How Realistic are Synthetic Tomograms?

A Quantitative Embedding-Based Analysis

Joshua P. Blaser

A capstone report submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Gus L. W. Hart, Advisor

Department of Physics and Astronomy

Brigham Young University

ABSTRACT

How Realistic are Synthetic Tomograms?
A Quantitative Embedding-Based Analysis

Joshua P. Blaser
Department of Physics and Astronomy, BYU
Bachelor of Science

Cryogenic electron tomography (CryoET) is a method that biologists use for taking 3D images of bacteria on the nanoscopic scale. CryoET uses an electron microscope to capture a series of projections which are then reconstructed into 3D volumes called tomograms. Then, scientists attempt to identify and label every bacterial structure that can be found in the tomogram. In the past, the tedious task of labeling cellular complexes was solved by either hand-labeling or very slow and computationally expensive methods. Recently, the use of deep learning methods (AI) has proven to be much more efficient. However, these new adoptions of machine learning have led to the increase in demand for reliable ground truth data for the purpose of training the models. To provide such reliable ground truth data (perfectly labeled tomograms), programs are being developed to simulate synthetic tomograms. Imperfections in such simulated tomograms can cause failures in the models that they are being used to train. In order to improve the synthetic tomograms, I developed a method of evaluating how 'realistic' they are using an open-source software called TomoTwin to embed tomograms into a multidimensional latent space. The embeddings of synthetic and authentic tomograms are then compared using Maximum Mean Discrepancy (MMD). A significant difference is seen and measured between synthetic and authentic tomogram embeddings. Measuring the shortcomings of the synthetic data will give rise to specific targets of improvement in synthetic data generation and aid those developing the high-demand tomogram simulation software. Ultimately, this will allow for better trained deep learning models to extract valuable information from authentic tomograms. Doing so may help biologists to better understand the inner-workings of bacteria, fight diseases, and save lives.

Keywords: Cryogenic Electron Tomography (CryoET), Synthetic Tomograms, Tomogram Realism, Sub-tomogram Averaging (STA), Particle Picking, Machine Learning, Deep Learning, Embedding, TomoTwin, Maximum Mean Discrepancy (MMD), Distribution Comparison, Point Cloud Analysis

Contents

Chapter 1

1.1 Introduction: Why Bacteria?

Bacteria are the oldest, most numerous, and arguably the most important life forms on planet Earth. Your first thoughts might be that bacteria just live in pond water or on your dirty dishes, but they're inside you too! Bacteria are essential to digestion, they trigger rain and snow formation, and some can live miles underground and survive off of energy from radioactive rocks. Bacteria are also responsible for causing deadly diseases like salmonella, cholera, and tuberculosis.

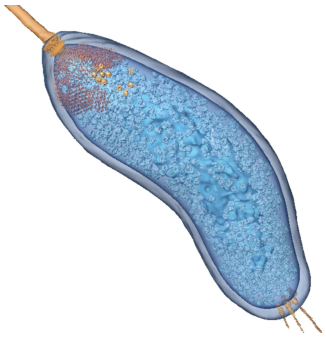


Figure 1.1 Model of Bdellovibrio bacteriovorus cell. From the Atlas of Bacterial and Archaeal Cell Structure[1]

There is no question as to why bacteria might be a field of interest among modern scientists. Understanding how bacteria work has the potential impact of curing disease, engineering drug targets, and even solving environmental problems. One species at a time, biologists are seeking to make molecular models of bacteria cells that accurately represent their functions.

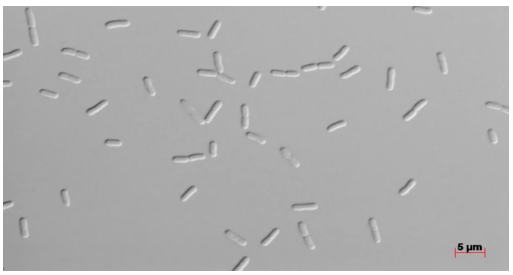
Like most organisms, bacteria are made up of various structures and 'machines', each of which performs specific functions. Biologists have developed genomic and biochemical methods that kill the bacteria and have isolated and identified which genes and proteins make up some molecular machines. However,

these methods do not reveal how the molecules are arranged or how they operate within their native environment.

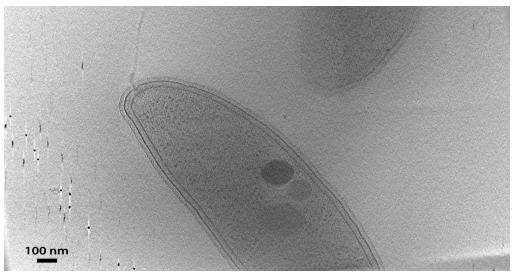
Richard Feynman, a Nobel Prize-winning physicist, is known to have said that "it is very easy to answer many of these fundamental biological questions; you just look at the thing!" That raises the question: How can we 'look at' bacteria? Bacteria are so small that not even a powerful light microscope is able to resolve important details. What can be done?

1.2 Electron Microscopy

Light microscopes use visible light to see samples, which allows users to detect details at about 200 nm resolution. Bacteria, only a few thousand nanometers in length, are only pictured as little specks through a light microscope (see Figure 1.2a). In 1931, German scientists Ernst Ruska and Max Knoll developed the world's first electron microscope[2], which took advantage of the smaller wavelength of electrons (100,000 times smaller than that of visible light) to achieve a resolution of about 0.1 nm. The invention of the electron microscope would open science to a world of new possibilities, including the detailed visualization of bacteria.



(a) Bacteria pictured with a light microscope. From CSIR- Centre for Cellular and Molecular Biology.[3]



(b) Bacteria pictured with an electron microscope. From the Atlas of Bacterial and Archaeal Cell Structure.[1]

Figure 1.2 Two images showing the difference between light and electron microscopy. Light microscopy reveals little to no details (just the relative cell shape) while electron microscopy reveals much finer details like the components and structures contained in the cell. Note the large difference in scale shown in the bottom corners of the images.

Transmission Electron Microscopy (TEM) is a type of microscopy that produces a beam of electrons and shoots them through a thin sample. The electrons that have been transmitted and scattered through the sample are then captured by an electron detector, revealing the physical details of the sample. These special microscopes require extreme precision and special components, including several magnetic electron lenses, a vacuum chamber, and an electron gun.

Cryogenic Electron Microscopy (CryoEM) is a type of TEM that is used to image bacterial cells. It requires that the sample be frozen by liquid ethane to create a layer of vitreous ice. This preserves the cells' natural state, allowing biologists to study and observe them in a near-native, undamaged, and hydrated condition.[4]

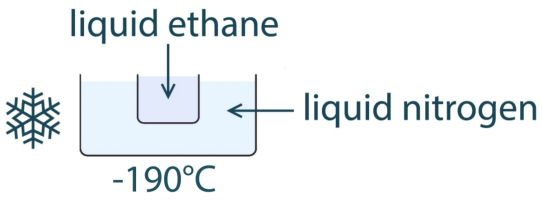


Figure 1.3 Tank of liquid nitrogen and ethane in which the bacteria is submerged to create a flash frozen sample of a cell in its natural environment. Used in CryoEM. From the Atlas of Bacterial and Archaeal Cell Structure [1]

The frozen sample is then placed inside of the electron microscope. Plunge-freezing fixes the specimen in place "fast enough to cause minimal changes to atomic positioning"(Wikipedia). Electron beams are shot through the sample, the sample is tilted, and a beam is shot again. This process is repeated until a group of 2D projections (maps of electron scattering density from a 3D object) called a *tilt series* have been collected. These tilt series projections are then reconstructed to form a three-dimensional volume called a tomogram (See Figure 1.4). Collecting tomograms in this manner is called Cryogenic Electron Tomography (CryoET).

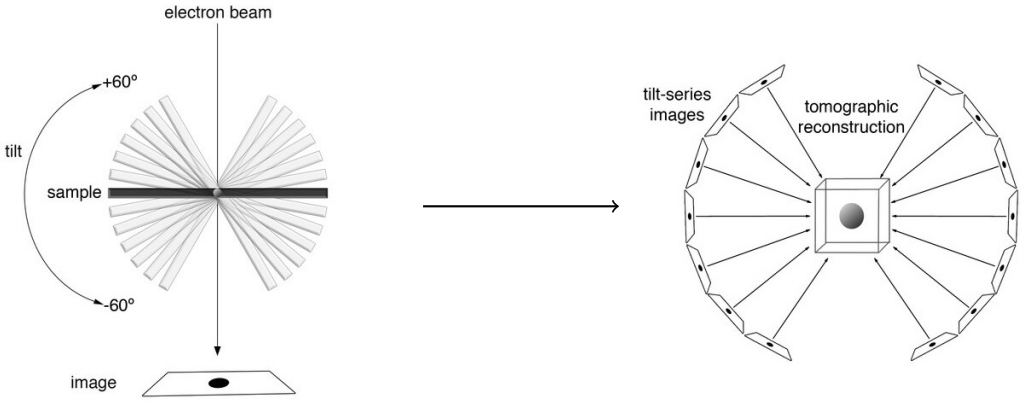
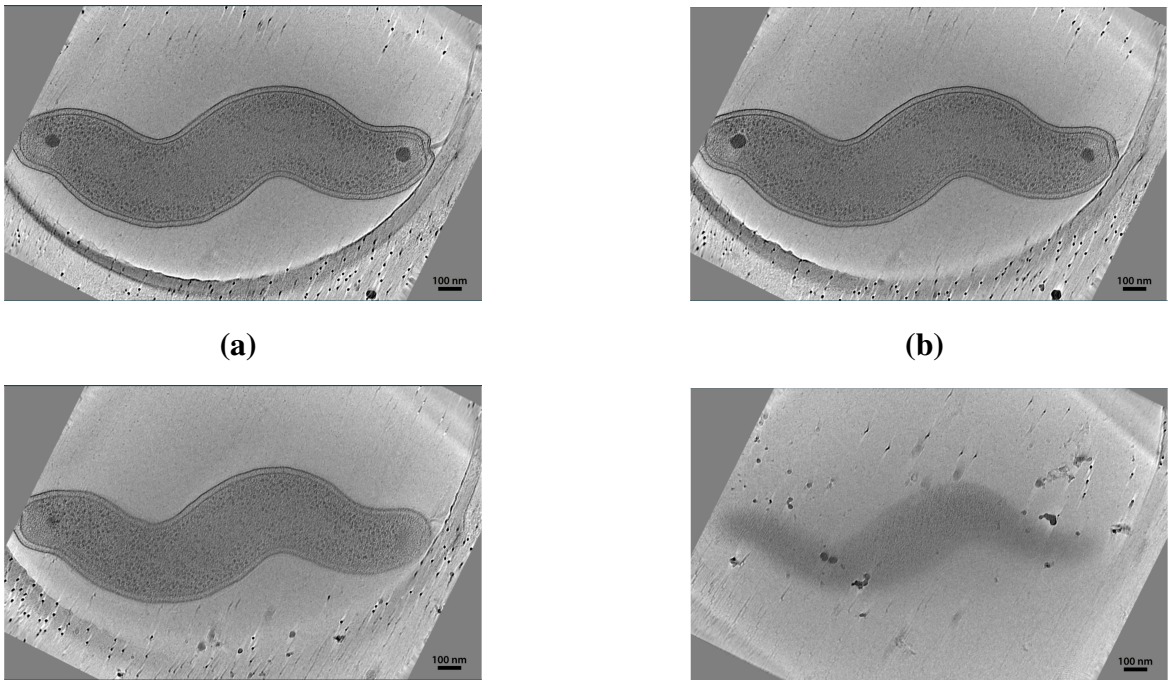


Figure 1.4 Tilt-series acquisition (left) and reconstruction into a tomogram (right). From Eikosi.



(a) **(b)**

(c) **(d)**

Figure 1.5 2D slices from a 3D tomogram. Images are seen moving through the z-axis of the tomogram. From the Atlas of Bacterial and Archaeal Cell Structure.

1.3 Inside the Tomogram

Figure 1.5 depicts several 2D slices of a tomogram captured using Cryogenic Electron Microscopy. While some tomograms clearly reveal the details of bacterial structures, most contain substantial amounts of noise (what looks like TV static in figure 1.6) and not a lot of signal (the important information). Essentially, high levels of noise make the details of the bacteria difficult to discern.

The whole goal with taking the 3D picture of bacteria is to develop detailed atomic models of the different nanostructures located within a given cell. In order to do this, scientists must identify and isolate specific instances of each cellular structure found in the image. This process is called *particle picking*. However, since the signal to noise ratio (SNR) is so low, there is usually not enough detail to build a model of a specific nanomachine from just one tomogram.

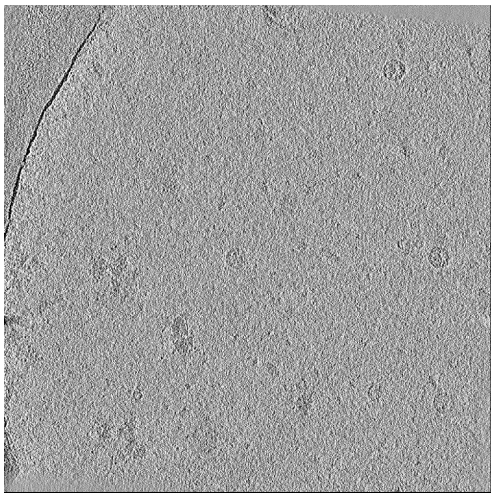


Figure 1.6 This figure serves as a demonstration of the low Signal to Noise Ratio (SNR) common in tomograms.

The low SNR has led scientists to develop a strategy called Sub-tomogram averaging (STA). This involves finding hundreds of instances of the same object and computationally combining their images to increase the resolution (See Figure 1.7). From this now detailed image of a bacterial nanostructure, biologists are able to develop an atomic model that accurately describes the makeup and functions of that specific complex. See Figure 1.8 for an example of the atomic model of the flagellar motor.

These models can then be used to run simulations and discover the functionalities and movement mechanisms of each part of the cell. In the case of the flagellar motor, the atomic model can be used to discover the mechanics of how it spins.

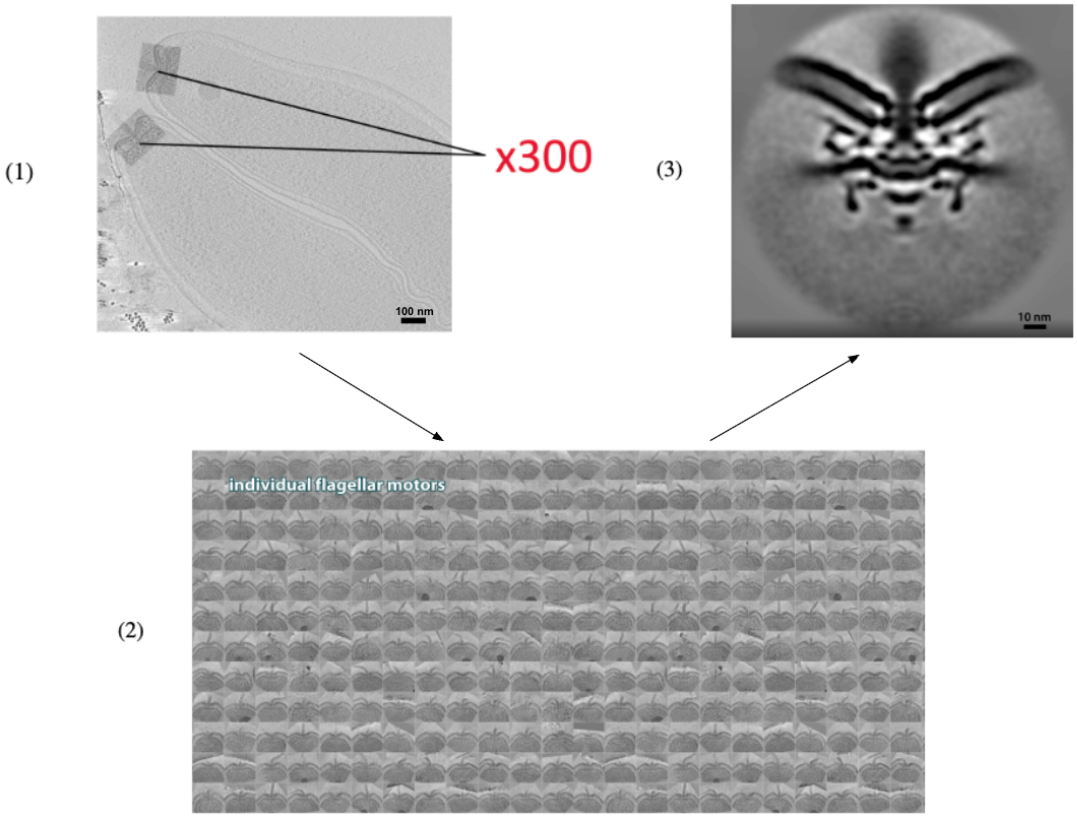


Figure 1.7 Objects are identified and extracted from tomograms (1). This process is repeated hundreds of times to build a large dataset (2). Averaging these particles reveals detailed structural information, a process known as Sub Tomogram Averaging (STA) (3). Note the change in scale. Averaging leads to a large increase in resolution. From The Atlas of Bacterial and Archaeal Cell Structure

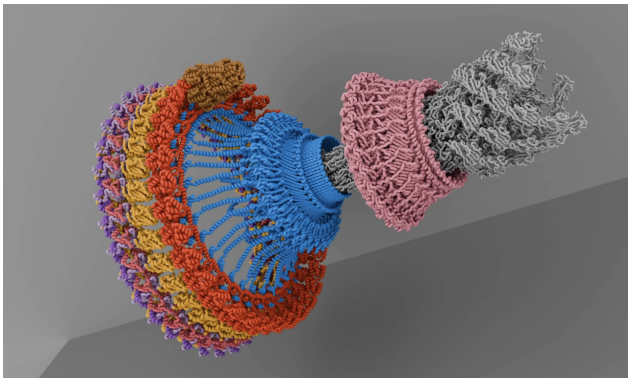


Figure 1.8 An atomic model of a bacterial Flagellar Motor from SmarterEveryDay[5]

In order for sub-tomogram averaging to work, tomograms must be collected and then hundreds of the same object must be identified and located in each tomogram. This particle-picking process is a major bottleneck in CryoET. As a result, researchers are exploring methods to automate this tedious task.

1.4 Machine Learning

In recent years, machine learning and artificial intelligence strategies have made their way into just about every field of business and science. Complex algorithms called models are trained to complete tasks that normally only humans can do. Training the models *requires* a plethora of 'training data' that must first be accurately labeled by humans.

Machine learning has recently shown successes in fixing the particle picking problem.[6] The models are able to identify and locate different bacterial structures, allowing for the hundreds of images needed for STA to be collected quickly.

1.5 Synthetic Tomograms

With machine learning having proved its ability to solve the particle picking bottleneck, many scientists are working now on improving these models by creating more training data. Training requires tomograms that have accurately labeled nanomolecular complexes. Rather than relying on humans' ability to find tiny structures inside noisy images of the cell, image scientists have started developing programs that can make computer-generated tomograms.

These simulators generate tomograms that are not real but accurately reflect the cellular context found in CryoET tomograms. In just the last four years, three different synthetic tomogram simulators have been created to aid the machine learning training process.[7, 8, 9] There exists few techniques to evaluate just how well the synthetic tomograms simulate the cellular environment

contained in authentic tomograms. An interesting question can be posed: Can we measure the realism gap between synthetic and authentic tomograms?

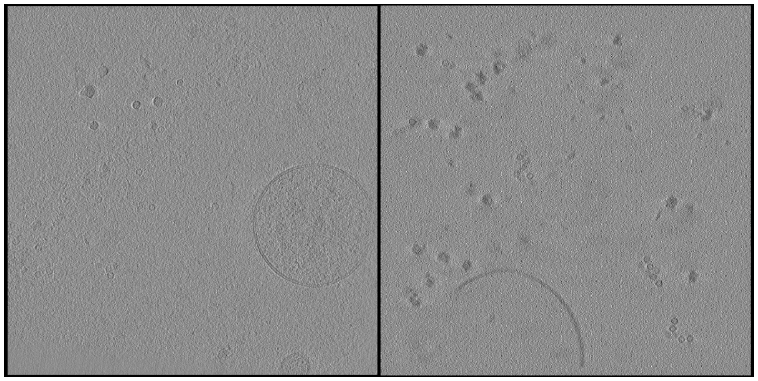


Figure 1.9 A side by side look at synthetic and authentic tomograms used in the experiment. Particle types seen here include apo-ferritin, ribosomes, and virus-like-particles as dictated by the CZII Kaggle competition.[10] (Visualized using Napari.)

1.6 Getting the Data

In order to answer the question: 'Can the gap between real and synthetic be measured?' the proper data must be obtained. To accurately compare synthetic and real tomograms, both of them should be tomograms that depict the same species of bacterial complexes. Luckily, such sets of tomograms already exist. For a machine learning competition hosted from November 2024 to February 2025 by the Chan-Zuckerberg initiative, the Chan-Zuckerberg Imaging Institute (CZII) created both a phantom and synthetic dataset.[11] The dataset included 27 computer-generated synthetic tomograms and 28 tomograms that were taken with an electron microscope. Both sets of tomograms contain the same six protein complexes. The similar nature of these tomograms makes them a great candidate for testing the hypothesis. (Phantom refers to biological samples that are specially engineered to contain specific desired protein complexes.)

1.7 Hypothesis

The proposed hypothesis:

While detecting the differences between real and synthetic tomograms is not a trivial task that can be accomplished through human observation (see Figure 1.9), by *embedding* each respective set of tomograms into an embedding space built for CryoET, a significant difference between synthetic and authentic tomograms will be measured.

1.8 Embedding: What is it?

For those unfamiliar with the basics of machine learning, I will give an explanation of what embedding is. The *New Oxford American Dictionary* definition of embed is to "implant (an idea or feeling) within something else so it becomes an ingrained or essential characteristic of it." In the context of machine learning, embedding is representing the meaning of something as a list of numbers, or a vector. The principle behind embedding is to take large amounts of data (such as 3D or text) and express the information from that data in a *lower-dimensional* format. This process reduces the size of data, making it more computationally efficient, while still retaining key information for further analysis and manipulation. Specifically, these lower-dimensional representations are often smaller than the original data but still capture the most essential features.

What does 'lower dimensional' mean in this context? Dimensionality refers to the number of features or variables that make up a data point. In a 256×256 RGB photo, for example, there are

$$256 \times 256 \times 3 = 196608$$

different feature points (one for each pixel's RGB color value). Embedding techniques reduce the vast number of features into a smaller and more manageable number—often by creating a vector with hundreds, or even fewer, numbers. So, 'lower dimensional' refers to reducing the number of features being used to express the original data.

1.8.1 Embedding: A comparison to Fourier analysis

Fourier analysis is a helpful comparison for understanding embedding. In such a process, a signal is represented with its frequency components. Specifically, the signal is decomposed into a sum of sinusoidal waves at different frequencies, amplitudes, and phases. This shifts the representation from the time or spatial domain into the frequency domain (See Figure 1.10) and reveals new patterns. In other words, a complex signal is changed into something much simpler, although no information is lost. Embedding is a similar idea; some sort of signal or data is represented in a simpler numerical way, while virtually all of the meaning is preserved.

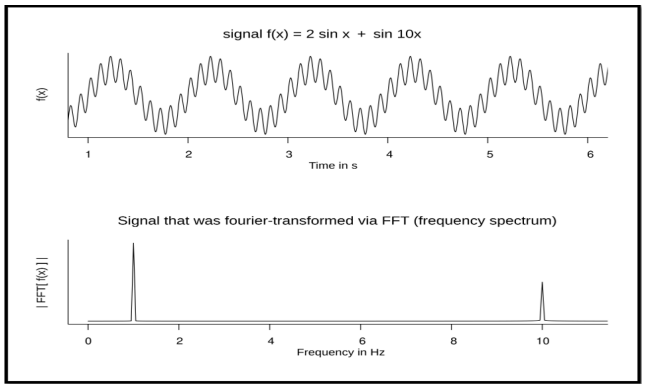


Figure 1.10 Visual representation of a signal depicted in spatial and Fourier space. From Wikimedia Commons.[12]

1.8.2 Example 1: Embedding Words

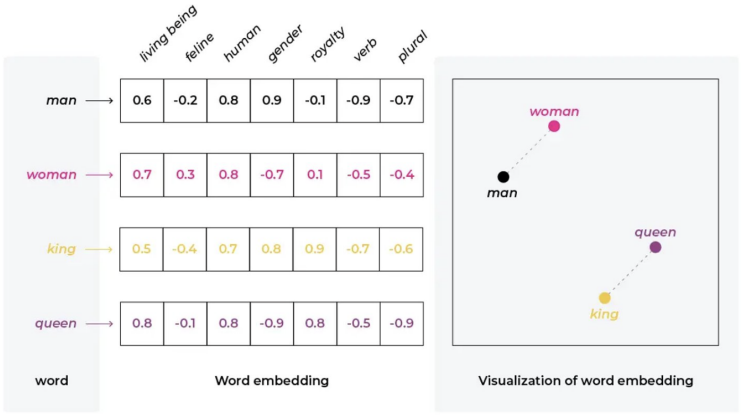


Figure 1.11 From arize.com[13]

Figure 1.11 depicts the idea of embedding words. This is the basics of how ChatGPT and other large language models work. Each word is made into a vector, and the numbers of the vector are like a score for how much the meaning of the word fits into that category. The 6 numbers contain the meaning of the word, just in a different format. Notice the different positive and negative values under the gender category—a more negative value corresponds to being more female and a more positive value is more male. A quick look at the human category shows similar numbers for all four words, which is to be expected. The graphic representation of the word embeddings helps viewers visually see the relation between different words. The distance between different words on the graph relates to a difference in the semantic meaning of the word.

1.8.3 Example 2: Embedding an Image

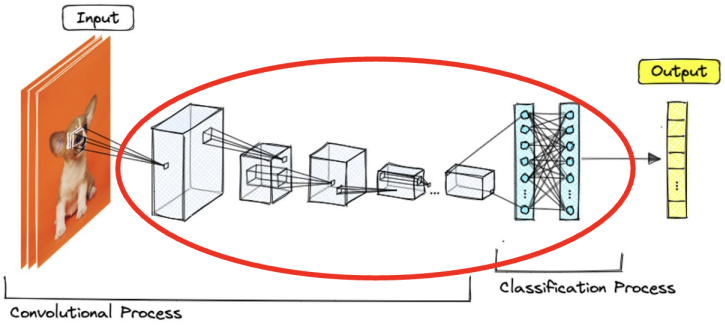


Figure 1.12 An example of a 2D image being embedded into a vector. From Pinecone.io.[14]

Figure 1.12 gives an example of what the embedding process might look like. Just as the words in the previous example were changed into simple vectors, this image of a dog is also changed into a vector. The middle section (circled in red) is where the magic happens, where the meaning of a picture of a dog is turned into a simple vector. So the question arises: How can one change a 256×256 RGB image of a dog into a much smaller vector? The answer is simply: A combination of high dimensional linear-algebra, complicated derivatives, and a whole lot of iterations.

This is where the name machine *learning* comes from. Just as humans learn to recognize a dog through repetition and interpretation, the computer learns its own way to recognize a dog through repetition, and embedding!

1.9 TomoTwin: A tomogram embedding software

In 2023, Thorsten Wagner, Gavin Rice, Yousef Metwally and Kyle Harrington published an open source software called TomoTwin[15]. TomoTwin is a specially trained deep learning model and pipeline that embeds CryoET tomograms and provides a strategy to annotate them quickly. By using TomoTwin, scientists are able to take tomograms that contain 580 million data points and embed it into a representation 2000 times smaller. With tomograms being so large and complex, reducing the size of the data is a necessary step for computational analysis. Synthetic and authentic tomograms must be simplified before they can be compared. TomoTwin completes this step in a creative way.

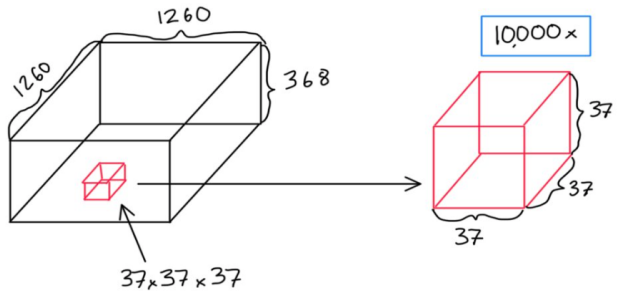


Figure 1.13 A drawing of the sub-tomogram chunking process that TomoTwin uses to prep tomograms for embedding.

We learned in the example of embedding the image of a dog that an image is typically embedded into a vector. In the case of tomograms, which are essentially 3D images, it is not possible to reasonably represent the whole volume in just one regularly-sized vector. For this reason, each tomogram that is fed into TomoTwin's model is split up into smaller cubes called sub-tomograms.

To ensure full representation of the tomogram's structure, adjacent subvolumes overlap, which helps maintain the integrity of the original data when generating the embedding. These are typically 37×37×37 voxels in size. Each of these sub-tomograms is then sent into the model, which embeds the meaning into a single vector of 32 elements (See Figure 1.14).

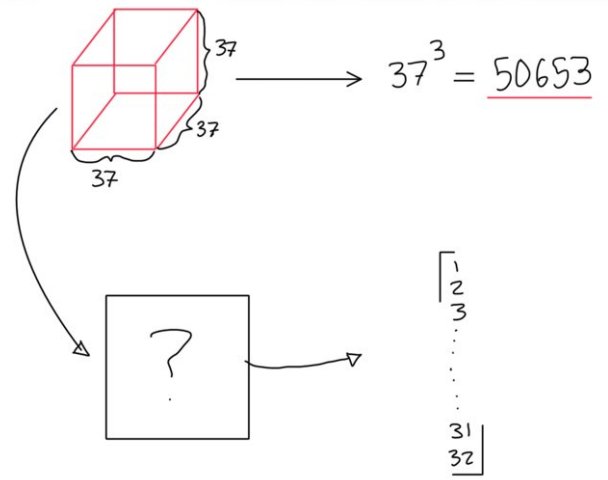


Figure 1.14 Each sub-tomogram goes through the TomoTwin model (the box with '?') and is turned into a 32 element vector.

Since each tomogram has been split into tens of thousands of sub-tomograms, the TomoTwin embedding process yields an equal number of 32 element vectors. Each of the vectors captures the essential features of the sub-tomogram, transforming it into a point in a 32-dimensional vector space. This means that each point represents a small but meaningful section of the original tomogram, preserving its structural information in a more compact, numerical form.

However, visualizing this high-dimensional space directly is challenging due to the sheer number of dimensions (32) and the number of vectors (tens of thousands). Dimensionality reduction techniques are employed to map these embeddings into a dimension that can be visualized. Specifically, UMAP (Uniform Manifold Approximation and Projection)[16], a popular dimensionality reduction algorithm is used to transform the 32-dimension embeddings into a 2-dimensional plot (See figure 1.15). UMAP is particularly well-suited for this task because it preserves both local and global structures within the data, making it ideal for visualizing complex point clouds of embeddings. Note that only the original 32 dimensional point clouds are used in the comparison steps of this experiment, the UMAP projections are just used as visualizations.

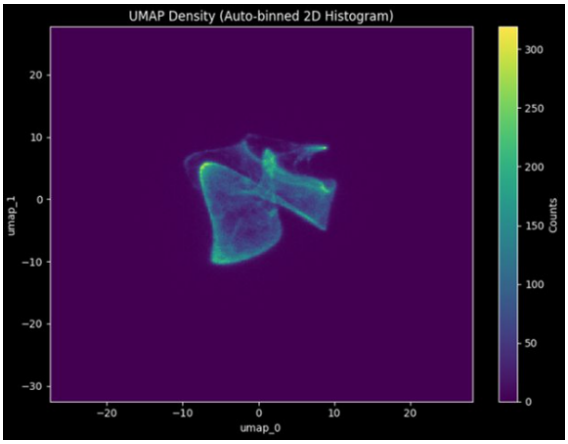


Figure 1.15 A 2D representation of the point cloud of embeddings. This cloud of points contains the meaning and essential characteristics of the original tomogram. Density of points is represented by color on the histogram scale.

In summary, TomoTwin embeds tomogram sub-volumes so that they can be represented as point clouds in a 32-dimensional embedding space. These point clouds can then be visualized using UMAP. The reduction in dimensionality from the original high-dimensional tomograms to a 32-dimensional space enables more efficient analysis and comparison. Now that we understand the inner workings of TomoTwin's embedding process, let's jump into the details of how it was used to compare the synthetic tomograms with their authentic counterparts.

1.10 Setting up for tomogram comparison

In order to use TomoTwin software on the tomograms obtained from the CZII CryoET Data Portal[17], specific resources need to be obtained and specific configuration processes need to be followed. As mentioned earlier, sets of 27 synthetic (simulated) tomograms and 27 authentic (phantom) tomograms containing the same protein complexes were gathered and downloaded to perform the experiment.

Following the documentation on TomoTwin's website, the software environment was set up with the necessary packages and installs to run the TomoTwin pipeline. This included the visualization program Napari[18], as well as the specially trained sub-tomogram embedding model[19]. In addition to these setup procedures, it was necessary to run and configure TomoTwin on a remote supercomputer with access to NVIDIA Graphics Processing Units (GPUs). Special thanks to Brigham Young University's research computing center for their resources and assistance.

Once TomoTwin was properly configured and the necessary resources obtained, each simulated and phantom tomogram was fed through the pipeline and embedded into a point cloud representation (Figure 1.16). Each tomogram took about 30 minutes to embed while running the TomoTwin embedding pipeline using h200 GPUs and Strategy 3 from the documentation. When all tomograms were embedded, a collection of the embeddings were split into the two groups of simulated and phantom tomograms.

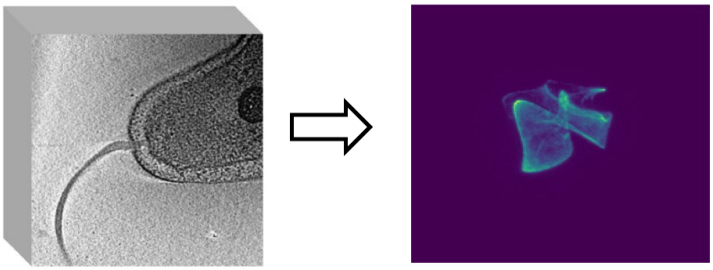


Figure 1.16 The embedding process of TomoTwin

1.11 Comparing the tomograms

Now that the synthetic and authentic tomograms have all been embedded into point cloud representations, they are much easier to work with. Each embedding cloud contains the information and meaning of each tomogram but in a smaller format. In order to discover the similarities and differ-

ences across these point cloud distributions, a statistical metric called Maximum Mean Discrepancy was calculated.

1.11.1 What is Maximum Mean Discrepancy (MMD)?

Maximum Mean Discrepancy is a kernel-based statistical metric that measures the distance between distributions and has "numerous applications in machine learning"[20]. MMD was invented by Arthur Gretton of UCL in 2012[21]. The strategies of MMD creatively invoke properties of the reproducing kernel Hilbert space (RKHS) to calculate the distance between two distributions of data. Essentially, the RKHS is just a function that takes in data points x and y and tells you how similar they are.

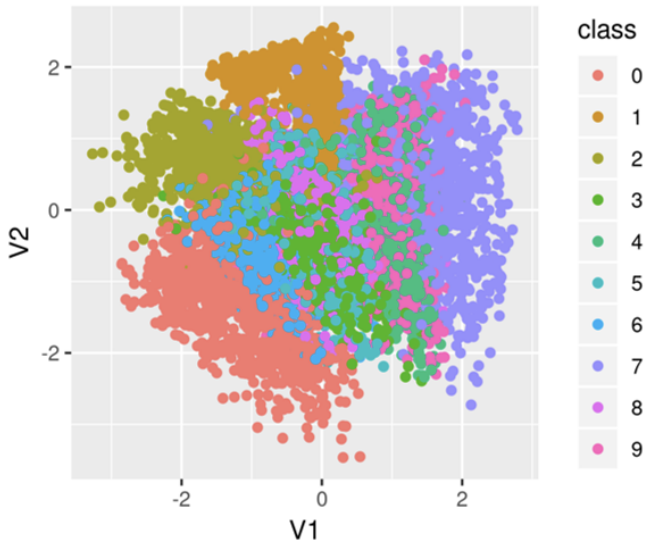


Figure 1.17 From Representation learning with MMD-VAE[22]

In Figure 1.17, the MMD of the orange-red distributions will be much larger than that of green-blue distributions. Similarly, by using the method of MMD, we are able to find exactly how similar the synthetic tomograms are to their authentic counterparts. The results of such calculation

quantify the similarity between synthetic and authentic tomograms, providing a repeatable method for assessing simulation accuracy.

1.12 Results

To act as a control, the MMD^2 distance was calculated both between each phantom (authentic) tomogram and the rest of the entire distribution of phantom tomograms, and between the simulated (synthetic) tomograms and their accompanying distribution. Then, the MMD^2 distance was calculated between each phantom tomogram and the distribution of simulated tomograms, and each simulated tomogram against the distribution of phantom tomograms. These calculations are plotted in Figure 1.18.

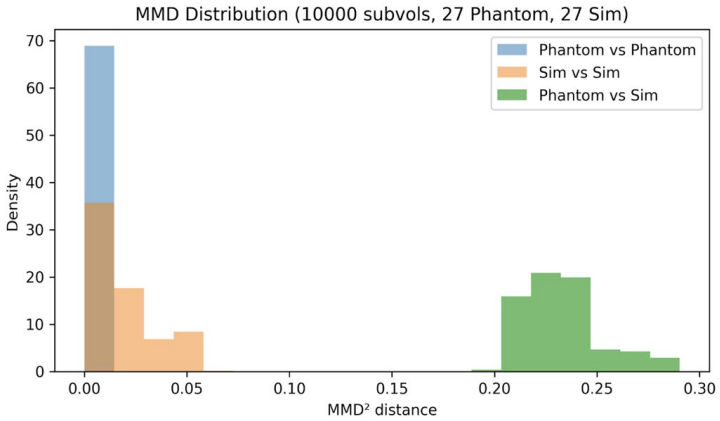


Figure 1.18 The MMD^2 distances of Phantom-Phantom, Sim-Sim, and phantom-Sim tomogram embedding distributions.

The above figure reveals a much larger difference between the phantom and simulated tomogram embeddings than the control comparisons. The mean values from the plot can be seen in the table below.

It is important to note that the Phantom-Sim mean is 500 times larger than the Phantom-Phantom mean. This indicates a substantial difference between the two tomogram types.

Distribution Pair	MMD^2 mean
Phantom-Phantom	0.00047
Sim-Sim	0.018
Phantom-Sim	0.23

1.12.1 Application of Results

The pipeline developed for measuring the difference between synthetic and authentic tomograms offers a reproducible way to quantitatively assess the realistic nature of simulated tomograms. By providing specific and objective comparison, the pipeline ensures that improvements made in synthetic tomogram generation are based on measureable criteria. The metrics obtained from this analysis are valuable for developers of synthetic tomogram simulators, offering concrete insights into how simulations may be falling short.

Not only will this pipeline yield improvements in the fidelity of synthetic tomograms, but it will also have benefits further down the CryoET pipeline. As synthetic tomogram generation improves, it will enhance the accuracy of machine learning models that rely on high-quality tomographic data. Doing so will allow for faster particle picking, better sub-tomogram averaging, and improved visualization of bacterial structures. Such visualizations will increase the pace at which the atomic models of bacterial structures crucial for understanding biological processes are developed.

Looking at broader implications, the enhancement of atomic models of bacterial structures opens the door to novel biological discoveries. Researchers can potentially harness the natural capabilities of bacteria for medical and industrial applications. Ultimately, this could lead to breakthroughs in areas like drug delivery, environmental cleanup, and disease prevention, offering profound impacts on human health and well-being.

Bibliography

[1] Cell Structure Atlas, Cell structure atlas (2024).

[2] American Association for the Advancement of Science, Development of the electron microscope, https://www.aaas.org/membership/scientia/development-electron-microscope (2024).

[3] YouTube, Bacteria under light microscope (2024).

[4] J. Dubochet *et al.*, Quarterly Reviews of Biophysics **21**, 129 (1988).

[5] SmarterEveryDay, Flagellar motor visualization (2024).

[6] Kaggle, Byu locating bacterial flagellar motors 2025, https://www.kaggle.com/competitions/byu-locating-bacterial-flagellar-motors-2025/leaderboard (2025), accessed: 2026-04-11.

[7] P. Harar, L. Herrmann, P. Grohs, and D. Haselbach, Structure **33**, 820 (2025).

[8] C. Purnell, J. Heebner, M. T. Swulius, R. Hylton, S. Kabonick, M. Grillo, S. Grigoryev, F. Heberle, M. N. Waxham, and M. T. Swulius, bioRxiv 10.1101/2023.04.28.538636 (2023), https://www.biorxiv.org/content/early/2023/04/28/2023.04.28.538636.full.pdf .

[9] A. Martinez-Sanchez, L. Lamm, M. Jasnin, and H. Phelippeau, bioRxiv 10.1101/2023.05.26.542411 (2023).

[10] Kaggle, Czi cryo-et object identification, https://www.kaggle.com/competitions/czii-cryo-et-object-identification/data (2023), accessed: 2026-04-11.

[11] A. Peck, Y. Yu, J. Schwartz, A. Cheng, U. H. Ermel, J. Hutchings, S. Kandel, D. Kimanius, E. A. Montabana, D. Serwas, *et al.*, Nature Methods **22**, 1819 (2025).

[12] Wikimedia Commons, Time domain vs frequency domain visualization (2024).

[13] Arize AI, Embeddings: Meaning, examples, and how to compute (2023).

[14] Pinecone Systems, Image embeddings explained (2023).

[15] T. Wagner, G. Rice, Y. Metwally, and K. Harrington, bioRxiv 10.1101/2023.04.01.535894 (2023).

[16] L. McInnes, J. Healy, and J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction (2020), arXiv:1802.03426 [stat.ML] .

[17] Chan Zuckerberg Imaging Institute, Cryoet data portal (2025).

[18] Napari Developers, Napari: Multi-dimensional image viewer for python (2024).

[19] T. Wagner, Tomotwin pretrained model (2023).

[20] S. Liu *et al.*, in *Advances in Neural Information Processing Systems (NeurIPS)* (2016).

[21] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, Journal of Machine Learning Research **13**, 723 (2012).

[22] RStudio AI Blog, Representation learning with mmd-vae (2018).