Supplementary Document: Image-derived generative modeling of pseudo-macromolecular structures – towards the statistical assessment of Electron CryoTomography template matching

Kai Wen Wang[1]
kaiwenw1@andrew.cmu.edu
Xiangrui Zeng[1]
xiangruz@andrew.cmu.edu
Xiaodan Liang[1]
xiaodan1@andrew.cmu.edu
Zhiguang Huo[2]
zhuo@ufl.edu
Eric P. Xing[1]
epxing@cs.cmu.edu
Min Xu (Corresponding author)[1]
mxu1@cs.cmu.edu

[1] Carnegie Mellon University
Pittsburgh, PA, USA
[2] University of Florida
Gainesville, FL, USA

1 Supplementary Materials

1.1 Implementation details

A modified version of the Tomominer package [7] was used for processing the images. EMAN2 was used for back projection reconstruction. Mayavi was used for plotting the iso-surfaces. Keras and Tensorflow were used for constructing and training the 3D-WGAN. The test is performed on a computer equipped with Intel i7 CPU, 128GB memory, and Nvidia GTX 1080 and GTX 1080 Ti GPUs.

1.2 Generating simulated subtomograms

For our experiments on simulated subtomograms to be as reliable as possible, we simulated subtomograms by performing the actual tomographic image reconstruction process in a similar way as previous works [3, 5, 16, 23]. We properly included noise, and the missing wedge effect, and electron optical factors, such as the Contrast Transfer Function (CTF) and

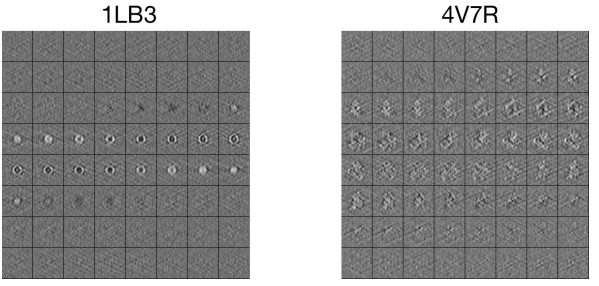

Figure 1: Raw slices of simulated subtomograms constructed from Mouse L-chain ferritin (PDB ID: 1LB3) and ribosome (PDB ID: 4V7R), with SNR of 0.03.

Modulation Transfer Function (MTF), assuming that the electron optical density of macromolecular complexes is proportional to the electrostatic potential. We used the PDB2VOL program from the Situs [21] package to generate volumes of 64^3 voxels with a resolution and voxel spacing of 0.6nm. The density maps of known structure were used to simulate electron micrograph images using the tilt-angle of $\pm 60^\circ$. We added noise to electron micrograph images [5] to achieve the desired SNR of 0.03, which was the lowest SNR for the fast alignment method to achieve successful alignment on all simulated subtomograms. Next, we convolved the electron micrograph images with the CTF and MTF to simulate optical effects [6, 16]. The acquisition parameters used are typical of those found in experimental tomograms [24], with spherical aberration of 2mm, defocus of -5μm, and voltage of 300kV. The MTF is defined as $\mathrm{sinc}(\pi\omega/2)$ where ω is the fraction of the Nyquist frequency, which corresponds to a realistic detector [15]. Finally, a direct Fourier inversion reconstruction algorithm (implemented in the EMAN2 library [8]) is used to produce the simulated subtomogram according to the tilt-angle. Figure 1 shows examples of such simulated subtomograms.

The templates used for template matching were constructed using a tilt-angle range of ± 90 degrees and infinite SNR, and with the same acquisition parameters used to simulate the subtomograms. The low SNR and missing wedge values make it very difficult to properly align subtomograms, especially smaller ones such as Ornithine carbamoyltransferase (PDB ID: 1A1S), but we found that including a Gaussian blur of 0.2nm increased the success rate of our fast alignment method.

1.3 Table of PDB ID

We collected 15 macromolecular complexes from the Protein Databank (PDB) [4], shown in the following table.

PDB ID	Macromolecular Complex
1A1S	Ornithine carbamoyltransferase
1BXR	Carbamoyl phosphate synthetase
1EQR	Aspartyl tRNA-synthetase
1KYI	HslUV (H. influenzae)-NLVS Vinyl Sulfone Inhibitor Complex
1LB3	Mouse L-chain ferritin
1VPX	Transaldolase
1VRG	Propionyl-CoA carboxylase, beta subunit
1W6T	Octameric enolase
1YG6	ClpP
2BYU	Small heat shock protein Acr1
2GLS	Glutamine synthetase
2H12	Acetobacter aceti citrate synthase
2IDB	3-octaprenyl-4-hydroxybenzoate decarboxylase
3DY4	Yeast 20S proteasome
4V7R	Yeast 80S ribosome

Table 1: The experimental macromolecular complexes used in this paper. They were used as training data for the 3D-WGAN, as complexes used to construct real templates and as complexes for simulation of subtomograms.

1.4 Details of the 3D-WGAN

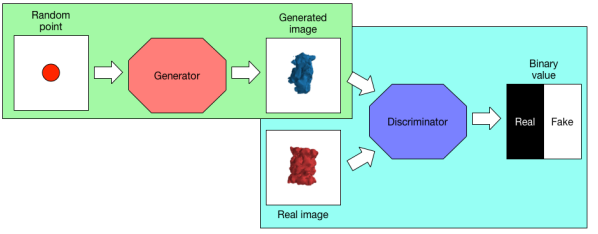

Figure 2: This diagram shows the concept of GAN.

Our generative approach uses techniques from deep learning, specifically the convolutional neural network and the GAN. Illustrated by Figure 2, the GAN involves a generator G and a discriminator D. When given a sample of random points z. the goal of G is to output its best attempt of sampling an image from the distribution on the manifold of training images. The discriminator model D is trained to label generated (pseudo) images as 1 and training (real) images as 0. During training, we optimize D to maximize its ability to classify real and pseudo images correctly, and we optimize G to minimize the discrimination ability of D. This cost can be described by the binary cross-entropy equation [9, 22]

$$\min_G \max_D \; \mathbb{E}_{x\sim\mathbb{P}_{\text{data}}(x)} \log(D(x)) + \mathbb{E}_{z\sim\mathbb{P}_{\text{latent}}(z)} \log(1 - D(G(z))). \tag{1}$$

In our case, $x \sim \mathbb{P}_{\text{data}}(x)$ represents sampling from training data of real structures, and $z \sim \mathbb{P}_{\text{latent}}(z)$ represents sampling from the standard multidimensional Gaussian distribution in the latent space.

For our model, we used an improved and stabilized version of the GAN, called the Improved Wasserstein GAN (WGAN), which instead optimizes the Wasserstein distance, a cost function that is more favorable for optimization [1, 10]. Given two distributions $\mathbb{P}_1$ and $\mathbb{P}_2$, the Wasserstein distance can be calculated as

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{||f||_L \leq 1} \; \mathbb{E}_{x\sim\mathbb{P}_1}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_2}[f(x)], \tag{2}$$

according to [20], where the supremum is over all the 1-Lipschitz functions $f : \chi \to \mathbb{R}$. The Wasserstein distance describes the “distance" between two probability distributions. A benefit of using the Wasserstein distance is that it is continuous everywhere and almost differentiable everywhere. This allows us to train D until convergence. As D becomes more accurate, G can learn to generate more realistic images. G and D continue to play the minimax game until they converge near to an optimum state, at which point the G would produce diverse and realistic samples from the training image distribution.

The generator G and the discriminator D are implemented as convolutional neural networks (CNN). The CNN is a stack of convolution layers that can extract a complex hierarchy of image features, which has shown very successful results in computer vision applications such as object recognition and classification for both 2D and 3D images[13, 14]. [22] was the first attempt of using 3D CNNs with GANs and we based our network architecture around it. Although [22] uses five convolution layers, we found that four layers were enough to capture the structure of macromolecular complexes. In our 3D-WGAN generative model, we

also use the transposed convolution layer [22] instead of the up-sampling layers to produce larger-size images. Each convolution layer is a collection of learned feature extractors [2]. Given N filters of size K^3 and stride S, the output y is the result of "sliding" each filter in steps of S, on top of the input and summing up the dot product at every location. Let the input to the layer be X with D volumetric slices, the filters of the layer be W and the output be y. Mathematically [19], the convolutional layer is represented by

$$y^m_{i,j,k} = \sum_{p=0}^{D-1}\sum_{a=0}^{K-1}\sum_{b=0}^{K-1}\sum_{c=0}^{K-1} W^m_{a,b,c} \cdot X^p_{a+S\cdot i,b+S\cdot j,c+S\cdot k} \tag{3}$$

, where, $y^m_{i,j,k}$ is the index (i,j,k) of the m^{th} output volumetric slice , $W^m_{a,b,c}$ is the index (a,b,c) of the m^{th} filter, and $X^p_{a+S\cdot i,b+S\cdot j,c+S\cdot k}$ is the index $(a+S\cdot i, b+S\cdot j, c+S\cdot k)$ of the p^{th} input volumetric slice. For example, a filter can learn to be like an edge extractor by learning weights similar to a Sobel kernel.

The Batch Normalization layer guarantees 0 mean and unit variance for inputs to hidden layers, which has been shown to stabilize training of deep networks [12]. Given a batch of inputs x, the batch normalized output is

$$\text{BN}(x) = \gamma \cdot \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \varepsilon}} + \beta \tag{4}$$

, where γ and β are learned parameters, ε is a small constant to prevent division by 0, and $\mathbb{E}[x]$ and $\text{Var}[x]$ are the mean and variance of x respectively.

The activation functions used in our model are LeakyReLU and the hyperbolic tangent tanh. The LeakyReLU activation with a gradient of m is described by Equation 5. The purpose of the Rectified Linear Unit (ReLU) and its variants (e.g. LeakyReLU) is to introduce non-linearity into the network, and have become widely used for its ability to speedup the convergence of deep networks [11, 13]. The hyperbolic tangent is described by Equation 6, and was used to bound the output of the generator within the range [-1, 1], as suggested by [17].

$$\text{LeakyReLU}(x) = \max(mx, x) \tag{5}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

A common problem in GANs that we also experienced was mode collapse. This collapse of the generator occurs when the generator produces very uniform and non-diverse structures. The minibatch discrimination layer was proposed by [18] to reduce the collapse of the generator by penalizing low-entropy generators. Intuitively, this layer allows the discriminator to observe many samples at once, so that it can also take entropy of a batch of samples into account when deciding between real and pseudo images. Following the method proposed in [18], introducing such a measure of "(low) entropy" is as follows. For a minibatch discrimination layer, we introduce a trainable tensor T, and let $f(X_i)$ be some features extracted from the sample X_i (i.e. output of some intermediate layer). For every $f(X_i)$ s.t. $0 \leq i < n$, compute the matrix $M_i = f(X_i)T$. Now, for every row b of M_i, compare it with every M_j by calculating

$$MB_b(X_i) = \sum_{j=0}^{n} \exp(-|M_{i,b} - M_{j,b}|) \tag{7}$$

, and $MB(X_i) = [MB_1(X_i), MB_2(X_i), ..., MB_b(X_i)]$. These outputs define a measure of "(low) entropy". We concatenate $MB(X) = [MB(X_1), MB(X_2), ..., MB(X_n)]$ to the output of some intermediate layer, so that the discriminator can take this measure of entropy into account.

1.5 Cases of failed simulated hypothesis tests

In our simulated tests, on average, 6.67% of our tests resulted in Cond. 1 failure and 10.67% of our tests resulted in Cond. 2 failure.

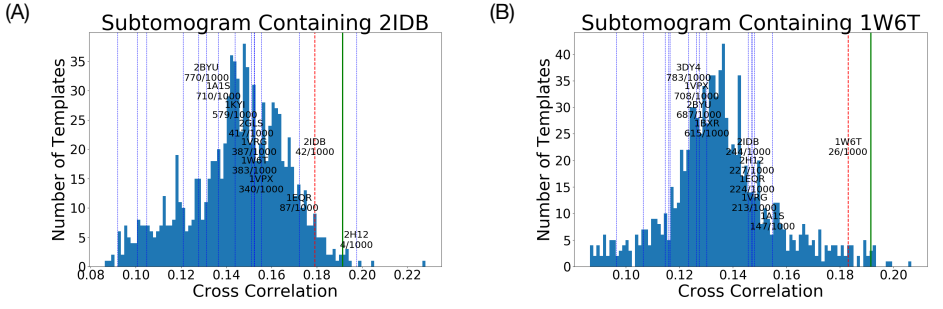

Figure 3: Examples of unsuccessful results. (A) is a Cond. 1 failure likely caused by bias in alignment due to low SNR; (B) is a Cond. 2 failure as it has a p-value above 0.01.

1.6 Supplementary Figures

1.6.1 Simulated Results

The following two pages are the remainder of the hypothesis tests on simulated subtomograms. All of these are successful with the exception of 1VPX.

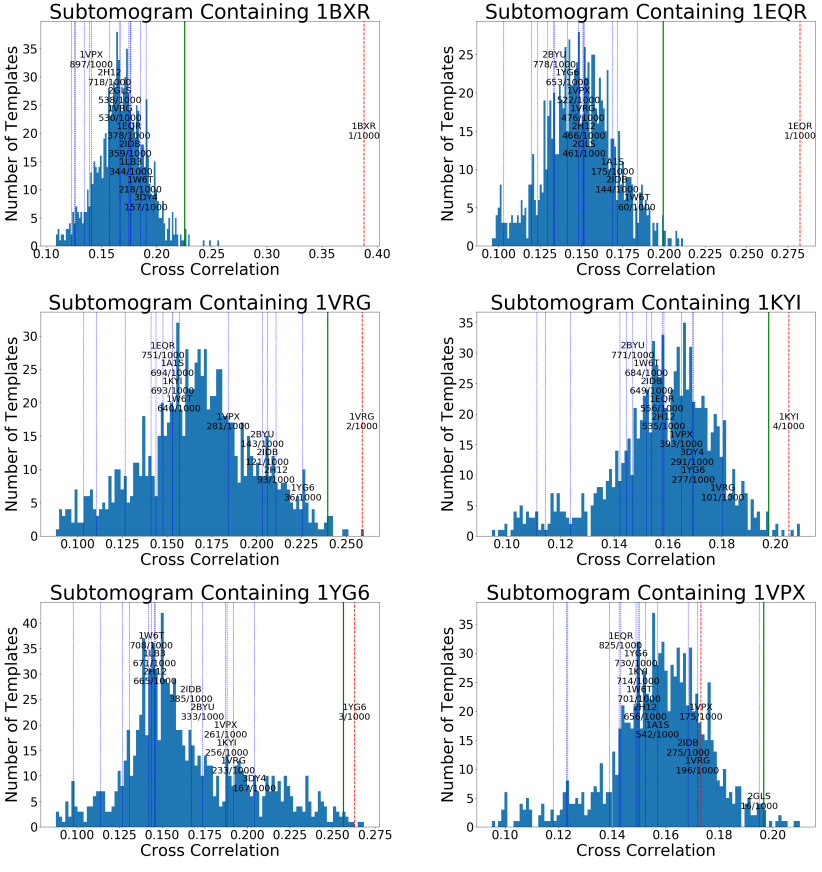

Figure 4: First batch of the remainder of the hypothesis tests on simulated subtomograms. Subtomograms were simulated using default parameters discussed in Appendix 1.2; 0.03 SNR with 0.2nm smoothing.

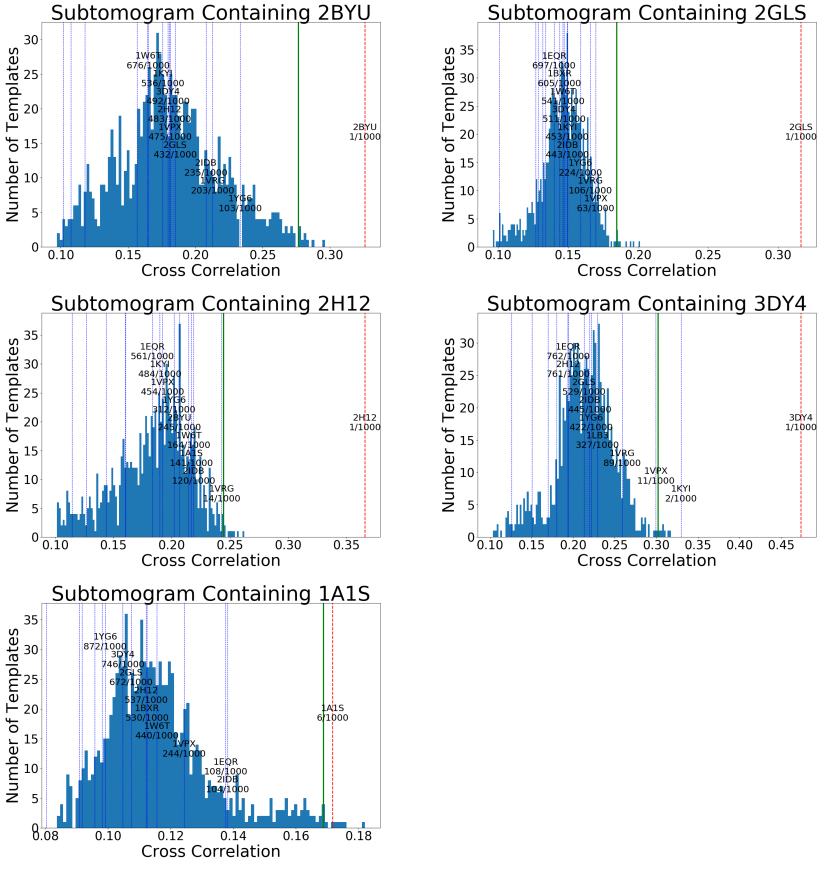

Figure 5: Second batch of the remainder of the hypothesis tests on simulated subtomograms. Subtomograms were simulated using default parameters discussed in Appendix 1.2; 0.03 SNR with 0.2nm smoothing.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Ben Athiwaratkun and Keegan Kang. Feature representation in convolutional neural networks. *arXiv preprint arXiv:1507.02313*, 2015.

[3] M. Beck, J.A. Malmström, V. Lange, A. Schmidt, E.W. Deutsch, and R. Aebersold. Visual proteomics of the human pathogen Leptospira interrogans. *Nature methods*, 6 (11):817–823, 2009. ISSN 1548-7091.

[4] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, TN Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235, 2000.

[5] F. Förster, S. Pruggnaller, A. Seybert, and A.S. Frangakis. Classification of cryo-electron sub-tomograms using constrained correlation. *Journal of structural biology*, 161(3):276–286, 2008. ISSN 1047-8477.

[6] J. Frank. *Three-dimensional electron microscopy of macromolecular assemblies*. Oxford University Press, New York, 2006. ISBN 0195182189.

[7] Zachary Frazier, Min Xu, and Frank Alber. Tomominer and tomominercloud: A software platform for large-scale subtomogram structural analysis. *Structure*, 25(6):951–961, 2017.

[8] Jesús G Galaz-Montoya, John Flanagan, Michael F Schmid, and Steven J Ludtke. Single particle tomography in eman2. *Journal of structural biology*, 190(3):279–290, 2015.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

[15] G. McMullan, S. Chen, R. Henderson, and A.R. Faruqi. Detective quantum efficiency of electron area detectors in electron microscopy. *Ultramicroscopy*, 109(9):1126–1143, 2009. ISSN 0304-3991.

[16] S. Nickell, F. Förster, A. Linaroudis, W.D. Net, F. Beck, R. Hegerl, W. Baumeister, and J.M. Plitzko. TOM software toolbox: acquisition and analysis for electron tomography. *Journal of Structural Biology*, 149(3):227–234, 2005.

[17] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[18] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[19] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[20] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[21] W. Wriggers, R.A. Milligan, and J.A. McCammon. Situs: A Package for Docking Crystal Structures into Low-Resolution Maps from Electron Microscopy. *Journal of Structural Biology*, 125(2-3):185–195, 1999. ISSN 1047-8477.

[22] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[23] Min Xu and Frank Alber. Automated target segmentation and real space fast alignment methods for high-throughput classification and averaging of crowded cryo-electron subtomograms. *Bioinformatics*, 29(13):i274–i282, 2013.

[24] Tzviya Zeev-Ben-Mordehai, Daven Vasishtan, Anna Hernández Durán, Benjamin Vollmer, Paul White, Arun Prasad Pandurangan, C Alistair Siebert, Maya Topf, and Kay Grünewald. Two distinct trimeric conformations of natively membrane-anchored full-length herpes simplex virus 1 glycoprotein b. *Proceedings of the National Academy of Sciences*, 113(15):4176–4181, 2016.