segmentation.

To address these limitations, we propose a hybrid approach based on a 2.5D model. In this model, the volume is processed as a sequence of 2D slices, typically in the XY plane. To capture continuity between these slices in the Z direction, we employ a bidirectional LSTM architecture. This architecture effectively models sequential dependencies in both axial directions (Z^+ and Z^-). This approach offers several advantages. First, it respects the anisotropic nature of the volume, avoiding the assumption of symmetry when it does not exist. Additionally, it captures structural coherence between adjacent slices without requiring full 3D convolutions. Moreover, this method reduces the risk of overfitting to the Z axis by preventing the model from learning spurious patterns that may arise from low SNR or artifacts in that dimension.

Current deep learning models often have thousands or even millions of parameters, requiring a very large training dataset to adjust them effectively. One way to overcome this limitation is to increase the number of annotated images by applying transformations to the existing images in the training dataset. These transformations create a database that preserves the underlying information while introducing variations.

Rotating images by multiples of 90°, along with flipping, does not require interpolation, which avoids introducing more artifacts. Such an operation would ensure preserving the digital information of 2D images and, hence, the morphology of imaged membranes and vesicles. However, these techniques may only produce a limited set of possible orientations. As a result, the hybrid model may overfit to these specific symmetries, leading to a hybrid model that is less robust in the face of small, real angular variations in the data. Consequently, its performance may decline when faced with slight, real rotations during inference.

We consider that simply restricting the images to 90° angles and their flips does not adequately capture the natural variability of orientations cellular structures can have during acquisition. Therefore, the introduction of slight rotations to the images benefits the hybrid model become invariant to subtle differences in orientation (as sort of simulated physiological variability). As a result, the model' s ability to generalize effectively is enhanced.

Following this approach, we first rotate the images with a random angle in the range $[0^\circ, 20^\circ]$ and then shift them vertically and horizontally, applying random offsets in the range $[0\%, 20\%]$ of an image's width and height, respectively. We also introduce a random zoom effect, adjusting an image's dimensions between 80% and 120% of its original size. A horizontal flip is also applied to an image, which helps to improve the model's robustness to variations in object orientation. In addition, any empty pixels created by transformations, such as rotations and shifts, are filled with the value of the nearest pixel.

After completing the data augmentation process, we have compiled a database of 4800 experimental images sourced from publicly available datasets in the EMDB and EMPIAR, as detailed in Section 2.2. We allocated 3360 (70%) of those for training, leaving 720 (15%) for validation and 720 (15%) for testing purposes. This database is sufficiently large to capture the fine details, complexity, and variability of membranes and vesicles in cryo-ET images. The diversity created through data augmentation enables the hybrid model to learn more general and robust representations, helping to prevent overfitting and enhancing its ability to generalize to new samples.

3.2. Comparison with other methods

We compare our hybrid model against the commercial software Dragonfly (Heebner et al., 2022), which is designed for 3D image segmentation and analysis using variations of U-Net architectures, the open-source Ais (Last et al., 2024), a machine-learning-based software designed for segmentation in cryo-ET, the MemBrain-seg, a core module of MemBrain v2 (Lamm et al., 2024) that utilizes a variant of the U-Net architecture for membrane segmentation in cryo-ET, and TARDIS (Kiewisz et al., 2024). This framework uses a pre-trained FNet model and features an encoder–decoder structure with two decoder branches. In TARDIS, the first decoder follows a traditional U-Net design, using skip connections to maintain spatial detail. In contrast, the second decoder incorporates hierarchical skip connections from all levels of the encoder. This approach generates probability maps that are then converted into binary masks and processed into point clouds. Because these programs are based on neural networks, they also have hyperparameters and parameters that need to be selected and set.

The Dragonfly software has default hyperparameters fixed by the system. The loss function used is the mean squared error, and the Adadelta optimizer is employed to update the parameters during training, with a fixed learning rate of 1. The parameters of Dragonfly are updated using the same training dataset of images used for our hybrid model. However, it is worth noting that the Dragonfly model was initially trained with 100 epochs, as this is also the default value provided by the software and is fixed. However, the software implements early stopping, which automatically stops training if no significant improvement in performance is observed.

In this way, if the Dragonfly model does not show any improvement in performance on the validation dataset, meaning there is no increase in accuracy or decrease in the loss function after 10 consecutive epochs, the learning rate will be reduced. Additionally, if the model's performance still has not improved after 15 consecutive epochs, training will be stopped early. This is done to prevent the Dragonfly model from overfitting by learning too much from the training data.

During each epoch, the training dataset for the Dragonfly model is divided into batches of size 512, a default value given by the software and fixed. Once all the batches have been processed, an epoch is completed, allowing the model in Dragonfly to see and learn from all the training images. The duration of an epoch across the entire training dataset is around 300 min. In Table 1, we present a summary of the coefficients for the validation metrics obtained during the training process.

Regarding Ais, it features a library that includes adaptations of various single-model architectures, such as the default architecture (a U-Net) available in EMAN2 (Dai et al., 2017), InceptionNet, ResNet, several U-Net variants, VGGNet, and the GAN network Pix2pix. In this software, the first step during segmentation involves making manual annotations, which are then extracted and utilized as ground truth labels.

The comparison presented in Table 2 was generated using a test dataset that is consistent across all Ais software architectures. This dataset comprises 240 images, each with dimensions 224 × 224 pixels, depicting membranous and vesicular structures, along with their corresponding annotations. Following the methodology proposed by the Ais software, the dataset was augmented by resampling the images in random orientations, resulting in each image being duplicated 10 times, which brings the total to 2,400 images. Finally, the Ais' neural networks were trained for 50 epochs with a batch size of 32 images. The selection of different dataset sizes is based on a strategy to assess the maximum performance that each type of architecture can achieve. Models such as InceptionNet, ResNet, and VGGNet are commonly used for classification or feature extraction, where each image is assigned a single label or a set of features. In contrast, semantic segmentation requires labeling each pixel, which significantly increases the complexity of the learning process.

Table 1
The coefficients of validation metrics evaluate the performance of neural network models. The similarity measure between the segmentation predicted by each neural network model and the reference segmentation ranges from 0 to 1, where 1 indicates complete overlap between the segmentations and 0 means no overlap.

CNN	#	Metric						
Model	Epochs	Accuracy	Loss	Value val_acc	Value val_loss	IoU	Dice	Jaccard
Hybrid model	50	0.8284	0.0134	**0.8246**	0.0118	**0.7761**	0.8565	0.7385
U-Net	90	0.8299	0.1078	0.8229	0.1543	0.7385	0.7287	0.6108
SegNet	150	0.8291	0.1186	0.8233	0.1466	0.6884	**0.8786**	**0.7739**
DeepLab	120	0.8279	**0.0016**	0.8219	**0.0034**	0.5439	0.8054	0.5751
Gated-SCNN	90	**0.8306**	0.1014	0.8176	0.2494	0.5101	0.8358	0.5065
Dragonfly	35		0.0057		0.0107			

Table 2
Comparison of some of the default models available in Ais.

Architecture	Metric		
	Training time[a]	Processing time[a],[b]	Vesicle and Membrane[c]
Hybrid model	114 s	617 ms	**0.0145**
EMAN2	**47 s**	**57 ms**	0.0594
InceptionNet	623 s	330 ms	0.0621
UNet	171 s	152 ms	0.0178
VGGNet	135 s	112 ms	0.0152
ResNet	1886 s	870 ms	0.0649

[a] The computational cost is only roughly proportional to the number of model parameters. The specificities of the network architecture have a more significant impact on processing speed.

[b] Time required to process one 224 × 224 pixel sized tomographic slice.

[c] These columns list the loss values after training, calculated as the binary cross-entropy (BCE) between the predicted values and the original annotations. The loss is a (rough) metric of how well a trained network performs.

Our hybrid model incorporates elements from various semantic segmentation architectures, which necessitates a larger dataset to ensure effective generalization and stability during training, helping to prevent overfitting. By utilizing a substantial amount of data for the hybrid model, it can achieve its optimal performance. This, in turn, provides a more meaningful comparison with individual architectures that are trained on smaller datasets.

For the comparison against MemBrain-seg, we standardized the pixel size to 10 Å, and data preprocessing involved pixel-size matching, Fourier amplitude matching, and Wiener-like deconvolution. Moreover, for data augmentation with MemBrain-seg, we used two primary techniques: missing wedges and Fourier amplitude boosting. In the missing-wedge data augmentation method, input subtomograms were randomly rotated, and an artificial missing wedge was introduced by masking the Fourier coefficients into a wedge-like shape. Additionally, a variety of geometric transformations were applied, which included rotations at arbitrary angles, zooming within the range [0.7,1.4], as well as axis shuffling and flipping.

Furthermore, we implemented intensity transformations to modify image characteristics, such as median filtering, Gaussian blurring, the addition of Gaussian noise, adjustments to brightness and contrast, low-resolution simulation, random erasing, the application of an additive brightness gradient, local Gamma transformation, and sharpening.

We used binary cross-entropy, Dice loss, and Surface-Dice loss as loss functions for the MemBrain-seg architecture. The Surface-Dice loss not only serves as a metric for evaluating the binarized segmentation results but also functions as a loss during network training, utilizing a differentiable skeletonization method. A polynomial scheduler managed the learning rate, which gradually decreased from 0.01 to zero. Before training, the patch intensities were normalized by subtracting their means and dividing by their standard deviations. The training process consists of five rounds, equivalent to epochs, of incremental training.

As recommended in Kiewisz et al. (2024), for TARDIS we implemented a normalization strategy with a pixel size of 10 Å. First, each slice of the tomogram undergoes intensity normalization, where the mean intensity is subtracted from the values and then divided by the standard deviation. We extract patches of size 224 × 224 pixels from the micrographs. The dataset was then split, with 80% allocated for training and 20% for validation.

Training was conducted using the NAdam optimizer with a learning rate of 0.0001 and binary cross-entropy (BCE) as the loss function. To ensure standardized training across different modules and to prevent overfitting, an early stopping criterion was implemented. Training was halted if no improvement in the validation loss was observed for 250 epochs.

Semantic segmentation is assessed using several metrics, including the F1 score, precision, recall, and average precision (AP). In contrast, instance segmentation is evaluated using the mean coverage (mCov) score. The mCov metric measures the overlap between ground-truth data instances and predicted instances within the point cloud, utilizing IoU as the basis for measurement. These metrics range from 0 to 1, with higher values indicating better performance.

Published segmentation methods or available software, either commercial or public, report the metrics used for comparison. For comparison with those methods we tried to use the same training dataset and standard quantitative metrics, such as Dice and IoU. However, the values reported by other methods are presented as global aggregates, lacking details on variance or the distribution of those values. This limitation prevents direct statistical comparisons. As a result, statistical tests can only be conducted within the hybrid model itself.

We present in Appendix C one such statistical test. Because metric values are known at all times during the training process of the hybrid model, we can evaluate whether the observed differences are statistically significant and not the result of chance or inherent training variability. Having metric values for each epoch is similar to having repeated samples or paired data regarding the performance of the hybrid model.

3.3. Robustness evaluation

Each individual architecture possesses unique and complementary capabilities. The results from the hybrid model are difficult to explain

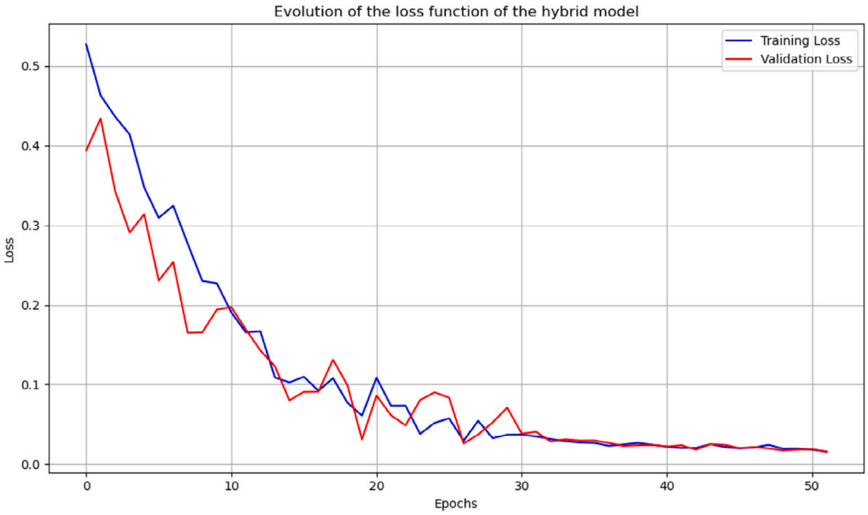


Fig. 6. Evolution of the loss function during hybrid model training. Progressive convergence is observed with a steady decrease in loss for both the training and validation datasets.

solely through its individual components, as it operates as a nonlinear and synergistic system: no single architecture encompasses all the necessary aspects for accurate segmentation in cryo-ET. For instance, the Gated-SCNN architecture is effective in delineating edges, which provide valuable input to the U-Net. The U-Net, in turn, integrate synergically with DeepLab' s multi-scale context. In turn, the LSTM architecture is useful for resolving ambiguities in image sequences that neither U-Net nor DeepLab can detect on their own.

Each architecture specializes in a specific type of feature, such as fine details, edges, context, or continuity, as shown in Table A.13. The fusion process functions like a committee that combines the decisions from each architecture to produce a more robust output. Each individual network transforms the input into a different latent space, and by fusing these representations, we achieve a richer and more expressive embedding. This enhances the accuracy of membrane and vesicle segmentation. Additionally, with the integration of spatial and channel attention mechanisms, the hybrid model learns to dynamically select which parts of each individual architecture are most beneficial based on the context. For example, in noisy regions, the model may rely more on LSTM, while in well-defined regions, it may depend more on Gated-SCNN.

It is challenging to determine the specific contribution of each module; however, the hybrid model was empirically tested through an ablation study. In this study, the hybrid model was compared to each of the individual architectures: U-Net, SegNet, DeepLab, and Gated-SCNN, as shown in Table 1. A detailed description of each model can be found in Appendix B. Each model is evaluated based on the metrics outlined in Section 2.3. The results of the ablation study are discussed in Section 4.

4. Results

We present the performance results of the hybrid model that was pre-trained using generated synthetic data. In Fig. 5, we compare the predictions of the hybrid neural network model with the reference segmentations. These predictions primarily depict membranes (spherical, ellipsoidal, and toroidal) with various combinations of local curvatures.

Although the IoU coefficient is 0.6941, our primary goal is to ensure that the hybrid model effectively learns useful representations of typical shapes, textures, and structures of membranes and vesicles. We aim to include controlled variability in orientation, shape, contrast, and noise, and to establish reasonable initial weights before fine-tuning with real data. This approach accelerates convergence, improves performance, and enhances the hybrid model's robustness against the real variations it will face in experimental data.

Through the pre-training process of segmenting additional structures, the hybrid architecture learns more comprehensive representations of cellular space. This approach can help identify unlabeled or poorly understood structures, enhance generalization on real data, and set the stage for refinement through fine-tuning. After pre-training, the hybrid model is then trained with real data. The datasets used in this experiment are described in Section 2.2. The performance metrics evaluating the hybrid model are shown in Table 1.

Fig. 6 shows the evolution of the loss function during training and validation of the hybrid model. In addition, Fig. 7 shows the performance curves that evaluate the implemented metrics. These metrics are a measure of the overlap between the prediction mask generated by the hybrid model and the reference mask. This measure ranges from 0 to 1, where 0 indicates no overlap and 1 indicates a complete overlap.

Moreover, the fact that both the training and validation losses converge close to zero with minimal gaps indicates that the hybrid model generalizes effectively. This suggests that data augmentation through rotation does not introduce excessive noise that could confuse the model. If the rotations were to adversely impact the SNR, we would likely observe a stagnation or divergence in the validation loss.

The results from the segmentation metrics (IoU, Dice, accuracy) on the validation and test datasets indicate that the hybrid model effectively identifies small structures and membranes without significant degradation in performance. The stability in segmentation metrics, such as the Dice coefficient and IoU, suggests that the hybrid model successfully learns discriminative features, even with data augmentation through interpolation. If the artifacts resulting from data augmentation had severely impacted the SNR or the morphology, we would have observed a decline in the hybrid model's performance.

In an ablation study that analyzes each model by examining the effects of its exclusion, we found that the hybrid model demonstrated the best overall balance across all segmentation metrics. Removing this model would negatively impact all metrics, particularly validation accuracy (val_acc of 0.8246) and Intersection over Union (IoU of 0.7761), which are critical balanced metrics. This suggests that the hybrid architecture is highly effective at accurately detecting the contours of both membranes and vesicles, which can be thin and challenging to segment.

The U-Net architecture showed acceptable performance in intermediate metrics, but it falls short compared to the hybrid model and SegNet in key metrics such as IoU (0.7385) and Dice (0.7287). While it served as a reliable baseline model, it was not the most optimal choice. U-Net could segment sharp edges effectively, but it often lost detail or struggled with small structures. Improvements were needed for it to

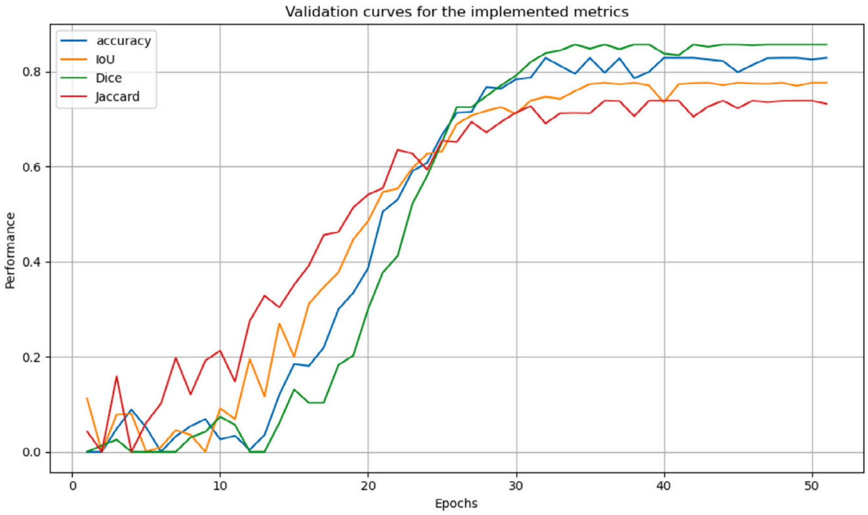


Fig. 7. The performance curves for the implemented metrics are displayed. This illustrates the simultaneous progression of the metrics that evaluate the performance of the hybrid model. The initial fluctuations represent the adjustment of the weights during the early epochs. As the curves stabilize, it indicates effective model generalization and demonstrates a good overlap between the hybrid model's predictions and the reference annotations.

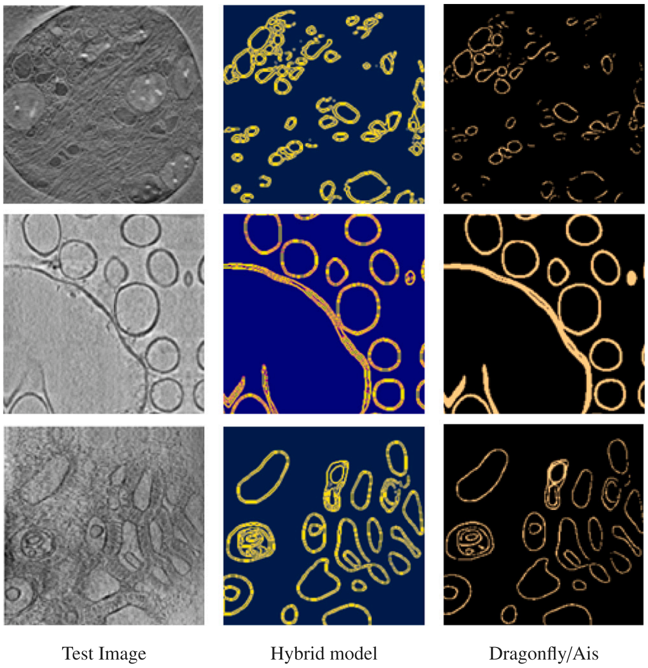


Fig. 8. Comparison of the predictions generated by the hybrid neural network model with other deep learning-based approaches. The top section presents the prediction of a 2D tomogram slice obtained using the hybrid model, visually compared with the results using Dragonfly software. The middle section shows the comparison of the mask segmented by the hybrid model with that generated by the EMAN2 network available in Ais. The bottom section shows the comparison between the mask predicted by the hybrid model and that obtained using the InceptionNet architecture available in Ais.

compete with SegNet or the hybrid model. If the U-Net were removed, the impact on global metrics would be moderate.

The SegNet architecture was highly effective for tasks that require fine-grained segmentation. It demonstrated superior performance in the Dice metric (0.8786) and the Jaccard index (0.7739). Simulating the exclusion of these metrics would likely result in a significant negative impact. Although SegNet had a lower IoU compared to the hybrid model, it excelled in metrics that are sensitive to false negatives. This architecture was particularly advantageous for detecting thin membranes and small vesicles, as it reduced the chances of "missing" tiny segments. While it might experience slight over-segmentation, it captured more critical structures effectively.

The DeepLab architecture was beneficial in situations where stability in loss is prioritized over precise segmentation. In terms of training and validation losses, it showed better performance with a training loss of 0.0016 and a validation loss of 0.0034. However, its Intersection over Union (IoU) score was relatively low at 0.5439, and its Jaccard index was 0.5751. DeepLab demonstrated excellent stability during training, which is beneficial for working with datasets that contain a lot of noise. While it could effectively segment smooth contours, it struggled with small structures, leading to omissions. Overall, the architecture was better at generalization than at achieving high accuracy in segmentation. Despite its lower accuracy, DeepLab could be a strong foundation for knowledge transfer training.

The Gated-SCNN architecture demonstrated a higher accuracy of 0.8306; however, it exhibited lower Jaccard accuracy (0.5065) and Intersection over Union (IoU) score (0.5101). This model was capable of effectively segmenting larger or frequently used areas, but it struggled with clearly defining edges. Additionally, it showed bias towards the dominant class, such as the background. While it serves as a helpful auxiliary model for refinement, its contribution leans more towards classification rather than segmentation. Removing this architecture would likely have a minimal overall impact.

In the case of the Dragonfly model, training was stopped at 35 epochs due to the lack of performance improvement. The U-Net variant implemented in this model only reports the validation loss, val_loss of 0.0107. In Fig. 8, we compare the predictions of the hybrid neural network model with Dragonfly segmentations in a selected image from the test database.

The comparison shown in Table 2 illustrates the binary cross-entropy metric of the hybrid model with various single-model architectures, including InceptionNet, ResNet, U-Net variant, VGGNet, and EMAN2's default architecture. In Figs. 8 and 9, we compare the predictions of the hybrid neural network model with those of the single-model architectures in selected images from the test database.

The hybrid model demonstrated the highest accuracy, with the lowest binary cross-entropy loss of 0.0145, and it clearly delineated membranes, vesicles, and the background. The segmented masks produced by the hybrid model show well-defined contours and accurately detect even small or near-noise vesicles.

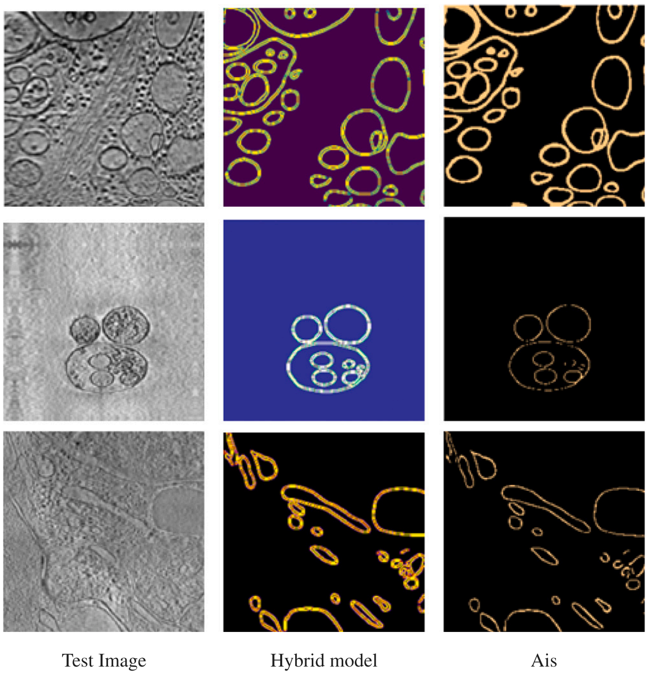


Fig. 9. Comparison of the predictions generated by the hybrid neural network model with other deep learning-based approaches. The top section presents the prediction of a 2D tomogram slice obtained using the hybrid model, visually compared with the results from a UNet variant of the Ais software. The middle section shows the comparison of the mask segmented by the hybrid model with that generated by the VGGNet network. The bottom section shows the comparison between the mask predicted by the hybrid model and that obtained using the ResNet architecture.

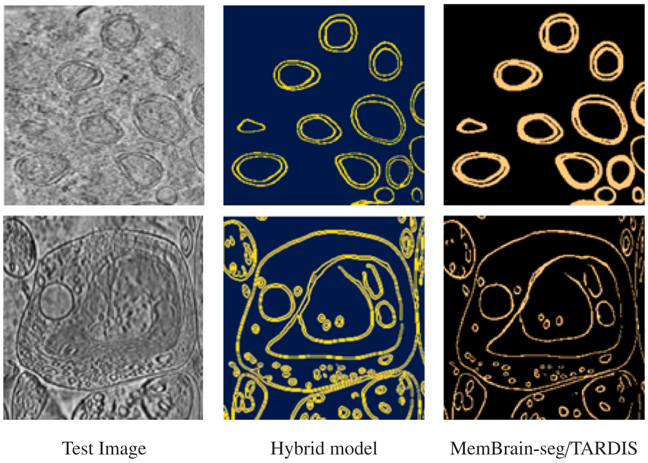


Fig. 10. Comparison of the predictions generated by the hybrid neural network model with other deep learning-based approaches. The top section presents the prediction of a 2D tomogram slice obtained using the hybrid model, visually compared with the results from MemBrain-seg. The bottom section shows the comparison between the mask predicted by the hybrid model and that obtained using the TARDIS architecture.

In the case of the VGGNet architecture, its loss performance of 0.0152 was very close to that of the hybrid model. It achieved segmentations that were similar to those of the hybrid model but had minor imperfections, such as slightly less defined edges. This model is an excellent choice if we are looking for architectural simplicity with minimal loss of accuracy.

The U-Net architecture performs slightly worse than the hybrid model, achieving an error rate of 0.0178. Its encoder–decoder design, which includes skip connections, helps to preserve spatial details. This structure yields more complete segmentations in continuous regions but may be sensitive to noise. In complex areas, it can produce slightly coarser or fragmented masks. While the U-Net is powerful for capturing large or continuous structures, it requires further fine-tuning to segment small objects, such as vesicles effectively.

The EMAN2 model demonstrated a high loss rate of 0.0594, indicating a weak performance in semantic segmentation. It tends to produce a significant number of false positives and negatives, resulting in fuzzy contours and inadequate detection of small structures. As it stands, EMAN2 was not suitable as a primary model for membrane and vesicle segmentation unless its architecture undergoes further improvements.

The InceptionNet model, while a deep architecture featuring multiple kernel sizes and parallel paths, demonstrated subpar loss performance (0.0621). It struggled to capture relevant contextual information necessary for accurately segmenting objects that closely resemble the background. This resulted in issues such as mask fragmentation, false positives where no relevant structures are present, and difficulty in detecting low-contrast vesicles.

The ResNet model was a slow architecture, achieving the highest loss result of 0.0649, which indicated poor performance. While its skip connections help with gradient propagation, they do not ensure sufficient spatial attention. The model demonstrated issues such as artifact generation, inconsistent segmentation between similar regions, and overfitting in areas with structural noise. Ultimately, its high complexity did not lead to improved performance.

Since each semantic segmentation model presented significant differences in architecture, learning, representation capacity, and convergence speed, we decided to use different numbers of epochs during training, using a batch size of 32. This strategy enabled the optimization of individual network performance. Each model has a different propensity for overfitting. Training all models with the same number of epochs can cause the models to either not learn enough (underfitting) or to memorize the data excessively (overfitting).

Additionally, in Fig. 10, we compare the predictions of the hybrid neural network model with MemBrain-seg and TARDIS segmentations in a selected image from the test database. The performance metrics evaluating the hybrid model are shown in Table 3. The hybrid model achieved the highest accuracy at 0.7145, indicating its strong ability to distinguish real membranes from artifacts. In comparison, TARDIS and MemBrain-seg achieved lower accuracy values of 0.6733 and 0.6524, respectively. High accuracy is crucial as it reflects a lower false positive rate, which is especially important in cryo-ET. This helps prevent the oversegmentation of diffuse structures, such as fragmented vesicles or adjacent membranes.

High recall is vital because it ensures that more true structures are detected, helping to avoid the omission of biologically significant details. The TARDIS model had an impressive recall score of 0.7253, which is an improvement over the hybrid model's recall score of 0.6925. This indicates that the TARDIS is more effective at capturing true positives and is more sensitive to faint structures, such as small vesicles and discontinuous membrane segments. However, this increased sensitivity comes at the cost of lower precision. As a result, higher recall with lower precision may lead to an increase in false positives, particularly in noisy areas.

The mCov (mean coverage) metric measures the average overlap between the predicted mask and the reference mask. TARDIS achieved the best mCov score of 0.7285, indicating a greater completeness in its segmentations. In contrast, MemBrain-seg exhibited a significantly lower mCov of 0.3542, suggesting poor overlap with the ground truth regions. This discrepancy occurs despite MemBrain-seg's high Dice score, which may indicate the presence of isolated or incomplete false positives.

Table 3
The coefficients of validation metrics evaluate the performance of neural network models. The similarity measure between the segmentation predicted by each neural network model and the reference segmentation ranges from 0 to 1, where 1 indicates complete overlap between the segmentations and 0 means no overlap.

CNN	#	Metric						
Model	Epochs	Accuracy	Loss	Value precision	Value recall	mCov	Dice	F1
Hybrid model	50	0.8284	0.0134	**0.7145**	0.6925	0.6813	0.8565	**0.7033**
TARDIS	50			0.6733	**0.7253**	**0.7285**		0.6812
MemBrain-seg	5			0.6524	0.7011	0.3542	**0.9251**	0.3751

In the case of the Dice Coefficient, MemBrain-seg achieved the highest value of 0.9251. However, this high value was not reflected in the F1 score or mCov, suggesting that the predictions were highly localized rather than extensive. This may indicate overfitting to specific membrane and vesicle classes or a bias toward certain regions. When segmenting long membranes or interconnected vesicles, this can result in significant topological errors.

In terms of the F1 score, the hybrid model achieved the highest score of 0.7033, indicating a strong balance between precision and recall. This suggests that it effectively detects real structures while minimizing false predictions. The TARDIS model follows closely with a score of 0.6812; however, MemBrain-seg performs significantly lower with a score of only 0.3751, despite its high Dice coefficient. This indicates that the predictions from MemBrain-seg are not useful for further quantitative analysis. Only available for the hybrid model are the metrics accuracy (0.8284) and loss (0.0134), indicating good convergence.

In Figs. 11 and 12, we compare the predictions of the hybrid neural network model with the reference segmentations in select images from the test database. The predictions of the hybrid model demonstrate precise contour segmentation. By combining local and global features, the model effectively handled long-range contexts and paid attention to fine details. As a result, it achieved accurate and detailed segmentation of structures in cryo-ET images.

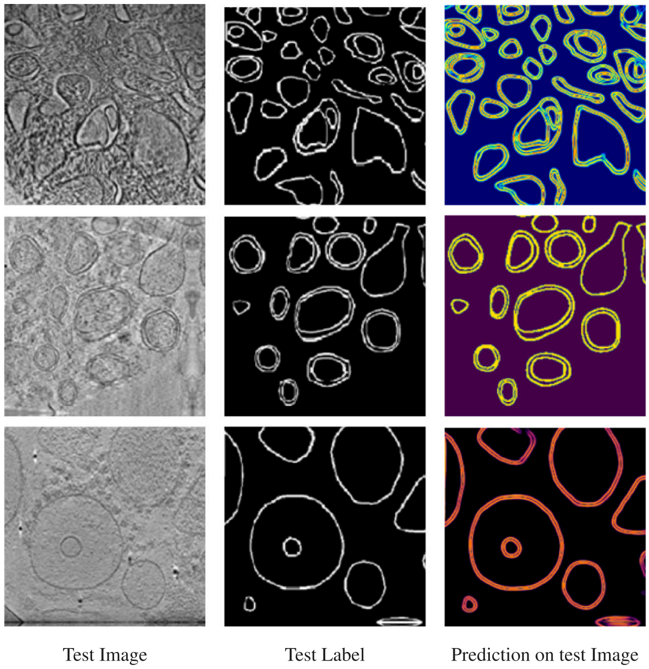


Fig. 11. Hybrid neural network model predictions for the datasets are as follows: 1. EMPIAR-10236 (top) showcases CryoET images of cis-mutated mouse protocadherin gamma B6 on membranes. 2. EMPIAR-12038 (middle) presents a Cryo-ET dataset of purified SARS-CoV-2 double membrane vesicles formed by nsp3-4. 3. EMPIAR-11751 (bottom) ER-derived vesicles from HEK293F cells. The images include a 2D tomogram slice, the input segmentation mask for the hybrid model, and the mask predicted by the model.

5. Discussion and conclusions

In this work, we present a novel approach for automated detection that efficiently segments various structural features of membranes and vesicles. This method closely mimics the performance of an expert human annotator, thereby saving valuable time and resources. We implement a hybrid model that combines the strengths of recurrent and convolutional neural networks. This integration allows us to capture fine details, facilitating the accurate segmentation of membranes and vesicles.

Some proposed methodologies work directly with 3D data without extensive filtering or retraining. However, this direct segmentation approach demands significant computing power. This can become costly, especially when handling large volumes of data, and it may be inefficient without powerful GPUs or high-performance clusters.

Additionally, earlier studies have relied on variations of U-Net, which restrict their adaptability to different scales and noise levels. Traditional U-Nets are designed to capture local contexts but are highly dependent on the quality and quantity of manually annotated training data. As a result, they may struggle to accurately segment complex, diffuse, or poorly defined structures, such as vesicles in cryo-ET.

The hybrid model addresses these limitations by combining multiple complementary architectures. This integration preserves spatial details, captures intricate features, facilitates accurate edge segmentation, and effectively handles complex cellular structures while reducing computational demands. Additionally, it employs multi-scale semantic segmentation by utilizing dilated convolutions to expand the receptive field and spatial attention modules, which are essential for segmenting membranes and vesicles of various sizes. The inclusion of attention mechanisms allows the hybrid model to filter out irrelevant information and emphasize meaningful features. As a result, it can capture fine edges and structural details, even when there are subtle variations in pixel intensity or when these changes are concealed by noise and artifacts. This capability is vital for segmenting thin membranes.

In addition, the hybrid architecture effectively captures long-term spatial, contextual, and temporal dependencies. This capability enhances semantic segmentation between successive slices in a 3D stack or within nearby regions. Its recurrent layer contributes to smoother segmentation and greater spatial coherence by learning temporal or sequential patterns from the extracted features.

The hybrid model is not limited to membranes and vesicles. However, it may encounter challenges when performing semantic segmentation on microorganisms or viruses with morphologies that differ from the training data. This model serves as a generic semantic segmentation architecture that can be adapted to other classes, provided it learns enough representative examples. To achieve good segmentation accuracy for new classes, fine-tuning or retraining with specific annotations for those structures is necessary. If the hybrid architecture has never encountered annotated examples of other structures, it will be unable to segment them accurately, even if it can detect their presence.

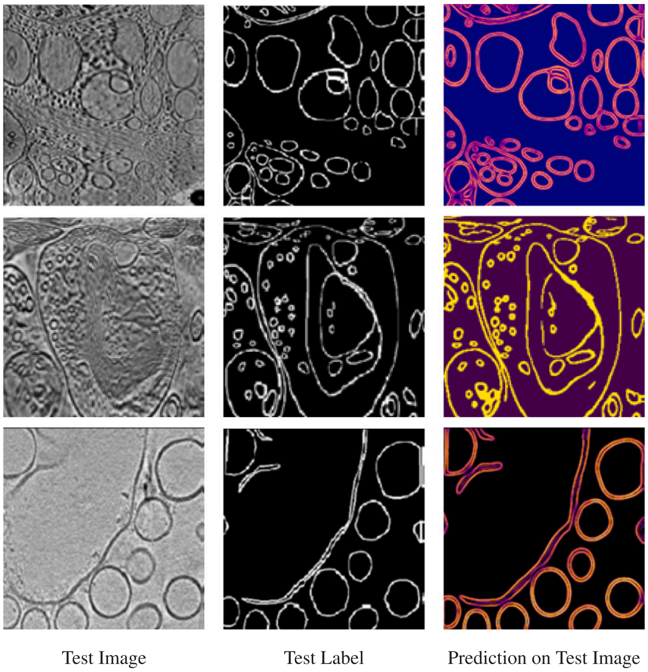


Fig. 12. Hybrid neural network model predictions for the datasets are as follows: 1. EMD-10439 (top) features an in situ tomogram of intact P19 cells acquired using a phase plate. 2. EMD-16084 (middle) includes the Munc13-SNAP25 cryo-ET dataset, specifically the synapse tomography of Munc13 DHet 115. 3. EMPIAR-10498 (bottom) examines how the arrangement of proteins at reconstituted synaptic vesicle fusion sites depends on membrane separation. The images presented include a 2D tomogram slice, the input segmentation mask for the hybrid model, and the mask predicted by the model.

If segmentation of additional cellular structures is needed, there are two approaches: multi-task or multi-class training. This can be accomplished using the categorical cross-entropy loss function. In cryo-ET, similar architectures, such as various versions of U-Net, have been employed to segment protein complexes like ribosomes, organelles such as mitochondria, microtubules, and filaments. This suggests that, with sufficient data, the hybrid model can be extended to include other structures.

The weights learned by our hybrid architecture effectively capture general features such as texture patterns, contours, and spatial relationships. These features are valuable for various related tasks. This hybrid architecture can serve as a foundation for specific applications in cryo-ET, including 3D reconstruction of cellular structures, segmentation of biological particles, and analysis of the structural variability of macromolecular complexes.

Potential practical applications for the hybrid model can be various, such as in structural biology laboratories that conduct repetitive analyses of similar cells. Our approach is particularly useful in studies involving vesicular trafficking or membrane fusion, because it can be specifically trained with annotations related to vesicles. Additionally, in clinical and pharmacological settings, the hybrid model would potentially enable fast segmentation of cellular components, something that is essential for modeling the effects of different compounds.

Code availability

The code generated in this study can be accessed at https://github.com/AlainMm/DLCryoTomo.

CRediT authorship contribution statement

Alain Morales-Martínez: Writing – original draft, Software, Methodology, Conceptualization. **Edgar Garduño:** Resources, Investigation, Funding acquisition. **José María Carazo:** Supervision. **Carlos Oscar S. Sorzano:** Validation. **José Luis Vilas:** Formal analysis.

Funding sources

The work of Alain Morales-Martínez was supported by the National Council for the Humanities, Sciences, and Technologies (CONAHCyT), México [Grant CVU229191, 2021]. A DGAPA-UNAM research grant IN107922 supported the work of Edgar Garduño and Alain Morales-Martínez.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Biocomputing Unit (BCU) of the National Center for Biotechnology (CNB-CSIC) and the Instruct Image Processing Center (I2PC) for the facilities granted for the preparation of this manuscript.

Appendix A. Hybrid architecture

The hybrid architecture features two essential components: a compression path, referred to as the encoder, and an expansion path, known as the decoder, as shown in Fig. A.13.

The encoder–decoder structure enables the transformation of a high-resolution image into a more abstract feature representation, which can then be reconstructed at a finer resolution. The incorporation of skip connections facilitates the incorporation of features extracted in the earlier layers of the network, as shown in Fig. A.13. This process is crucial for preserving fine spatial details.

We incorporate an Attention block, which includes both spatial and channel attention mechanisms (see Fig. A.14). These mechanisms enhance the robustness of the hybrid neural network against variations in image quality and noise.

The spatial attention mechanism enhances the efficiency of the hybrid network by focusing on identifying the most significant areas, leading to improved accuracy as the model can concentrate on the most critical areas and features for the segmentation task. Conversely, the channel attention mechanism enables the hybrid network to determine which input features should be prioritized by assigning them different levels of relevance (see Section 2.1). This channel attention is implemented as shown in Table A.4.

The implementation of the spatial attention mechanism begins with the incorporation of a single-7 × 7-filter 2D convolutional layer with a Sigmoid activation function and applying padding as necessary, see Fig. A.14.

We also incorporate a Convolutional block, consisting of two consecutive 2D convolutional layers, each using a 3 × 3-kernel, followed by a ReLU activation to introduce nonlinearity and enable the hybrid model to learn more complex representations of the extracted features, see Table A.5. Following these two consecutive layers, we add a channel attention mechanism trailed by one of spatial attention, see Fig. A.14.

The Encoder block consists of a Convolutional block followed by a MaxPooling2D layer, see Fig. A.14, a layer that reduces the spatial dimensions of the input image. Pooling is performed in 2 × 2 pixel blocks, which helps lowering the spatial resolution of the image while

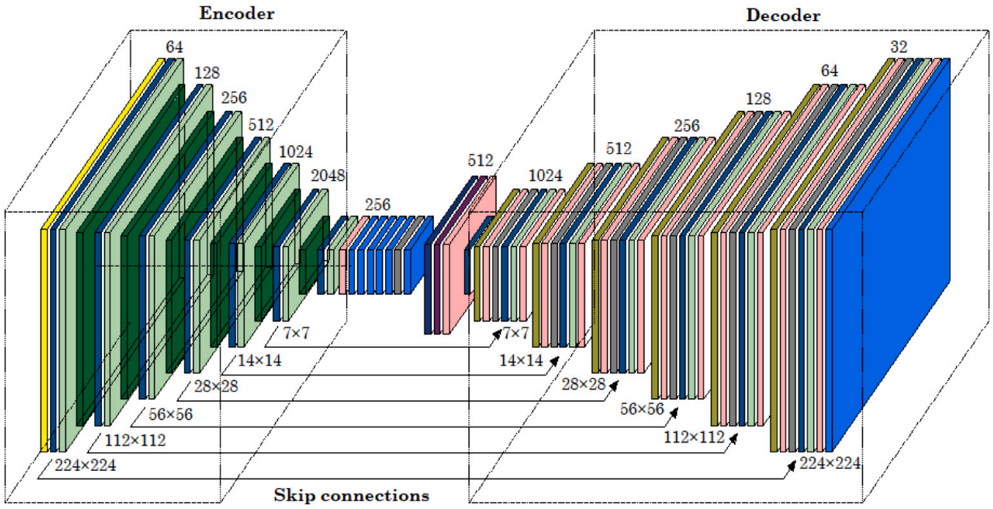


Fig. A.13. Hybrid neural network model combining an encoder–decoder architecture with skip connections, see details in Tables A.10–A.12.

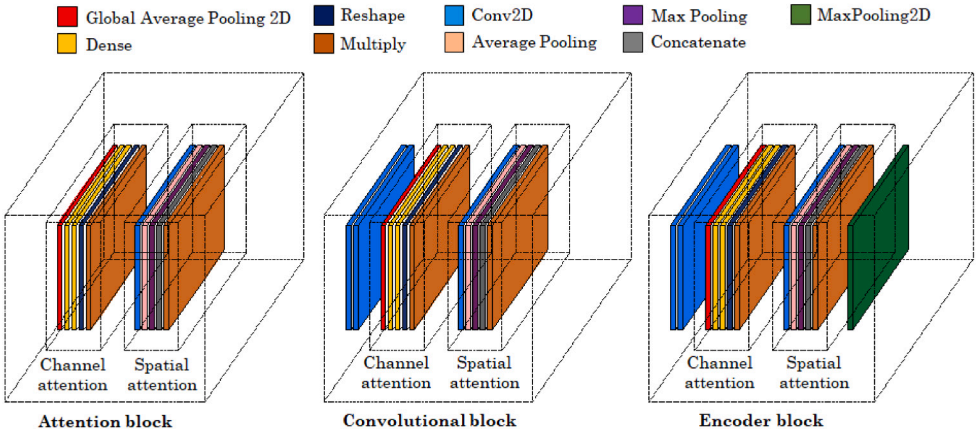


Fig. A.14. Spatial and channel attention mechanisms. Representation of the layers that form the attention, convolution, and encoder blocks, see details in Tables A.4–A.6.

Table A.4
The channel attention mechanism enables the hybrid network to determine which input features should be prioritized by assigning them different levels of relevance. This mechanism is implemented for 64, 128, 256, 512, 1024, and 2048 filters, see Section 2.1.

Layer	Hybrid model (Channel Attention block)					
type	No. filters	Size	No. params	Output	Activation function	Options
GlobalAveragePooling2D	64			(64)		
Dense	4		260	(4)	ReLu	
Dense	64		320	(64)	Sigmoid	
Reshape	64	1 × 1		(1,1,64)		
Multiply	64			(224,224,64)		

Table A.5
The Convolutional block is implemented for 64, 128, 256, 512, 1024, 2048, and 4096 filters, see Section 2.1.

Layer	Hybrid model (Convolutional block)					
type	No. filters	Size	No. params	Output	Activation function	Options
Conv2D	64	3 × 3	640	(224,224,64)	ReLu	padding = same
Conv2D	64	3 × 3	36,928	(224,224,64)	ReLu	padding = same
GlobalAveragePooling2D	64			(64)		
Dense	4		260	(4)	ReLu	
Dense	64		320	(64)	Sigmoid	
Reshape	64	1 × 1		(1,1,64)		
Multiply	64			(224,224,64)		
Spatial Attention	1	7 × 7	99	(224,224,64)	Sigmoid	padding = same

Table A.6
The Encoder block is implemented for 64, 128, 256, 512, 1024, 2048, and 4096 filters.

Layer	Hybrid model (Encoder block)					
type	No. filters	Size	No. params	Output	Activation function	Options
Conv2D	64	3 × 3	640	(224,224,64)	ReLu	padding = same
Conv2D	64	3 × 3	36,928	(224,224,64)	ReLu	padding = same
GlobalAveragePooling2D	64			(64)		
Dense	4		260	(4)	ReLu	
Dense	64		320	(64)	Sigmoid	
Reshape	64	1 × 1		(1,1,64)		
Multiply	64			(224,224,64)		
Spatial Attention	1	7 × 7	99	(224,224,64)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(112,112,64)		

Table A.7
Atrous Spatial Pyramid Pooling (ASPP) block uses multiple convolutions with different dilation rates, see Section 2.1.

Layer	Hybrid model (ASPP block)					
type	No. filters	Size	No. params	Output	Activation function	Options
Conv2D	256	1 × 1	1,048,832	(3,3,256)	ReLu	padding = same
Conv2D	256	3 × 3	9,437,440	(3,3,256)	ReLu	dilation rate = 6
Conv2D	256	3 × 3	9,437,440	(3,3,256)	ReLu	dilation rate = 12
Conv2D	256	3 × 3	9,437,440	(3,3,256)	ReLu	dilation rate = 18
Conv2D	256	1 × 1	1,048,832	(3,3,256)	ReLu	padding = same
Concatenate				(3,3,1280)		
Conv2D	256	1 × 1	327,936	(3,3,256)	ReLu	padding = same

Table A.8
The LSTM layer has an unique capability to process sequential data and learn long-term patterns, see Section 2.1.

Layer	Hybrid model (LSTM block)					
type	No. filters	Size	No. params	Output	Activation function	Options
Reshape	256	1 × 1		(9,256)		
LSTM	512		1,574,912	(9,512)		
Dropout				(9,512)		$p = 0.5$
Reshape	512	1 × 1		(3,3,512)		

Table A.9
The Decoder block is implemented for 1024, 512, 256, 128, 64 and 32 filters.

Layer	Hybrid model (Decoder block)					
type	No. filters	Size	No. params	Output	Activation function	Options
Conv2DTranspose	1024	2 × 2 stride	4,719,616	(6,6,1024)		padding = same
ZeroPadding2D	1024			(7,7,1024)		padding = same
Dropout				(7,7,2048)		$p = 0.5$
Concatenate				(7,7,3072)		
Conv2D	1024	3 × 3	28,312,576	(7,7,1024)	ReLu	padding = same
Conv2D	1024	3 × 3	9,438,208	(7,7,1024)	ReLu	padding = same
GlobalAveragePooling2D	1024			(1024)		
Dense	64		65,600	(64)	ReLu	
Dense	1024		66,560	(1024)	Sigmoid	
Reshape	1024	1 × 1		(1,1,1024)		
Multiply	1024			(7,7,1024)		
Spatial Attention	1	7 × 7	99	(7,7,1024)	Sigmoid	padding = same

preserving the most important features, see Table A.6. This process also reduces the computational complexity in the later stages.

In this way, the compression path consists of six encoder blocks, each with 64, 128, 256, 512, 1024, and 2048 filters, as shown in Section 2.1. To prevent overfitting, each of these blocks is followed by a Dropout layer, as shown in Fig. A.15, with a dropout rate set to $p = 0.5$ (i.e., during training, 50% of the neurons in the Dropout layer are randomly deactivated at each training step).

Importantly, we incorporate an Atrous Spatial Pyramid Pooling (ASPP) block that uses multiple convolutions with different dilation rates; these are used to determine the spacing between filter elements, allowing each filter to cover a larger area without increasing its size or the total number of parameters, see Fig. A.15. The ASPP block is implemented in according to Table A.7.

We integrate an LSTM layer for its unique capability to process sequential data and learn long-term patterns. Thus, the Conv2D output from the ASPP block is reshaped to be fed into an LSTM layer, which consists of 512 filters (see Fig. A.15). The LSTM layer is implemented as illustrated in Table A.8.

The Decoder block includes a layer that performs transposed convolution (i.e., deconvolution) to increase the resolution of the input features and serves as an upsampling method.

Then, we concatenate the upsampling result with the skip connection to merge detailed information from the encoder with processed

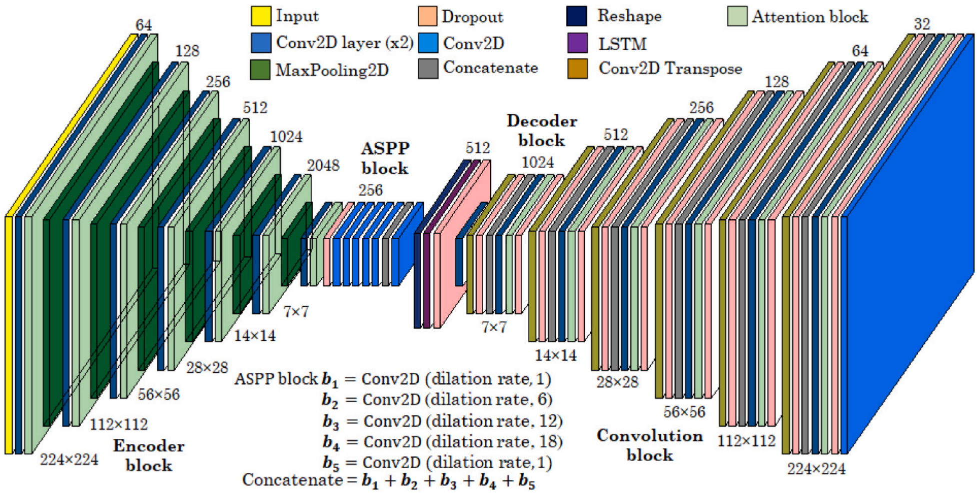


Fig. A.15. Hybrid neural network model combining an encoder–decoder architecture with attention blocks and dilated convolutions. This design enables multi-scale segmentation, capturing fine details. Additionally, the model's capacity to sequentially process extracted features allows it to identify temporal patterns and relationships, see details in Tables A.10–A.12.

Table A.10
Hybrid neural network model that combines recurrent and convolutional neural networks, with spatial and channel attention mechanisms, dilated convolutions, and that captures long-term spatial, contextual, and temporal dependencies, see Section 2.1.

Layer	Hybrid model (Section A)					
type	No. filters	Size	No. params	Output	Activation function	Options
Input		224 × 224				
Conv2D	64	3 × 3	640	(224,224,64)	ReLu	padding = same
Conv2D	64	3 × 3	36,928	(224,224,64)	ReLu	padding = same
Channel Attention	64					
Spatial Attention	1	7 × 7	99	(224,224,64)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(112,112,64)		
Conv2D	128	3 × 3	73,856	(112,112,128)	ReLu	padding = same
Conv2D	128	3 × 3	147,584	(112,112,128)	ReLu	padding = same
Channel Attention	128					
SpatialAttention	1	7 × 7	99	(112,112,128)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(56,56,128)		
Conv2D	256	3 × 3	295,168	(56,56,256)	ReLu	padding = same
Conv2D	256	3 × 3	590,080	(56,56,256)	ReLu	padding = same
Channel Attention	256					
Spatial Attention	1	7 × 7	99	(56,56,256)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(28,28,256)		
Conv2D	512	3 × 3	1,180,160	(28,28,512)	ReLu	padding = same
Conv2D	512	3 × 3	2,359,808	(28,28,512)	ReLu	padding = same
Channel Attention	512					
SpatialAttention	1	7 × 7	99	(28,28,512)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(14,14,512)		
Conv2D	1024	3 × 3	4,719,616	(14,14,1024)	ReLu	padding = same
Conv2D	1024	3 × 3	9,438,208	(14,14,1024)	ReLu	padding = same
Channel Attention	1024					
SpatialAttention	1	7 × 7	99	(14,14,1024)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(7,7,1024)		

information from the decoder. A final refinement is achieved by passing the output through a Convolutional block, as shown in Table A.9.

Therefore, the expansion path consists of six decoder blocks with 1024, 512, 256, 128, 64, and 32 filters, as shown in Fig. A.15. These blocks reverse the encoding process to put together the segmented image from its compressed features. At each step, a convolution block is applied to enhance the image resolution, and the outputs from the corresponding encoder blocks are concatenated. Finally, our hybrid network includes a Conv2D layer with a 1 × 1 filter and Sigmoid activation to generate the final output (see Table A.13).

Ultimately, in Table A.10, Tables A.11 and A.12, we show the different types of layers that make up the elements of hybrid architecture.

Appendix B. Individual architectures

The U-Net uses spatial and channel-wise attention mechanisms using a 2D convolution layer with a 7 × 7 filter and Sigmoid activation to generate these attention maps. The compression path of this architecture consists of layers of 16, 32, 64, and 128 filters. In addition, Global Average Pooling is introduced, which is then processed by two Dense layers to represent attention. Both layers, along with a Residual block that improves training stability, are applied in the expansion path using layers of 128, 64, 32, and 16 filters, respectively. In this case, the Residual blocks facilitate training by using *skip connections* that allow information and gradient to flow more easily through the architecture. While the expansion path utilizes Residual and Dropout

Table A.11
Hybrid neural network model that combines recurrent and convolutional neural networks, with spatial and channel attention mechanisms, dilated convolutions, and that captures long-term spatial, contextual, and temporal dependencies, see Section 2.1.

Layer	Hybrid model (Section B)					
type	No. filters	Size	No. params	Output	Activation function	Options
Conv2D	2048	3 × 3	18,876,416	(7,7,2048)	ReLu	padding = same
Conv2D	2048	3 × 3	37,750,784	(7,7,2048)	ReLu	padding = same
Channel Attention	2048					
Spatial Attention	1	7 × 7	99	(7,7,2048)	Sigmoid	padding = same
MaxPooling2D		2 × 2 pool size		(3,3,2048)		
Conv2D	4096	3 × 3	75,501,568	(3,3,4096)	ReLu	padding = same
Conv2D	4096	3 × 3	150,999,040	(3,3,4096)	ReLu	padding = same
Channel Attention	4096					
Spatial Attention	1	7 × 7	99	(3,3,4096)	Sigmoid	padding = same
Dropout				(3,3,4096)		$p = 0.5$
ASPP block	256					
LSTM block	512					
Conv2DTranspose	1024	2 × 2 stride	4,719,616	(6,6,1024)		padding = same
ZeroPadding2D	1024			(7,7,1024)		padding = same
Dropout				(7,7,2048)		$p = 0.5$
Concatenate				(7,7,3072)		
Convolutional block	1024					
Dropout				(7,7,1024)		$p = 0.5$
Conv2DTranspose	512	2 × 2 stride	4,719,104	(14,14,512)		padding = same
Dropout				(14,14,1024)		$p = 0.5$
Concatenate				(14,14,1536)		
Convolutional block	512					
Dropout				(14,14,512)		$p = 0.5$
Conv2DTranspose	256	2 × 2 stride	1,179,904	(28,28,256)		padding = same
Dropout				(28,28,512)		$p = 0.5$
Concatenate				(28,28,768)		
Convolutional block	256					
Dropout				(28,28,256)		$p = 0.5$

Table A.12
Hybrid neural network model that combines recurrent and convolutional neural networks, with spatial and channel attention mechanisms, dilated convolutions, and that captures long-term spatial, contextual, and temporal dependencies, see Section 2.1.

Layer	Hybrid model (Section C)					
type	No. filters	Size	No. params	Output	Activation function	Options
Conv2DTranspose	128	2 × 2 stride	295,040	(56,56,128)		padding = same
Dropout				(56,56,256)		$p = 0.5$
Concatenate				(56,56,384)		
Convolutional block	128					
Dropout				(56,56,128)		$p = 0.5$
Conv2DTranspose	64	2 × 2 stride	73,792	(112,112,64)		padding = same
Dropout				(112,112,128)		$p = 0.5$
Concatenate				(112,112,192)		
Convolutional block	64					
Dropout				(112,112,64)		$p = 0.5$
Conv2DTranspose	32	2 × 2 stride	73,792	(224,224,32)		padding = same
Dropout				(224,224,64)		$p = 0.5$
Concatenate				(224,224,96)		
Convolutional block	32					
Dropout				(224,224,32)		$p = 0.5$
Conv2D	1	1 × 1	33	(224,224,1)	Sigmoid	

blocks to prevent overfitting, it is followed by a 2 × 2 MaxPooling filter to increase the receptive field.

In the case of the SegNet architecture, it uses a pre-trained ResNet50 model (He et al., 2016) as the *encoder*; this way, the encoder extracts features using only the convolutional layers and excludes the fully connected layers at the end of the ResNet50 model, which captures features at different abstraction levels. In the expansion path, the spatial dimensions are doubled and 3 × 3 successive convolutions with 512 filters are applied along with Batch Normalization, concatenating the *encoder* features at each stage. This process is repeated in subsequent decoding stages with different filter layers: 256, 128, and 64. The process concludes with a 1 × 1 convolution and a Sigmoid activation function, which produces a binary segmentation mask.

The DeepLab architecture utilizes an encoder–decoder structure with skip connections. It includes an ASPP block to capture contextual information at multiple scales through the use of dilated convolutions, specifically with rates of 1, 6, 12, and 18. Dilated convolutions enhance the receptive field without sacrificing resolution. Additionally, the model incorporates spatial attention modules. The contraction path begins with an encoding block that utilizes an initial convolutional layer to extract fundamental image features, followed by Batch Normalization and ReLU activation. The spatial dimension is then reduced using a MaxPooling2D layer with a 2 × 2 kernel size, increasing the feature depth to 64, 128, 256, and 512 filters in successive blocks. In the ASPP block, 2D convolutions with a 3 × 3-kernel size are performed, applying Batch Normalization and ReLU activation at various dilation rates. Subsequently, Global Average Pooling and 1 × 1 convolutions are

Table A.13
Overview of the components integrated into the proposed hybrid model for semantic segmentation in cryo-ET. Each component plays a unique role in enhancing the segmentation of membranes and vesicles, especially in low signal-to-noise conditions. The U-Net and SegNet architectures focus on optimizing spatial preservation and capturing local details. DeepLab improves contextual understanding through the use of dilated convolutions. Gated-SCNN is effective for edge detection, particularly in data with a high noise level. Long Short-Term Memory (LSTM) models maintain volumetric continuity across tomogram slices. Additionally, the incorporation of a Generative Adversarial Network (GAN) aids in generating segmentations that are structurally plausible. This comprehensive integration addresses the inherent challenges associated with cryo-ET data, including structural heterogeneity, noise, and low resolution.

Architecture	Component	
	Key contribution	Improvement in cryo-ET
U-Net	Fusion of context and local details	Precise capture of fine structures
SegNet	Upsampling guided by pooling indexes	Precise spatial preservation
DeepLab	Multiscale context and dilated convolutions	Handling of variable size structures
Gated-SCNN	Edge detection and refinement	Sharper contours in high-noise environments
LSTM	Modeling continuity between slices (3D volume)	Structural coherence in 3D
GAN	Generation of additional synthetic data	Provides more variability and examples

employed to tune the number of filters and scale the dimensions of the output. In the expansion path, the spatial dimension is expanded by an 8×8 Upsampled layer, followed by convolutions with 256 filters. It concludes with a 1×1 convolution and Sigmoid activation to generate a binary segmentation mask.

In addition, for the Gated-SCNN architecture we utilize a variant of the DeepLabV3 architecture (Papandreou et al., 2017), where the *encoder* is implemented using a pre-trained ResNet50 model (He et al., 2016) as a feature extractor. Furthermore, the fully connected layers are dropped at the end of the ResNet50 model, using only the convolutional layers. This architecture also includes an ASPP block and introduces Global Average Pooling to reduce the dimensions of the output of convolutional layers to a more compact and global representation, which helps simplify the architecture and avoid overfitting. A 7×7-convolutional layer with 256 filters, ReLU activation, and dilation rates of 6, 12, and 18 is used to extract features at different resolutions. The spatial size is then resized using bilinear interpolation, and the ASPP block is used to extract multi-scale features. Finally, an 8×8 upsampled layer and a 1×1 convolution with Sigmoid activation are used to obtain the segmentation predictions.

Finally, a Generative Adversarial Network (GAN) architecture is utilized to generate additional synthetic data. Specifically, a Pix2Pix model is implemented, which consists of two neural networks: a generator (G) with a U-Net architecture that takes a mask as input and produces a realistic synthetic image, and a discriminator (D) with a PatchGAN architecture that assesses mask–image pairs to determine whether each image is real (from the original dataset) or generated by G. The real dataset used comprises 1,200 images, each with a size of 224×224 pixels. To simulate structural diversity, transformations like random rotations, flips, and Gaussian blur are applied to the masks. During the training process, the generator optimizes a loss function that combines both the adversarial loss (L_{GAN}) and the $L1$ loss, while the discriminator employs only binary cross-entropy. The training phase lasts for 100 epochs. Once training is complete, the model generates a new dataset consisting of pairs of synthetic images and the corresponding known masks. This allows the hybrid model to be trained without the need for additional manual annotations.

Appendix C. Statistical tests

The Friedman statistic is utilized to compare different treatments or conditions in experimental designs that involve repeated measures or paired data, particularly when the assumptions of normality are not satisfied, making it a nonparametric test. In this paper, it will be used to determine whether there are significant differences among the various metrics employed to evaluate the performance of the hybrid model. Given a set of n epochs, each evaluated with k different metrics, the Friedman statistic is calculated as follows:

$$\mathcal{X}_F^2 = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2\right] - 3n(k+1)$$

where n number of epochs, k number of metrics being compared, and R_j sum of ranks of metric j across the entire hybrid model training dataset. Each epoch is evaluated using each metric, and the results are then converted into ranks for that epoch.

During the hybrid model training, segmentation metrics are calculated for each epoch. Ranks are assigned based on these metrics (with 1 being the best and k being the worst; for example, a higher Dice coefficient indicates better performance). The value R_j is determined as the sum of ranks for each segmentation metric. The Friedman test is then applied to assess whether there is a significant difference in performance, which can be indicated by either a critical value of $\mathcal{X}^2$ or a p-value.

Under the null hypothesis, which posits that there is no difference between the metrics, the statistic $\mathcal{X}_F^2$ approximately follows a chi-square distribution with $k-1$ degrees of freedom when $n \geq 10$. The p-value is then calculated as follows:

$$p = P(\mathcal{X}^2 \geq \mathcal{X}_F^2)$$

This represents the area to the right of $\mathcal{X}_F^2$ on the chi-square curve with $k-1$ degrees of freedom. If $p < \alpha$ (typically 0.05), the null hypothesis is rejected, indicating that significant differences exist between the metrics. If $p \geq \alpha$, then no significant differences are evident.

The hybrid model was trained over 50 epochs, during which the accuracy, Dice coefficient, Intersection over Union (IoU), and Jaccard metrics were evaluated at each epoch. The Friedman statistic calculated for this evaluation is 3.114, and the p-value is 0.37445. This indicates that there is insufficient statistical evidence to conclude that the metrics behave significantly differently from one another, see Fig. C.16.

If the metrics show convergence, it is justifiable to use a primary metric (like Dice or IoU) without compromising representativeness. This approach simplifies comparisons with other proposed methods and aids in the interpretation of the results.

To enhance this study, we conduct a convergence and stability analysis of the hybrid model over recent epochs. This involves running descriptive statistics on the validation dataset, where we calculate the mean, standard deviation, maximum and minimum values, as well as a 95% confidence interval (CI) for the last 10 epochs, as shown in Table C.14.

Descriptive statistics, along with a 95% confidence interval (CI), provide insights into the average value of each final metric, the variation of each metric (measured by standard deviation), and the expected range (95% confidence interval) in which the true performance of the hybrid model is likely to be at the conclusion of training. This allows us to analyze how much each metric changes between consecutive epochs. If the change is small on average, it indicates stability or convergence.

In according to Table C.14, the mean change for each metric over the last 10 epochs is very low. This indicates that the metrics have remained stable during this period. Ultimately, this provides statistical evidence that the hybrid model has demonstrated robust, stable, and reliable performance.

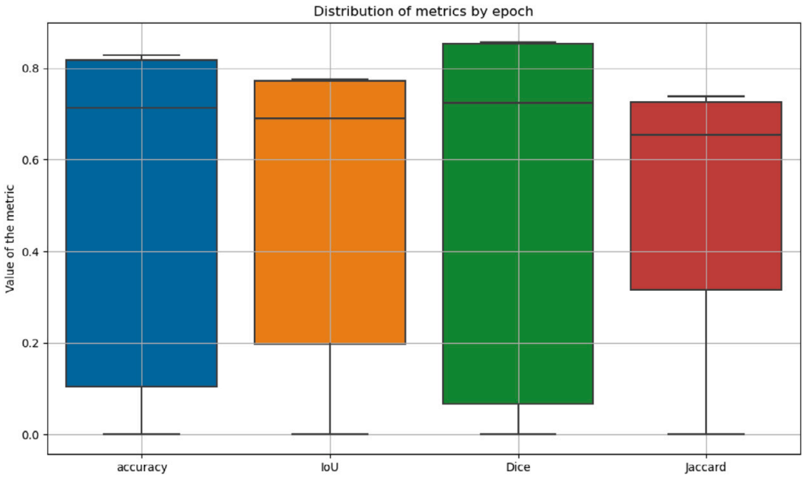


Fig. C.16. Distribution of metrics by epoch. The hybrid model demonstrates consistency across all metrics, with none of them standing out or exhibiting radically different behavior. This is particularly important since the metrics assess different aspects of performance, such as overlap, area, and sensitivity.

Table C.14
Statistical results of the hybrid model validation metrics are presented. This includes the final average values for performance metrics such as Accuracy, IoU, Dice, and Jaccard, along with their standard deviations (STD) and 95% confidence intervals. Additionally, the average changes from previous epochs and a qualitative stability rating are included. All results correspond to the final stage of model training.

Metrics	Statistical summary of validation metrics				
	Final mean	STD	IC 95%	Mean change	Stability
Accuracy	0.8226	0.0092	(0.8156, 0.8295)	0.007412	High
IoU	0.7745	0.0092	(0.7728, 0.7762)	0.003115	High
Dice	0.8558	0.0015	(0.8547, 0.8570)	0.001482	High
Jaccard	0.7313	0.0104	(0.7234, 0.7391)	0.008746	High

Data availability

The datasets generated and analyzed in this study can be accessed at https://github.com/AlainMm/DLCryoTomo.

References

Agulleiro, J.I., Fern andez, J.J., 2011. Fast tomographic reconstruction on multicore computers. J. Bioinform. 27, 582–583.

andler, M.S., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2019. Mobile net V2: Inverted residuals and linear bottlenecks. arXiv, 1801.04381.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2016. Seg net: A deep convolutional encoder- decoder architecture for image segmentation. arXiv, 1511.00561.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40, 834–848.

Conesa, P., Fonseca, Y., de la Morena, J.J., Sharov, G., de la Rosa-Trevín, J.M., Cuervo, A., Mena, A.G., de Francisco, B.R., del Hoyo, D., Herreros, D., Marchán, D., Střelák, D., Fernández-Giménez, E., Ramírez-Aportela, E., de Isidro-Gómez, F.D., Sánchez, I., Krieger, J., Vilas, J., del Caño, L., Gragera, M., Iceta, M., Martínez, M., Losana, P., Melero, R., Marabini, R., Carazo, J., Sorzano, C., 2023. Scipion3: A workflow engine for cryo-electron microscopy image processing and structural biology. Biol. Imaging 3.

Dai, W., Chen, M., Sun, S.Y., Jonasch, D., He, C.Y., Schmid, M.F., Chiu, W., Ludtke, S.J., 2017. Convolutional neural networks for automated annotation of cellular cryo-electron tomograms. Nature Methods 14, 983–985.

de Mariscal, E.G., Maška, M., Kotrbová, A., Pospíchalová, V., Matula, P., Muñoz-Barrutia, A., 2019. Deep- learning- based segmentation of small extracellular vesicles in transmission electron microscopy images. Sci. Rep. 9.

de Teresa, I., Goetz, S., Mattausch, A., Stojanovska, F., Zimmerli, C., Toro-Nahuelpan, M., Cheng, D., Tollervey, F., Pape, C., Beck, M., Kreshuk, A., Mahamid, J., Zaugg, J.B., 2022. Convolutional networks for supervised mining of molecular patterns within cellular context. Nature Methods 20, 284–294.

Dutta, A., Zisserman, A., 2019. The VIA annotation software for images, audio and video. In: Proceedings of the 27th ACM International Conference on Multimedia MM '19. ACM, New York, NY, USA.

Ermel, U., Cheng, A., Ni, J.X., Gadling, J., Venkatakrishnan, M., Evans, K., Asuncion, J., Sweet, A., Pourroy, J., Wang, Z.S., Khandwala, K., Nelson, B., McCarthy, D., Wang, E.M., Agarwal, R., Carragher, B., 2024. A data portal for providing st andardized annotations for cryo-electron tomography. Nature Methods 21 (12), 2200–2202.

Fatima, N., 2020. Enhancing performance of a deep neural network: A comparative analysis of optimization algorithms. ADCAIJ: Adv. Distrib. Comput. Artif. Intell. J. 9.

Google LLC, 2017. Google colab. https://colab.research.google.com.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778.

Hecksel, C.W., Darrow, M., Dai, W., Galaz-Montoya, J.G., Chin, J.A., Mitchell, P., Chen, S., Jakana, J., Schmid, M., Chiu, W., 2016. Quantifying variability of manual annotation in cryo- electron tomograms. Microsc. Microanal. 22, 487–496.

Heebner, J.E., Purnell, C., Hylton, R.K., Marsh, M., Grillo, M.A., Swulius, M.T., 2022. Deep learning- based segmentation of cryo- electron tomograms. J. Vis. Exp.: JoVE 189.

Iudin, A., Korir, P.K., Somasundharam, S., Weyand, S., Cattavitello, C., Fonseca, N., Salih, O., Kleywegt, G., Patwardhan, A., 2022. E MPIAR: the electron microscopy public image archive. Nucleic Acids Res. 51, D1503–D1511.

Jiménez de la Morena, J., Conesa, P., Fonseca, Y., de Isidro-Gómez, F., Herreros, D., Fernández-Giménez, E., Strelak, D., Moebel, E., Buchholz, T., Jug, F., Martinez-Sanchez, A., Harastani, M., Jonic, S., Conesa, J., Cuervo, A., Losana, P., Sánchez, I., Iceta, M., del Cano, L., Gragera, M., Melero, R., Sharov, G., Castaño Díez, D., Koster, A., Piccirillo, J., Vilas, J., Otón, J., Marabini, R., Sorzano, C., Carazo, J., 2022. Scipion tomo: Towards cryo-electron tomography software integration. Reprod. Valid. J. Struct. Biol. 214, 107872.

Kiewisz, R., Fabig, G., Conway, W., Johnston, J., Kostyuchenko, V.A., Bařinka, C., Clarke, O., Magaj, M., Yazdkhasti, H., Vallese, F., Lok, S.-M., Redemann, S., Müller-Reichert, T., Bepler, T., 2024. Accurate and fast segmentation of filaments and membranes in micrographs and tomograms with TARDIS. bioRxiv 629196.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment anything. arXiv, 2304.02643.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., Team, J.D., 2016. Jupyter notebooks – a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (Eds.), In:

Positioning and Power in Academic Publishing: Players, Agents and Agendas, vol. 13, pp. 87—-90.

Kremer, J.R., Mastronarde, D.N., McIntosh, J.R., 1996. Computer visualization of three-dimensional image data using IMOD. J. Struct. Biol. 116, 71–76.

Lamm, L., Righetto, R.D., Wietrzynski, W., öge, M.P., Martinez-Sanchez, A., Peng, T., Engel, B.D., 2022. Mem brain: A deep learning-aided pipeline for automated detection of membrane proteins in cryo-electron tomograms. Comput. Methods Programs Biomed. 106990.

Lamm, L., Zufferey, S., Righetto, R.D., Wietrzynski, W., Yamauchi, K.A., Burt, A., Liu, Y., Zhang, H., Martinez-Sanchez, A., Ziegler, S., Isensee, F., Schnabel, J.A., Engel, B.D., Peng, T., 2024. Mem brain v2: an end-to-end tool for the analysis of membranes in cryo-electron tomography. bioRxiv 574336.

Last, M.G., Abendstein, L., Voortman, L.M., Sharp, T.H., 2024. Streamlining segmentation of cryo-electron tomography datasets with Ais. ELife 13 (98552).

Li, X., Chang, D., Tian, T., Cao, J., 2019. Large- margin regularized softmax cross-entropy loss. IEEE Access 7, 19572–19578.

Liu, C., Zeng, X., Lin, R., Liang, X., Freyberg, Z., Xing, E., Xu, M., 2018. Deep learning based supervised semantic segmentation of electron cryo- subtomograms. In: 2018 25th IEEE International Conference on Image Processing. ICIP, pp. 1578–1582.

Martinez-Sanchez, A., Jasnin, M., Phelippeau, H., Lamm, L., 2024. Simulating the cellular context in synthetic datasets for cryo-electron tomography. IEEE Trans. Med. Imaging 43, 3742–3754.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D., 2022. Image segmentation using deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 44, 3523–3542.

Mishkin, D., Sergievskiy, N., Matas, J., 2017. Systematic evaluation of convolution neural network advances on the image net. Comput. Vis. Image Underst. Andin. 161, 11–19.

Moebel, E., Martinez-Sanchez, A., Larivière, D., Fourmentin, E., Ortiz, J., Baumeister, W., Kervrann, C., 2020. Deep learning improves macromolecules localization and identification in 3D cellular cryo- electron tomograms. bioRxiv 042747.

Papandreou, G., Chen, L.-C., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. arXiv, 1706.05587.

Qin, Z., Kim, D., Gedeon, T., 2021. Neural network classifier as mutual information evaluator. arXiv, 2106.10471.

Ridnik, T., Lawen, H., Ben-Baruch, E., Noy, A., 2022. Solving image net: a unified scheme for training any backbone to top results. arXiv, 2204.03475.

Shelhamer, E., Long, J., Darrell, T., 2014. Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 3431–3440.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., kin Wong, W., chun Woo, W., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. arXiv, 1506.04214.

Siggel, M., Jensen, R.K., Mahamid, J., Kosinski, J., 2024. Colabseg: An interactive tool for editing, processing, and visualizing membrane segmentations from cryo-ET data. J. Struct. Biol. 216, 108067.

Stimper, V., Bauer, S., Ernstorfer, R., Scholkopf, B., Xian, R., 2019. Multidimensional contrast limited adaptive histogram equalization. IEEE Access 7, 165437–165447.

Takikawa, T., Acuna, D., Jampani, V., Fidler, S., 2019. Gated- SCNN: Gated shape CNNs for semantic segmentation. arXiv, 1907.05740.

Teuwen, J., Moriakov, N., 2020. Convolutional neural networks. In: Essentials of Pattern Recognition. pp. 481–501.

Wang, J., Han, L., Ran, D., 2023. Architectures and applications of U- net in medical image segmentation: A review. In: 2023 9th International Symposium on System Security, Safety, and Reliability. ISSSR, pp. 84–94.

wwPDB Consortium, T., 2023. EMDB-the electron microscopy data bank. Nucleic Acids Res. 52, D456–D465.

Young, L., Villa, E., 2023. Bringing structure to cell biology with cryo- electron tomography. Annu. Rev. Biophys. 52, 573–595.

Zeng, X., Kahng, A., Xue, L., Mahamid, J., Chang, Y.-W., Xu, M., 2023. High-throughput cryo- ET structural pattern mining by unsupervised deep iterative subtomogram clustering. Proc. Natl. Acad. Sci. USA 120, 2213149120.

Zhou, L., Yang, C., Gao, W., Perciano, T., Davies, K.M., Sauter, N.K., 2020. Subcellular structure segmentation from cryo-electron tomograms via machine learning. bioRxiv 034025.

Zhou, L., Yang, C., Gao, W., Perciano, T., Davies, K.M., Sauter, N.K., 2023. A machine learning pipeline for membrane segmentation of cryo-electron tomograms. J. Comput. Sci. 66, 101904.

Zhou, B., Yu, H., Zeng, X., Yang, X., Zhang, J., Xu, M., 2021. One- shot learning with attention- guided segmentation in cryo- electron tomography. Front. Mol. Biosci. 7.