

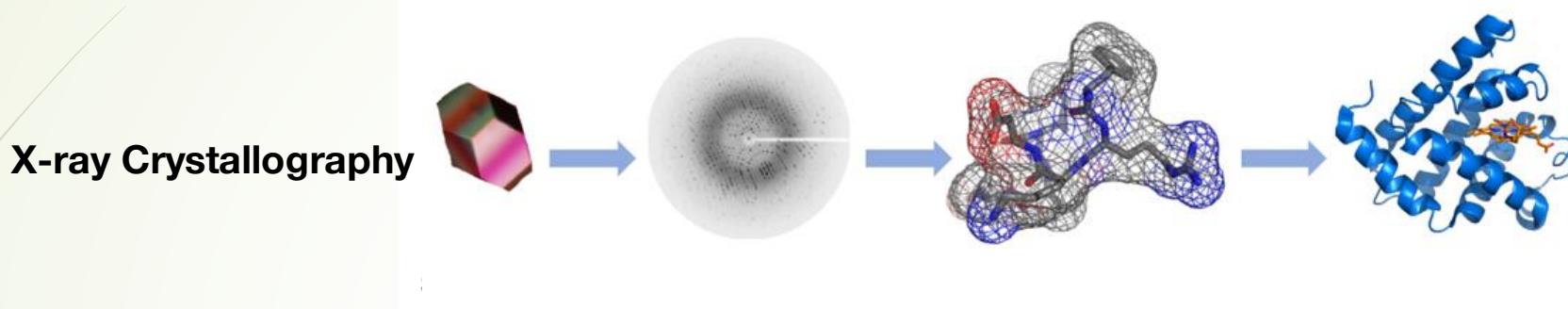
Machine Learning Problems in Cryo-EM

Hanlin Gu 2018-11-21

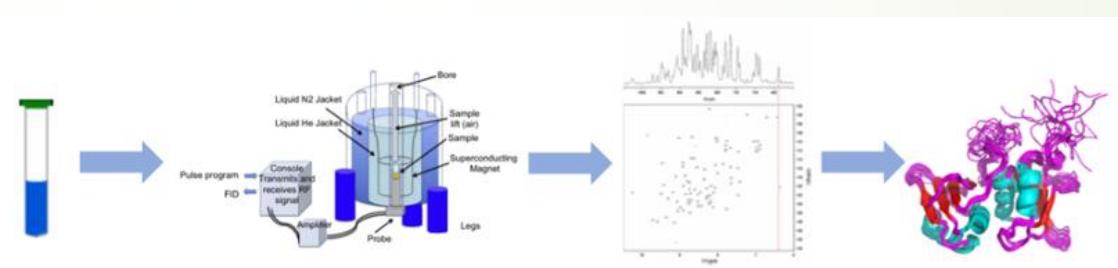
Outline

- ▶ Introduction for cryo-em
- ▶ Particle picking problem
- ▶ Denoising problem

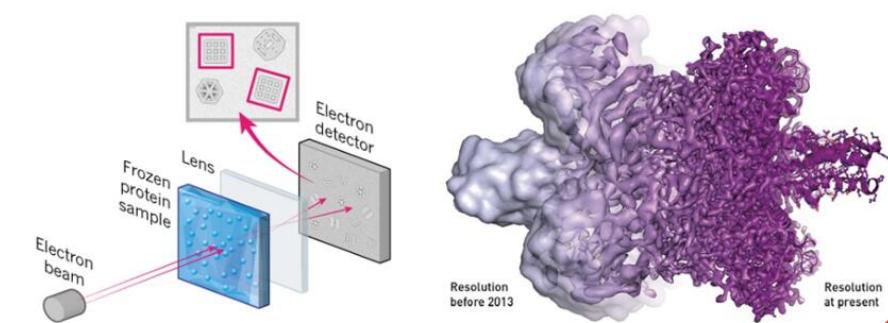
Experimental techniques to resolve atomic structure



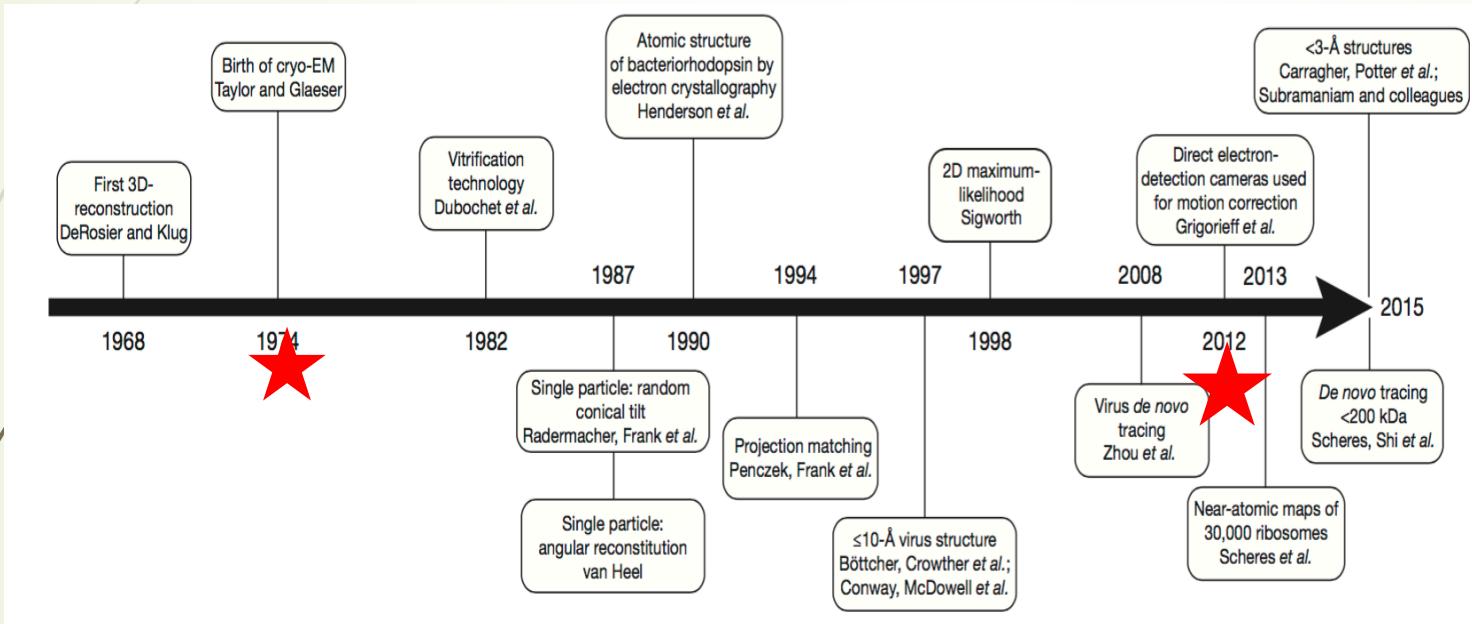
Nuclear magnetic resonance (NMR)



Cryo-Electron Microscopy (Cryo-EM)

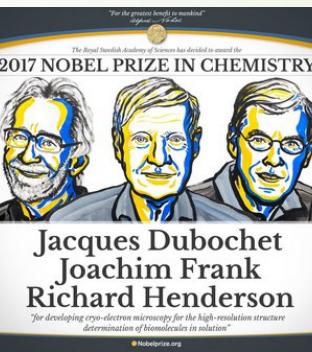


Cryo-EM: Rising star



1974: birth

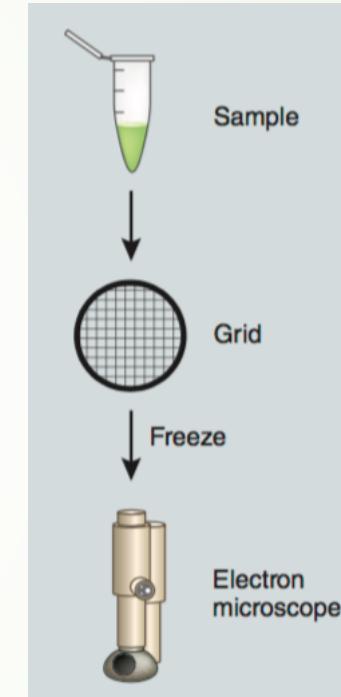
2012:
Resolution
Revolution



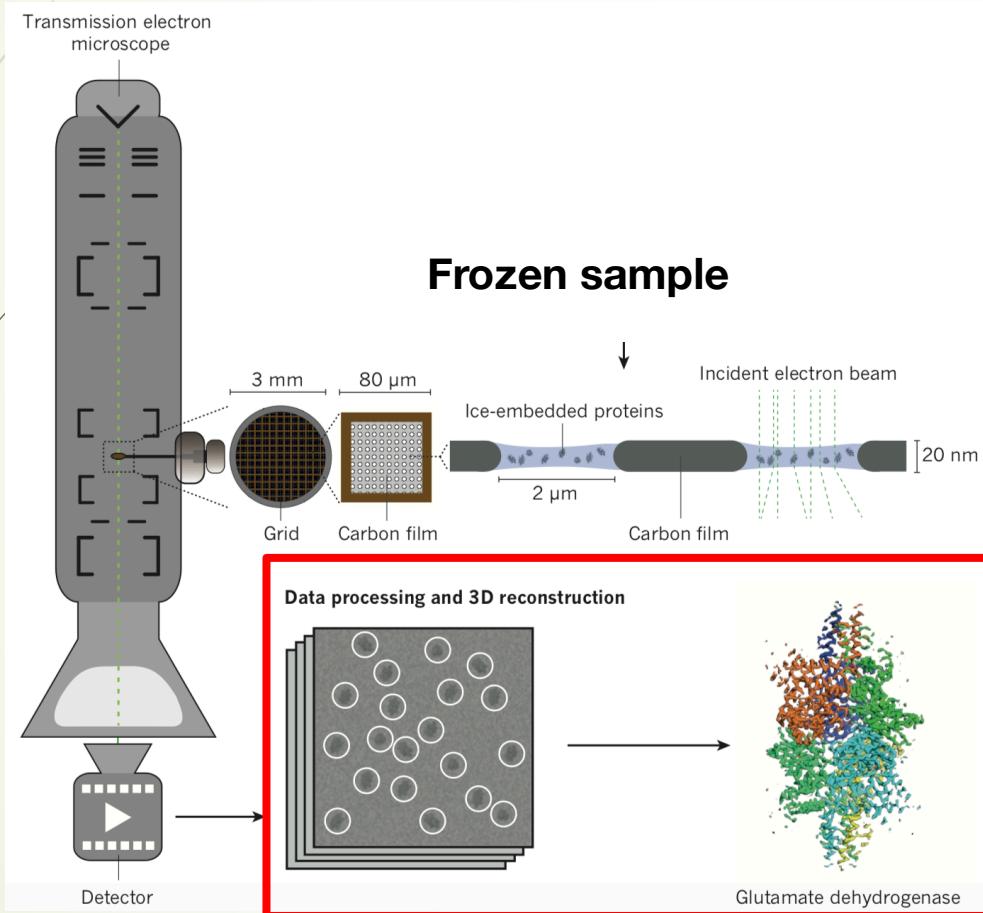
2017: Nobel
Prize

Cryo-EM

► *Cryo-electron microscopy (cryo-EM), or electron cryomicroscopy, is a form of transmission electron microscopy (TEM) where the sample is studied at cryogenic temperatures (generally liquid nitrogen temperatures ~ 80K).*

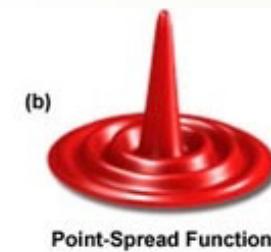
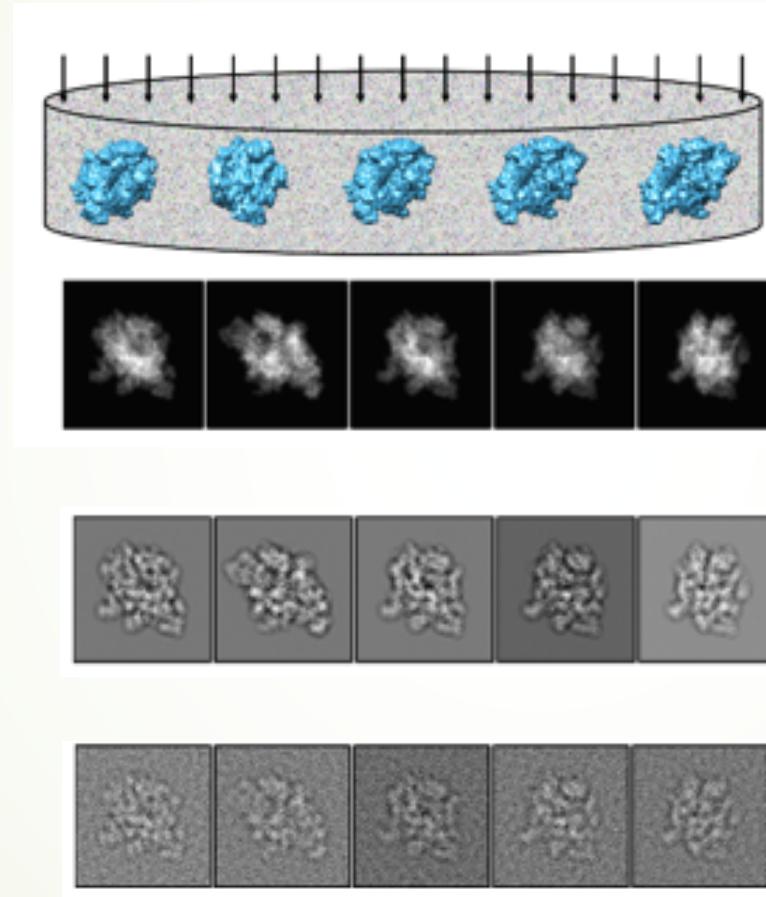
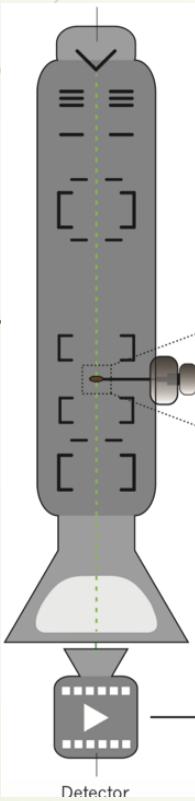


Cryo-EM: "Take photo" of biomolecular



Structural
reconstruction from the
micrograph

Photos: Projection of a molecular along a certain direction

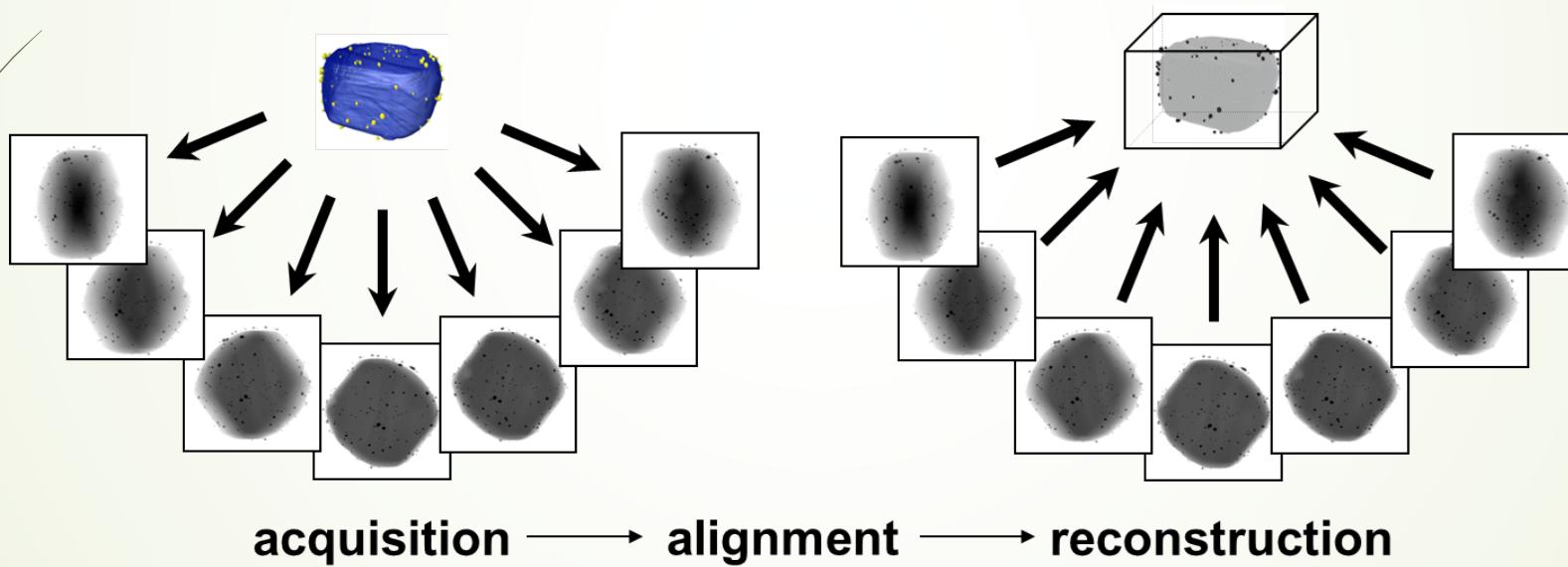

 $P^{\Phi V}$
 $PSF \otimes (P^{\Phi V})$

$$X = PSF \otimes (P^{\Phi V}) + N$$

Distorted by the non-ideal lens and noises from multiple sources

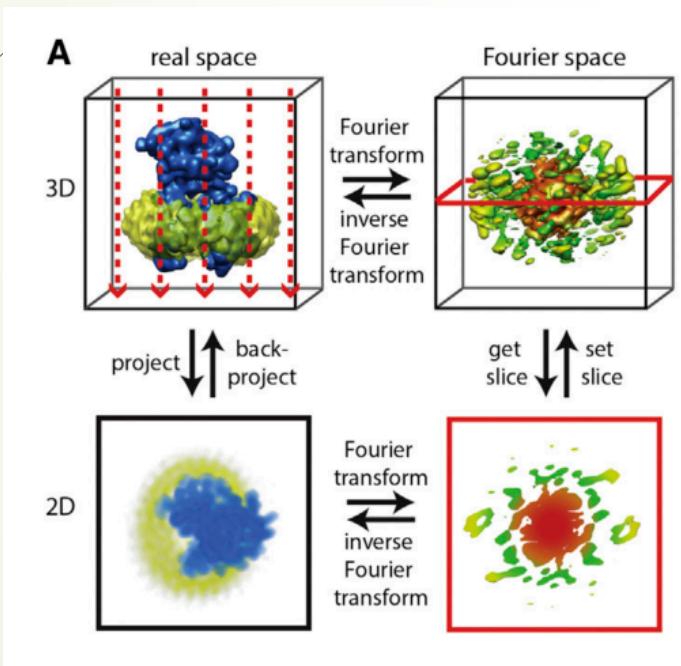
Experiments vs. Structural Reconstruction

- ▶ Assign correct angle for each noisy image

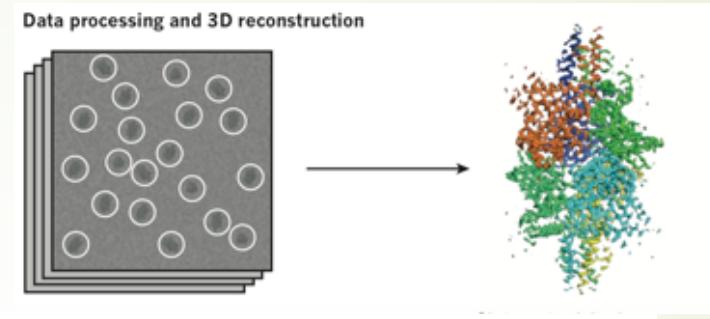
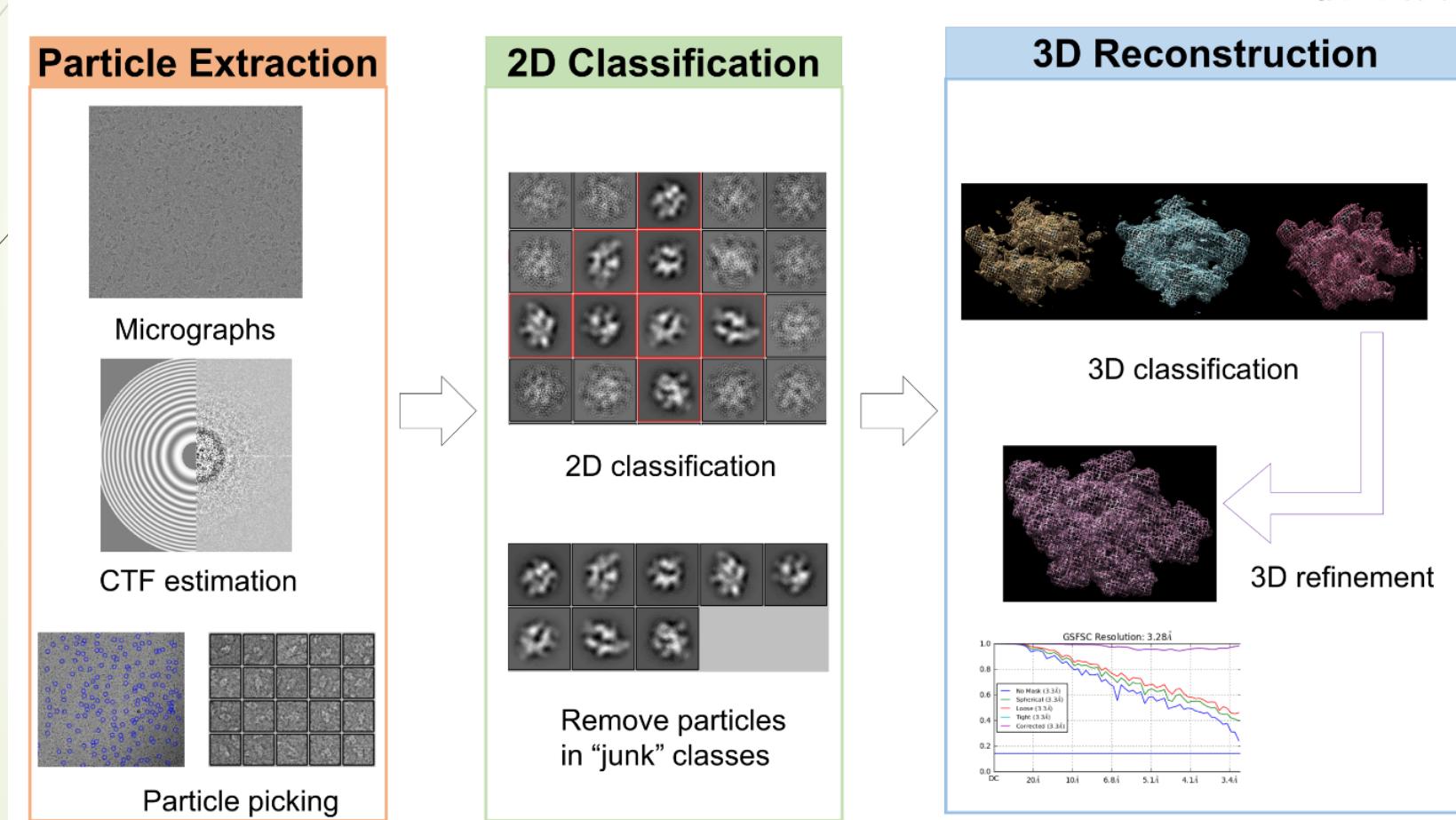


Projection slice theorem

- Once we have the correct angles, we can put the slices back and then get the structure



Common protocol



Popular software: cryoSPARC

- Maximized posterior probability



$$\arg \max_{V_{1\dots K}} \log p(V_{1\dots K} | X_{1\dots N}) =$$
$$\arg \max_{V_{1\dots K}} \sum_{i=1}^N \log \sum_{j=1}^K \frac{1}{K} \int p(X_i, \phi_i | V_j) d\phi_i + \log p(V_{1\dots K})$$

Outline

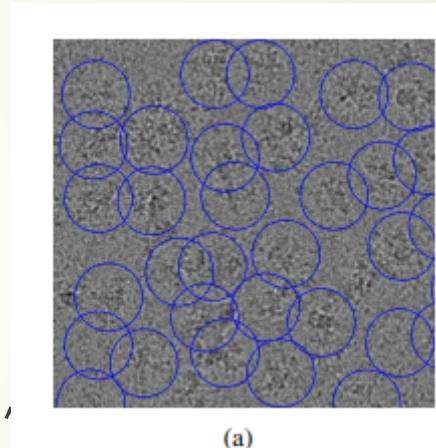
- ▶ Introduction for cryo-em

- ▶ Particle picking problem

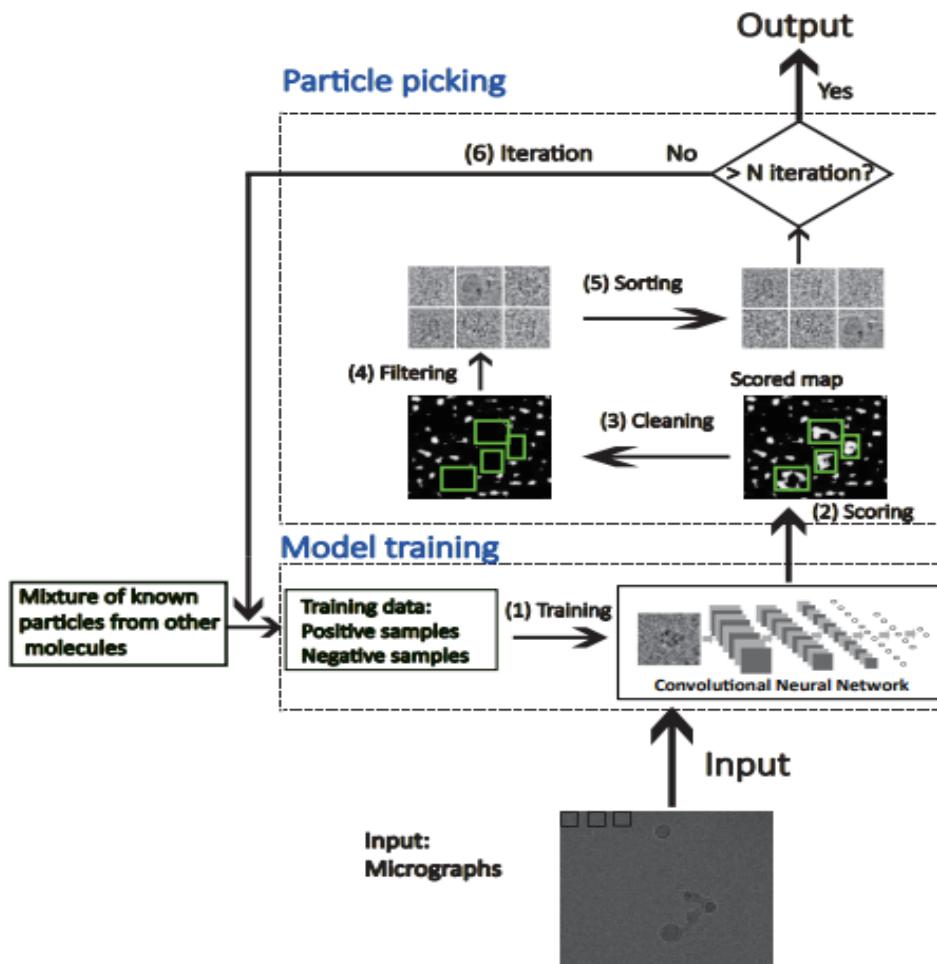
- ▶ Denoising problem

Problem 1: Particle picking

- ▶ Selection of a tremendous number of (e.g., hundreds of thousands of) high-quality particles from micrographs.
- ▶ This particle picking step is a labor-intensive step. In the past, particles from cryo-EM micrographs are often selected manually.



Fully automated particle picking





Steps:

- ▶ Scoring: a sliding window scans micrograph as input data. Getting a score map by cnn.
- ▶ Cleaning: Remove the pixel point which neighbors(include itself) are all larger than a cutoff.
- ▶ Filtering: Refine the current set of particle candidates and also identify the center coordinates of the final remaining particles from the scored map.
- ▶ Sorting: Sort the remaining particle candidates according to their prediction scores.
- ▶ Iteration : Using the particles picked by the previous CNN classifier which was trained over the known particles of other molecules to further refine the CNN model.

Evaluate

- ▶ Evaluated the performance of particle picking approach mainly by comparing the automated picking results to the corresponding reference particles picked manually by human experts

Test data Training data	γ -secretase	Spliceosome	TRPV1
1 type	0.933	0.881	0.841
2 types	0.925	0.892	0.813
3 types	0.908	0.871	0.835
4 types	0.913	0.901	0.818

Outline

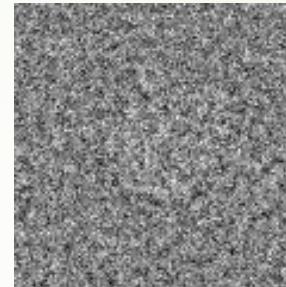
- ▶ Introduction for cryo-em

- ▶ Particle picking problem

- ▶ Denoising problem

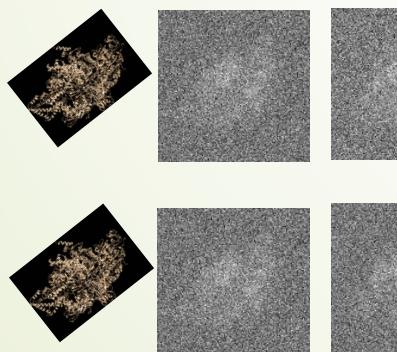
Noise in cryoEM images would obscure the structural differences

- Very large noise and unregular noise



Apo ribosome, K2 detector

- When noise level is comparable with differences between conformations, it is hard to discriminate conformations



MSE=0.029



MSE=0.026

Inter-class diffe ~ intra-class diff

Simulated images at SNR=0.1, MSE(X1,X2)=11, var(noise)=39

Benefits of denoising

- ▶ Denoising cryoEM images can
 - ▶ Benefit angular classification
 - ▶ **Better distinguish different conformations**
 - ▶ Others: improve the performances of common-line detection, improve resolution of reconstructed 3D maps

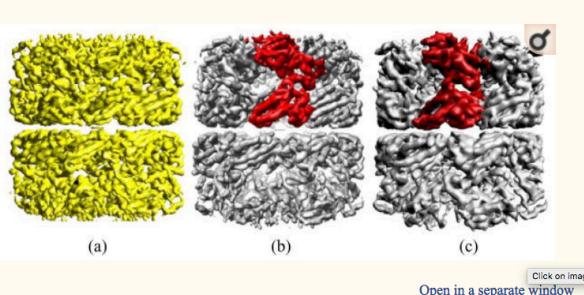


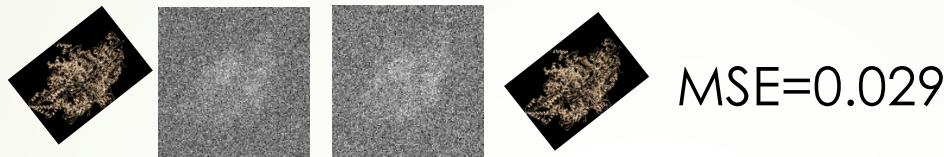
Figure 2

Comparative results between GDTV denoising and non-denoising. (a) shows the original density map. (b) shows the structure reconstruction result in Figure 1.B of [20]. (c) shows our structure reconstruction.

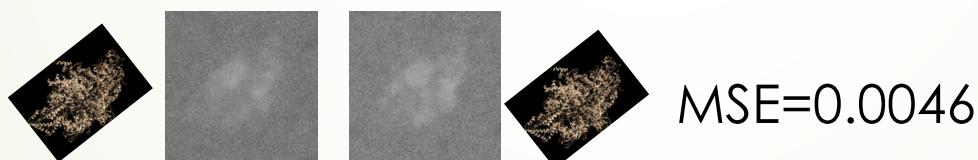
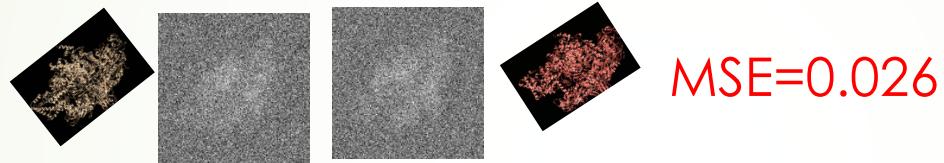
[Click on image](#)

[Open in a separate window](#)

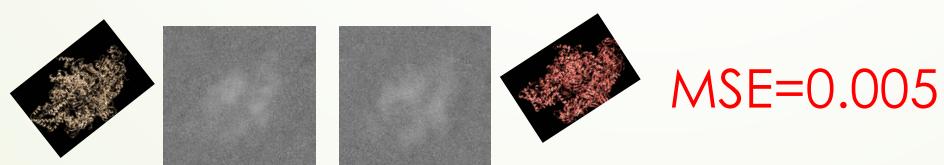
Denoising can benefit discriminating conformations



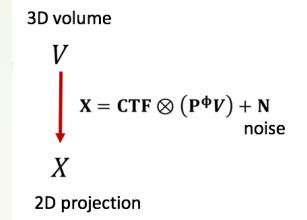
Before denoising



After denoising



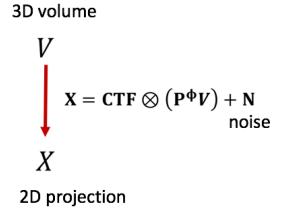
Existing methods in softwares



- ▶ Existing methods in softwares and challenges
 - ▶ Relion, CryoSPARC: Gaussian noise model, **denoising is coupled with the classification process:**
$$X - CTF \otimes (P^\Phi V) \sim N(0, \sigma^2), P(V|X) \propto \int likelihood * priors$$

▶ **Very time consuming and cost a lot of memory**
 - ▶ Aspire: Covariance Wiener Filtering (CWF) (Singer, et al.) needs large number of samples in order to have a good denoising performance.
 - ▶ CWF is computational intensive, and not work well with small data samples.
 - ▶ Ensemble method, memory cost is very high

Some other traditional methods



- ▶ Traditional Wiener Filter
- ▶ Total variation method: chambolle
- ▶ Nonlinear local
- ▶ Mean, median..

<code>skimage.restoration.wiener</code> (image, psf, balance)	Wiener-Hunt deconvolution
<code>skimage.restoration.unsupervised_wiener</code> (...)	Unsupervised Wiener-Hunt deconvolution.
<code>skimage.restoration.richardson_lucy</code> (image, psf)	Richardson-Lucy deconvolution.
<code>skimage.restoration.unwrap_phase</code> (image[, ...])	Recover the original from a wrapped phase image.
<code>skimage.restoration.denoise_tv_bregman</code> (...)	Perform total-variation denoising using split-Bregman optimization.
<code>skimage.restoration.denoise_tv_chambolle</code> (image)	Perform total-variation denoising on n-dimensional images.
<code>skimage.restoration.denoise_bilateral</code> (image)	Denoise image using bilateral filter.
<code>skimage.restoration.denoise_wavelet</code> (image[, ...])	Perform wavelet denoising on an image.
<code>skimage.restoration.denoise_nl_means</code> (image)	Perform non-local means denoising on 2-D or 3-D grayscale images, and 2-D RGB images.
<code>skimage.restoration.estimate_sigma</code> (image[, ...])	Robust wavelet-based estimator of the (Gaussian) noise standard deviation.
<code>skimage.restoration.inpaint_biharmonic</code> (...)	Inpaint masked points in image with biharmonic equations.
<code>skimage.restoration.cycle_spin</code> (x, func, ...)	Cycle spinning (repeatedly apply func to shifted versions of x).

Some cutting edge methods in applied mathematics

- ▶ K-SVD
- ▶ Data-Driven Tight Frame for Multi-Channel Images: DDTF
- ▶ Block-matching and 3D filtering: BM3D

The diagram illustrates the reconstruction process. At the top, a 3D volume V is shown. A red arrow points downwards to a 2D projection X . To the right of the arrow, the equation $\mathbf{X} = \mathbf{CTF} \otimes (\mathbf{P}^\Phi \mathbf{V}) + \mathbf{N}_{\text{noise}}$ is displayed, where \mathbf{CTF} represents the Convolutional Transport Function, \otimes denotes convolution, \mathbf{P}^Φ is a frame operator, and $\mathbf{N}_{\text{noise}}$ represents noise.

Block-matching and 3D filtering

- ▶ Finding the image patches similar to a given image patch and grouping them in a 3D block
- ▶ 3D linear transform of the 3D block and set threshold to remove high frequency data;
- ▶ inverse 3D transformation
- ▶ Aggregation: average images of block in different weights

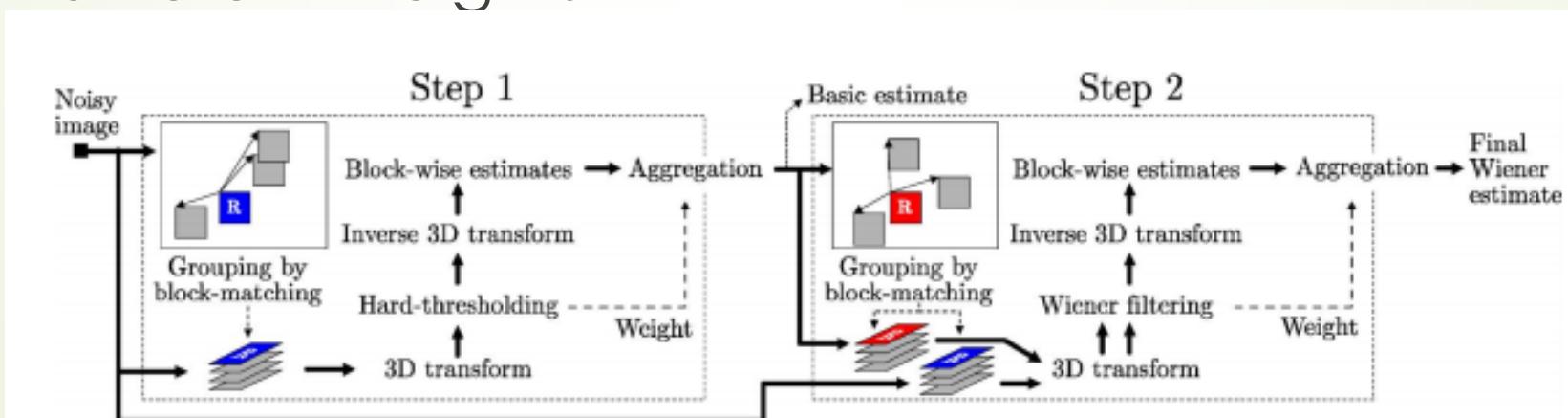


Fig. 3. Flowchart of the proposed image denoising algorithm. The operations surrounded by dashed lines are repeated for each processed block (marked w "R").

K-SVD

- ▶ In applied mathematics, K-SVD is a **dictionary learning** algorithm for creating a dictionary for **sparse representations**, via a singular value decomposition approach.
- ▶ K-SVD is a generalization of the k-means clustering method, and it works by iteratively alternating between sparse coding the input data based on the current dictionary, and updating the atoms in the dictionary to better fit the data.

Wikipedia: K-SVD

K-SVD

Task: Denoise a given image Y from white and additive Gaussian white noise with standard deviation σ .

Algorithm Parameters: n - block size, k - dictionary size, J - number of training iterations, λ - Lagrange multiplier, and C - noise gain.

$$\min_{\mathbf{X}, \mathbf{D}, \mathbf{A}} \left\{ \lambda \|\mathbf{Y} - \mathbf{X}\| + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0 + \sum_{ij} \|\mathbf{D}\alpha_{ij} - R_{ij}\mathbf{X}\|_2^2 \right\}$$

1. Initialization : Set $\mathbf{X} = \mathbf{Y}$, \mathbf{D} = overcomplete DCT dictionary.

2. Repeat J times:

- *Sparse Coding Stage:* Use any pursuit algorithm to compute the representation vectors α_{ij} for each patch $R_{ij}\mathbf{X}$, by approximating the solution of

$$\forall_{ij} \min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \quad \text{s.t.} \quad \|R_{ij}\mathbf{X} - \mathbf{D}\alpha_{ij}\|_2^2 \leq (C\sigma)^2.$$

- *Dictionary Update Stage:* For each column $l = 1, 2, \dots, k$ in \mathbf{D} , update it by

- Find the set of patches that use this atom, $\omega_l = \{(i, j) | \alpha_{ij}(l) \neq 0\}$.
- For each index $(i, j) \in \omega_l$, compute its representation error

$$\mathbf{e}_{ij}^l = R_{ij}\mathbf{X}_{ij} - \sum_{m \neq l} \mathbf{d}_m \alpha_{ij}(m).$$

- set \mathbf{E}_l as the matrix whose columns are $\{\mathbf{e}_{ij}^l\}_{(i,j) \in \omega_l}$
- Apply SVD decomposition $\mathbf{E}_l = \mathbf{U} \mathbf{\Delta} \mathbf{V}^T$. Choose the updated dictionary column $\tilde{\mathbf{d}}_l$ to be the first column of \mathbf{U} . Update the coefficient values $\{\alpha_{ij}(l)\}_{(i,j) \in \omega_l}$ to be the entries of \mathbf{V} multiplied by $\mathbf{\Delta}(1, 1)$.

3. Set:

$$\mathbf{X} = \left(\lambda \mathbf{I} + \sum_{ij} R_{ij}^T R_{ij} \right)^{-1} \left(\lambda \mathbf{Y} + \sum_{ij} R_{ij}^T \mathbf{D} \alpha_{ij} \right)$$

<https://sites.fas.harvard.edu/~cs278/paper>

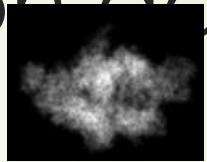
Algorithm proposed: Data Driven Tight Frame (DDTF)

Given an image g , let $G = \{g_1, g_2, \dots, g_m\} \in \mathbb{R}^{n \times m}$ be the training set of image patches of size $\sqrt{n} \times \sqrt{n}$ collected from the image after vectorization. Let $D := [A, D] \in \mathbb{R}^{n \times n}$, be the orthogonal dictionary. A is the input orthogonal atoms from the source, and D is the set of atoms learned from the input image. The dictionary learning scheme is [4]:

$$\begin{aligned} & \min_{D \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{n \times m}} \|G - [A, D]V\|_F^2 + \lambda^2 \|V\|_0, \\ & \text{s.t. } D^T D = I_r; A^T D = 0. \end{aligned}$$

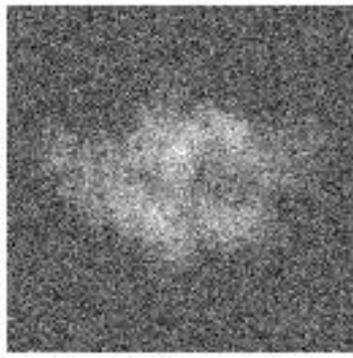
A defines low pass filters from the input image, and D defines high pass filter. In image denoising $A = n^{-1/2}[1, 1, \dots, 1]^T$. If A is empty, then $r = n$. For image denoising, after obtaining the value of G , we synthesize the denoised image g^* from G by averaging the overlapping pixels.

Comparison of the cutting edge algorithms in denoising simulation data

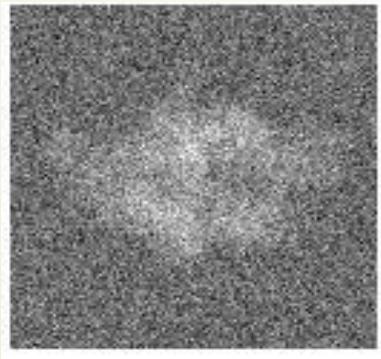


c.f. Clean image

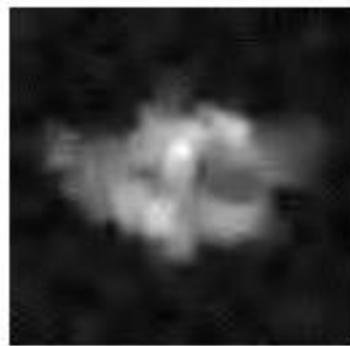
SNR 0.8



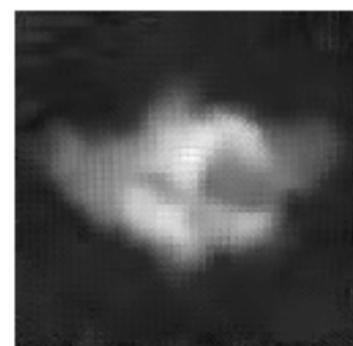
SNR 0.4



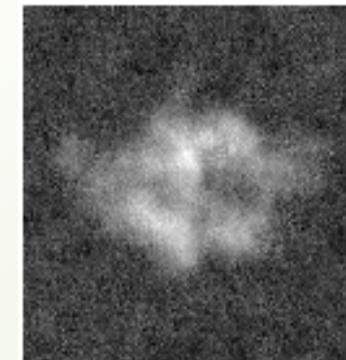
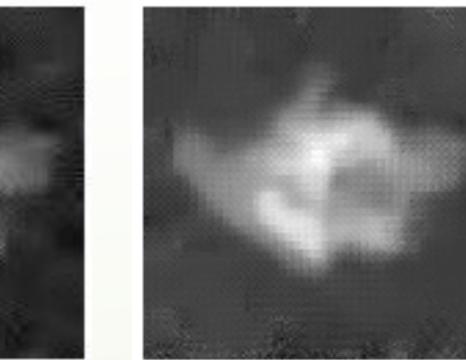
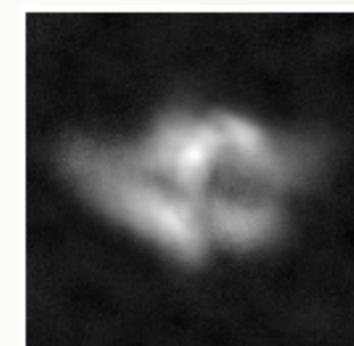
DDTF

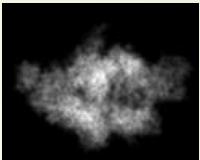


BM3D

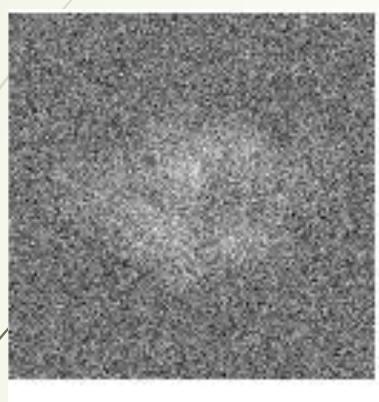


KSVD

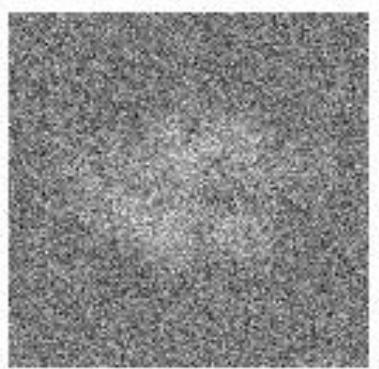




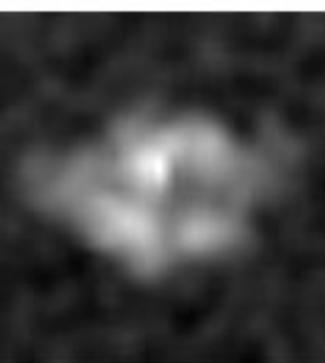
c.f. Clean image



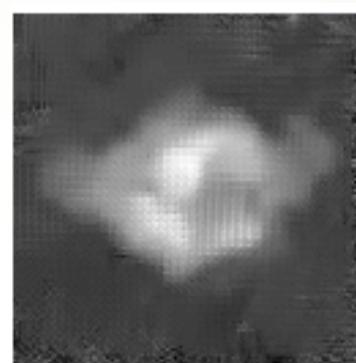
SNR 0.1



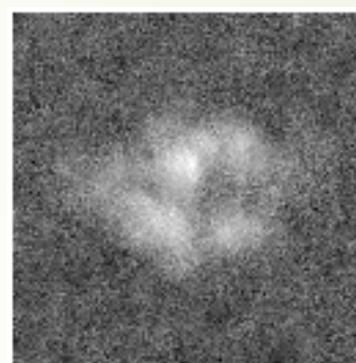
SNR 0.2



DDTF

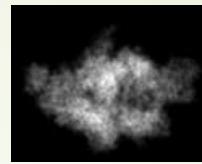


BM3D



KSVD

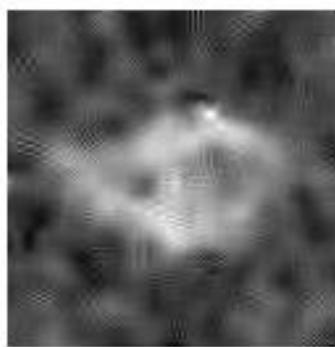
Output



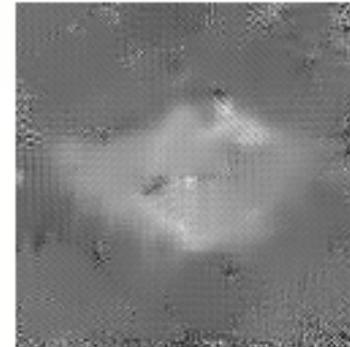
c.f. Clean image

Output

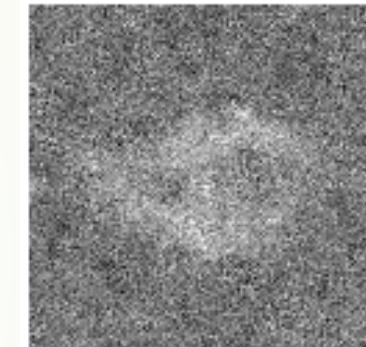
DDTF



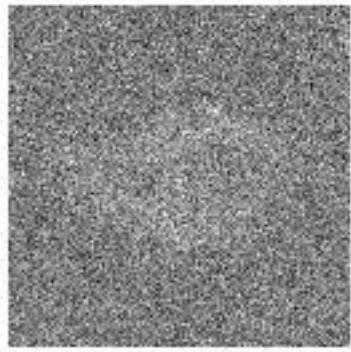
BM3D



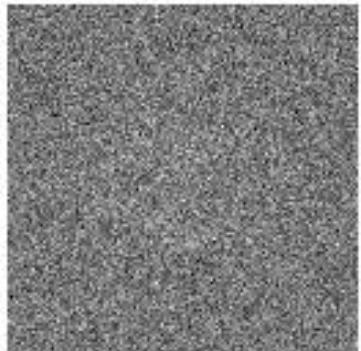
KSVD



SNR 0.05



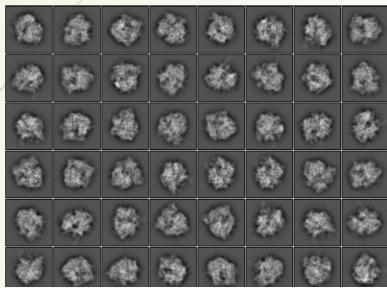
SNR 0.01



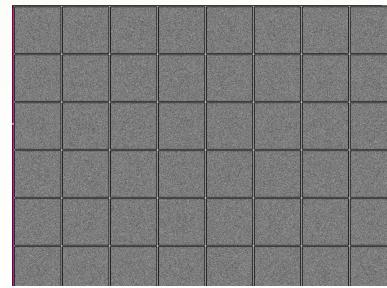
Performances of CWF in aspire

- ▶ Covariance Wiener filtering achieves the denoising of an image by Wiener filtering, where the statistical quantities (mean, covariance matrix) are estimated from massive noisy images.
- ▶ Fast since steerable PCA is applied to reduce the dimensionality
- ▶ Achieve better performances when the number of noisy images increases.

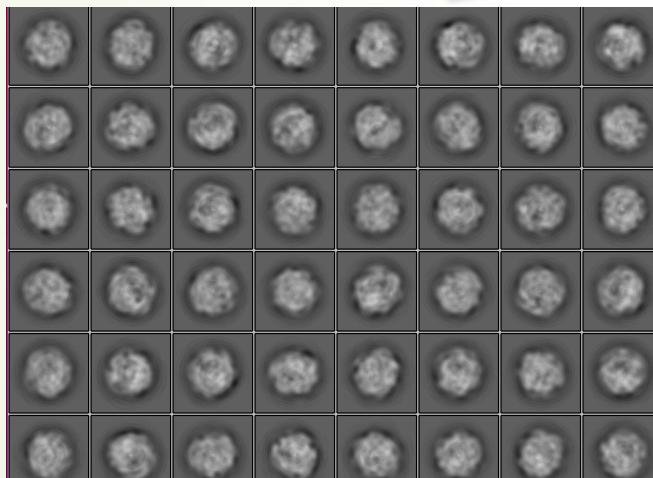
Performances of CWF vs. # of images



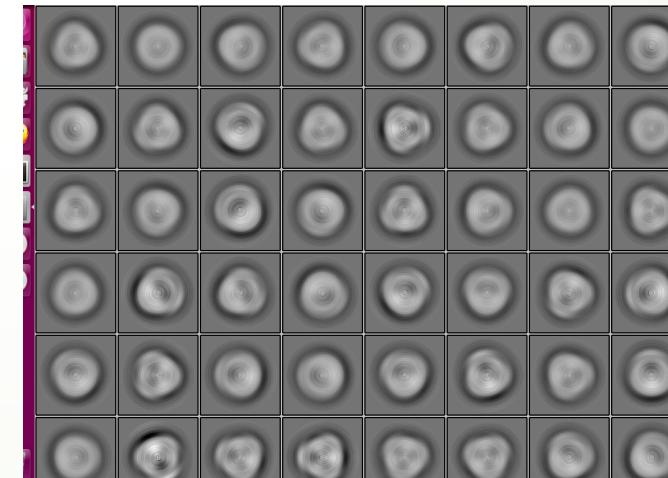
Clean images



SNR=0.01



Denoised results using 1000 images
Cost 2.5G memory

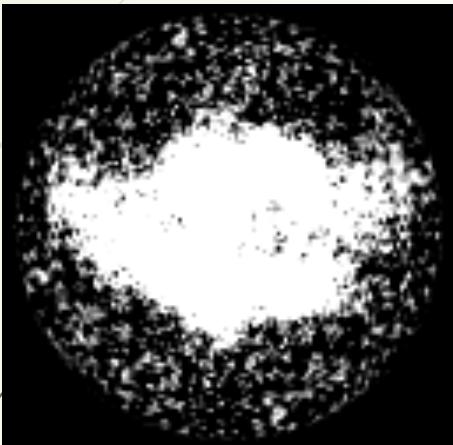


Denoised results using 100 images
Synthetic dataset using P.falciparum 80S ribosome bound to E-tRNA

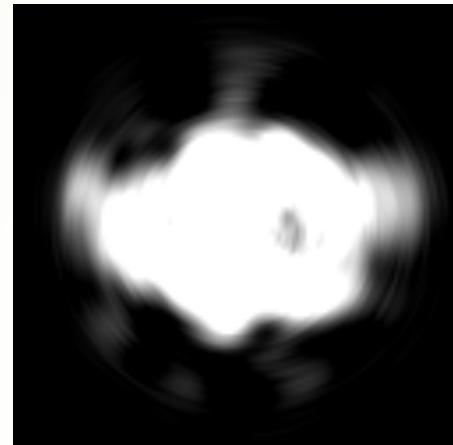
Performance of CWF vs. SNR

SNR=0.

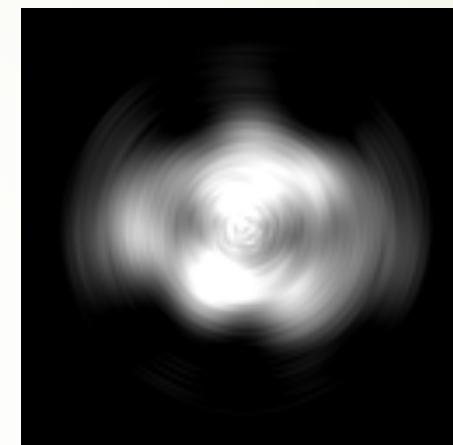
8



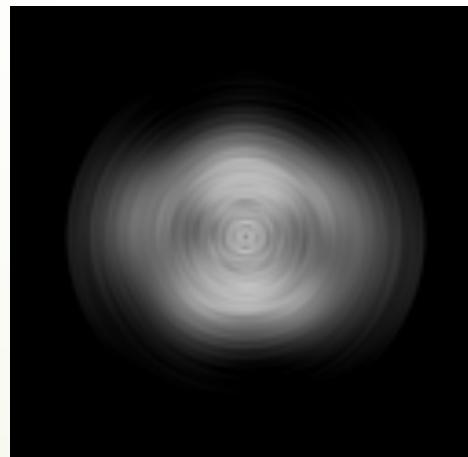
SNR=0.4



SNR=0.1

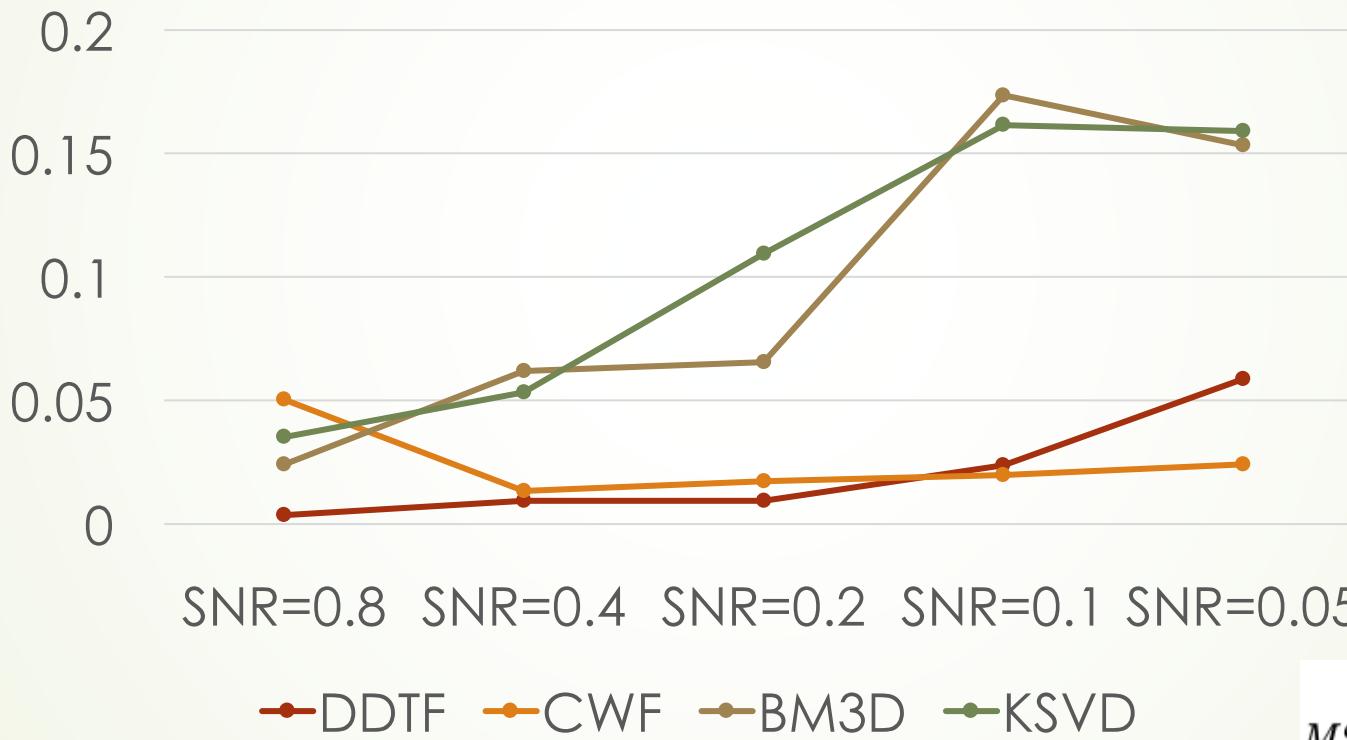


SNR=0.05



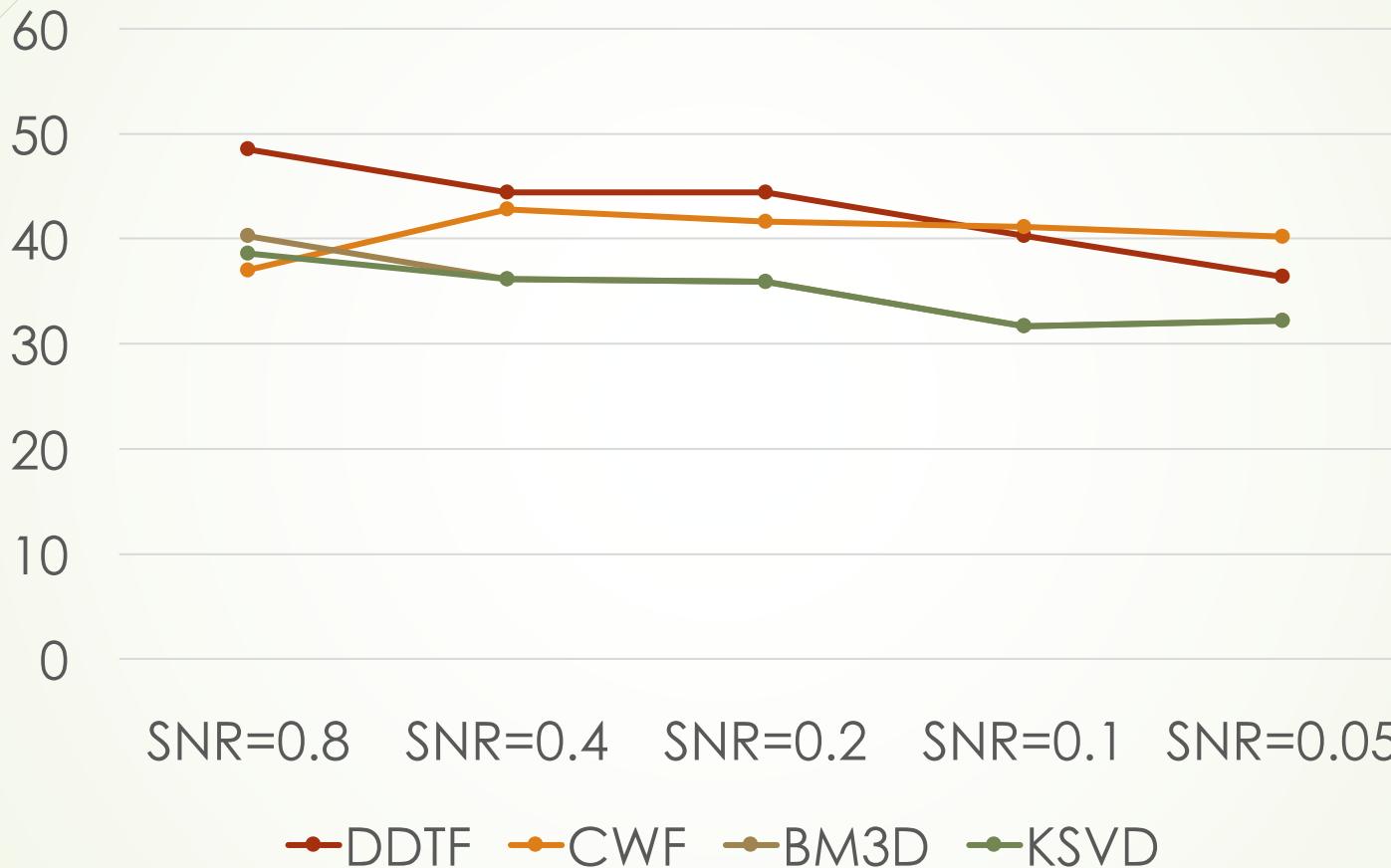
Using 2000 images to denoise

Comparison of Mean square error of denoised images relative to the clean images



$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

Comparison of PSNR of denoised images



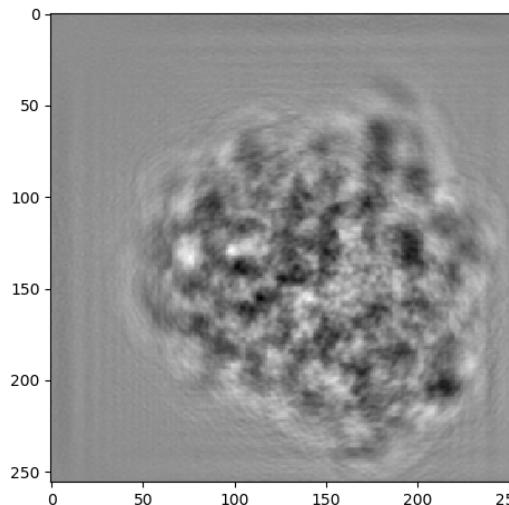
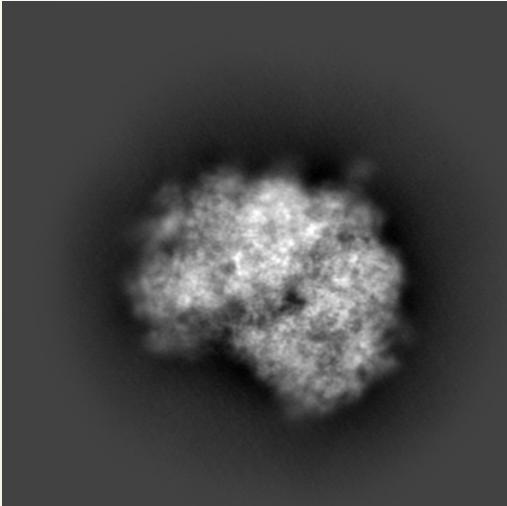
The result is bad as the SNR is low

The PSNR (in dB) is defined as:

$$\begin{aligned}PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\&= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\&= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE)\end{aligned}$$

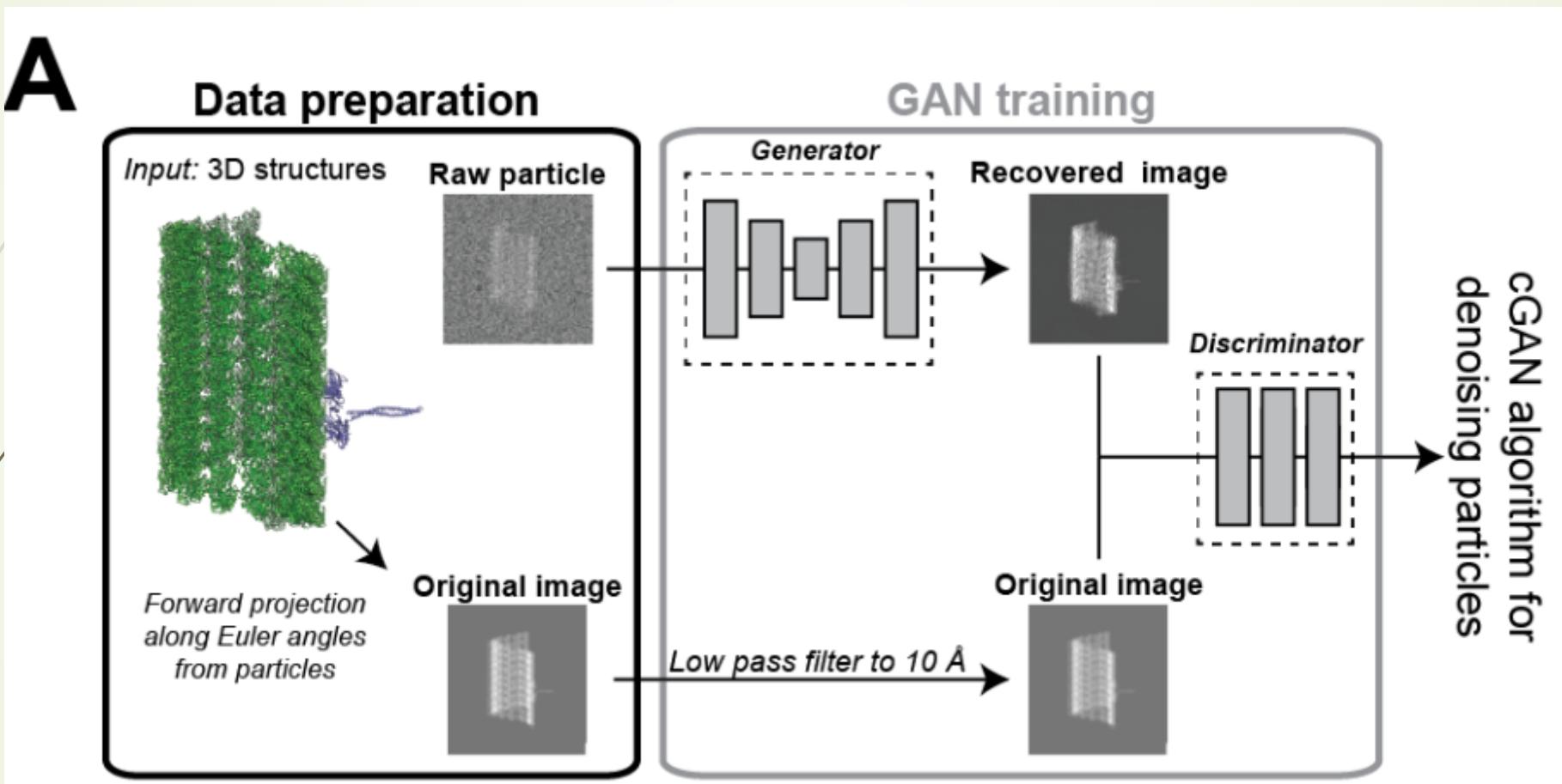
Dataset

- ▶ 1. Choose the first 10000 evidence clean images which from cryosparc.
- ▶ Sometimes, it is necessary to add contrast transfer function to clean images as reference.
Noisy image = clean image* CTF (in fourier domain)+ noise



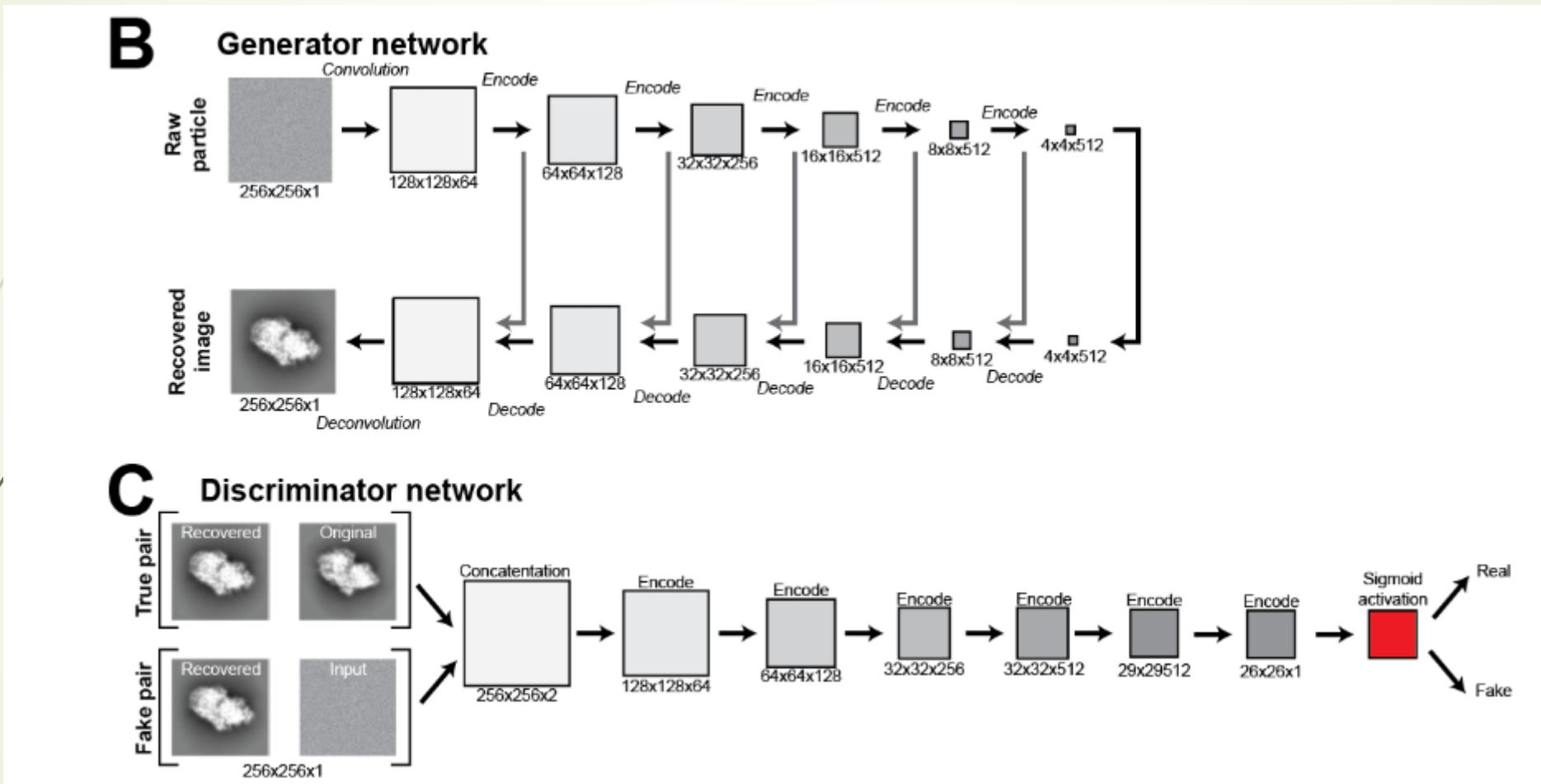
Blurring

GAN_structure

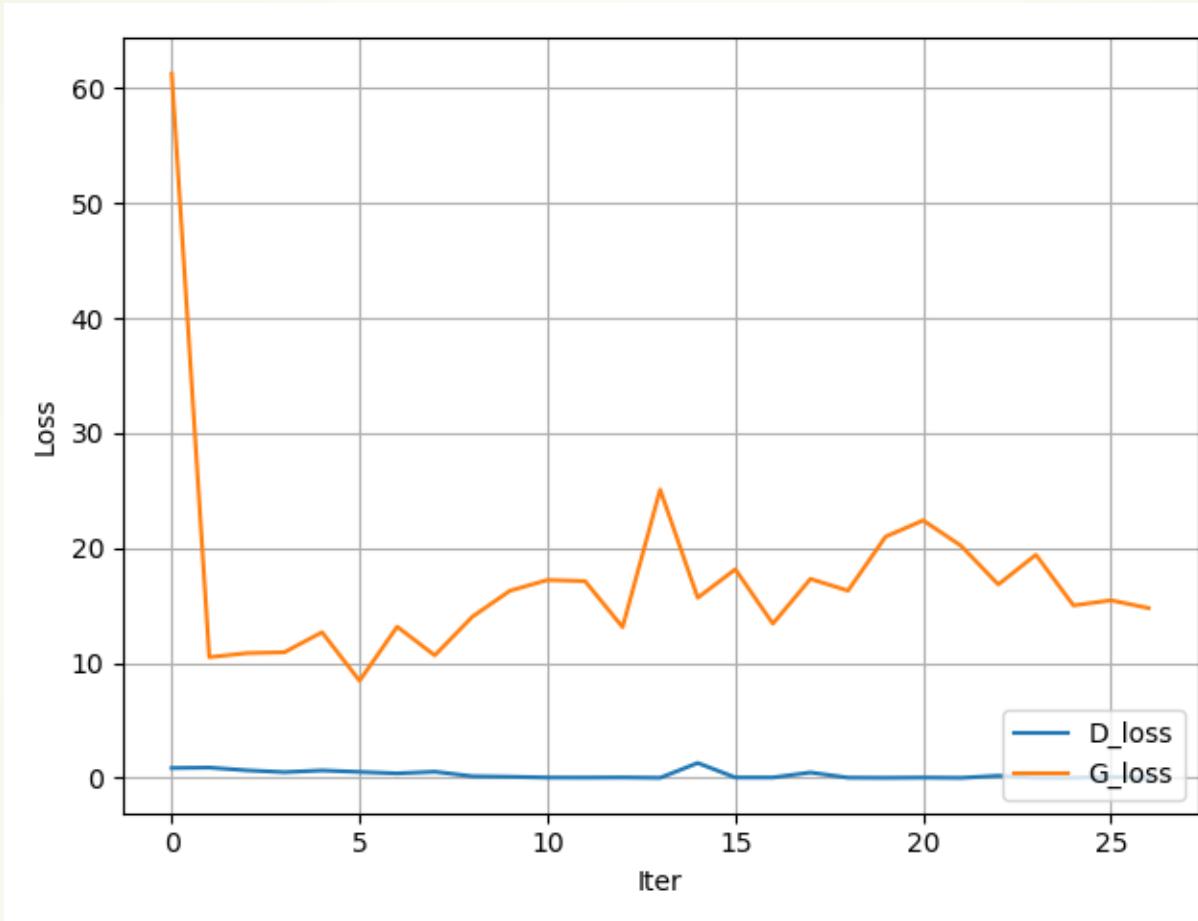


<https://www.biorxiv.org/content/biorxiv/early/2018/02/12/256792.full.pdf>

GAN_structure



Loss function



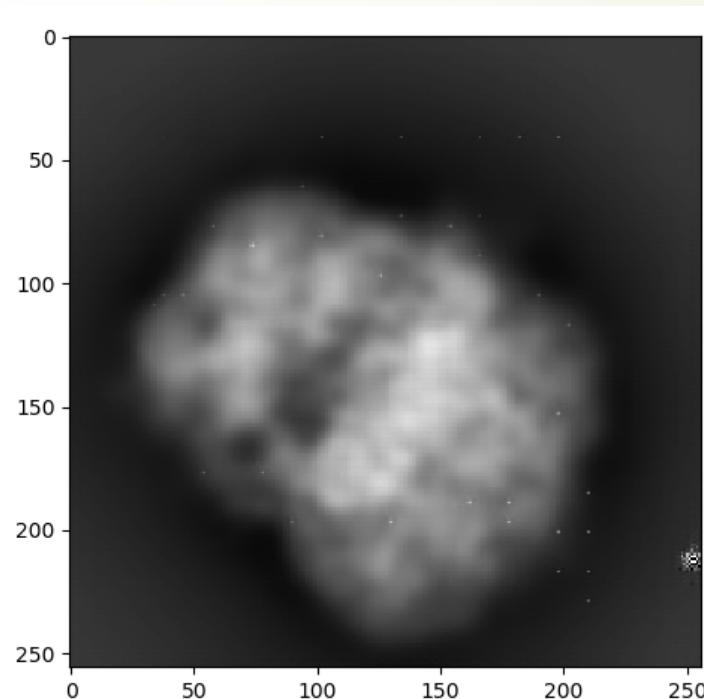
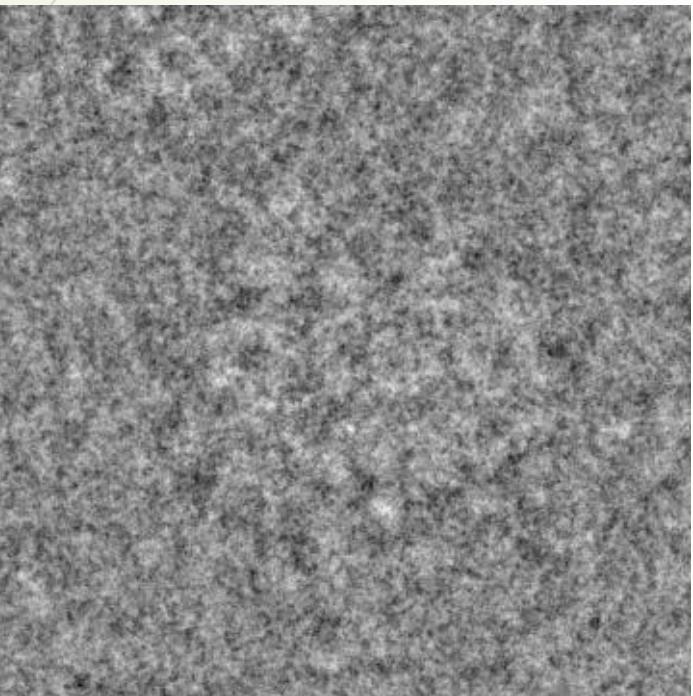
The g loss and d loss are gradually stable



One criterion: MSE between denoised image and clean image

	CTF	Non CTF
MSE(test for 500 , train for 9500)	0.0103	0.0046

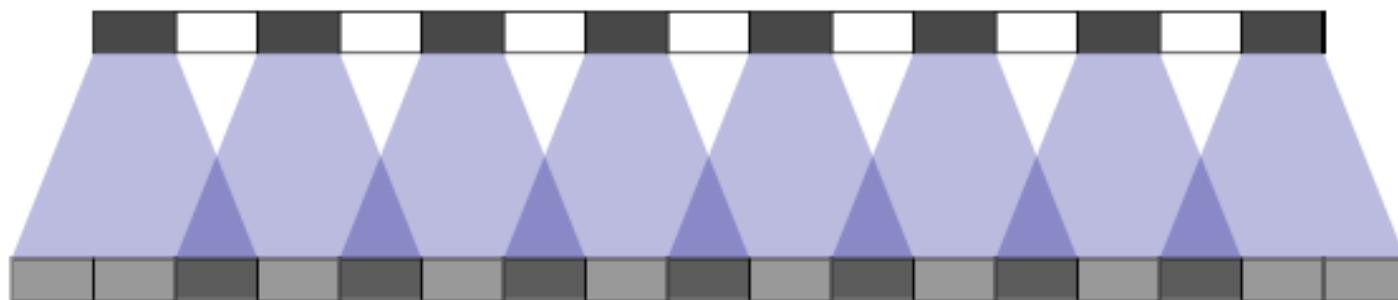
Result(no ctf):



But the right picture is a little blur, maybe you take some improvements.

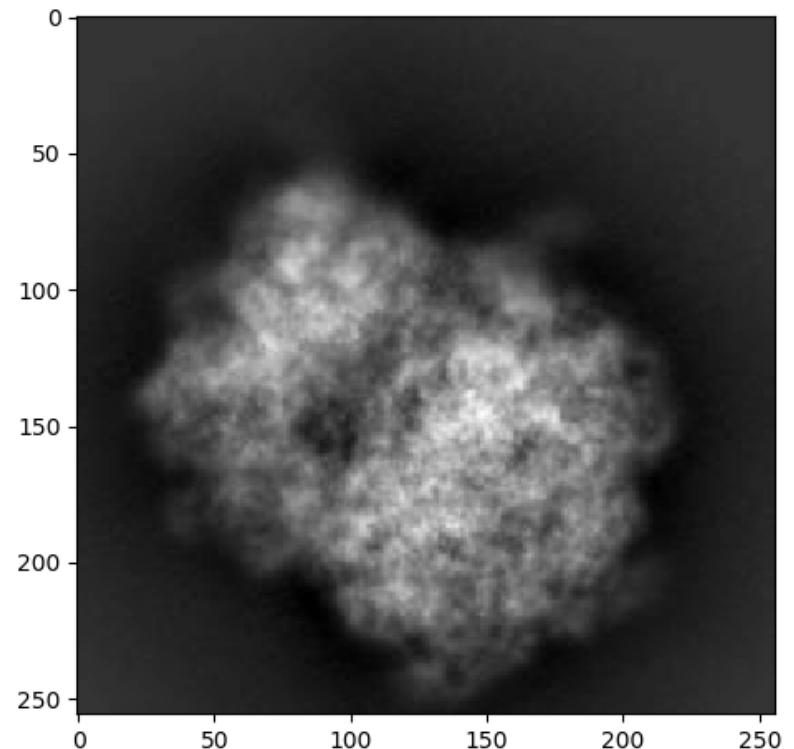
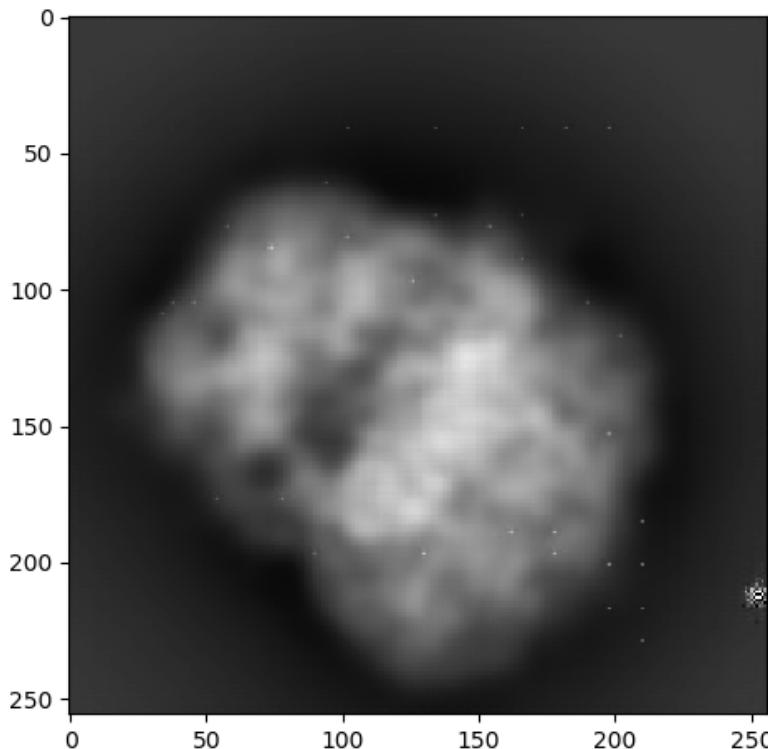
Reason for blurring

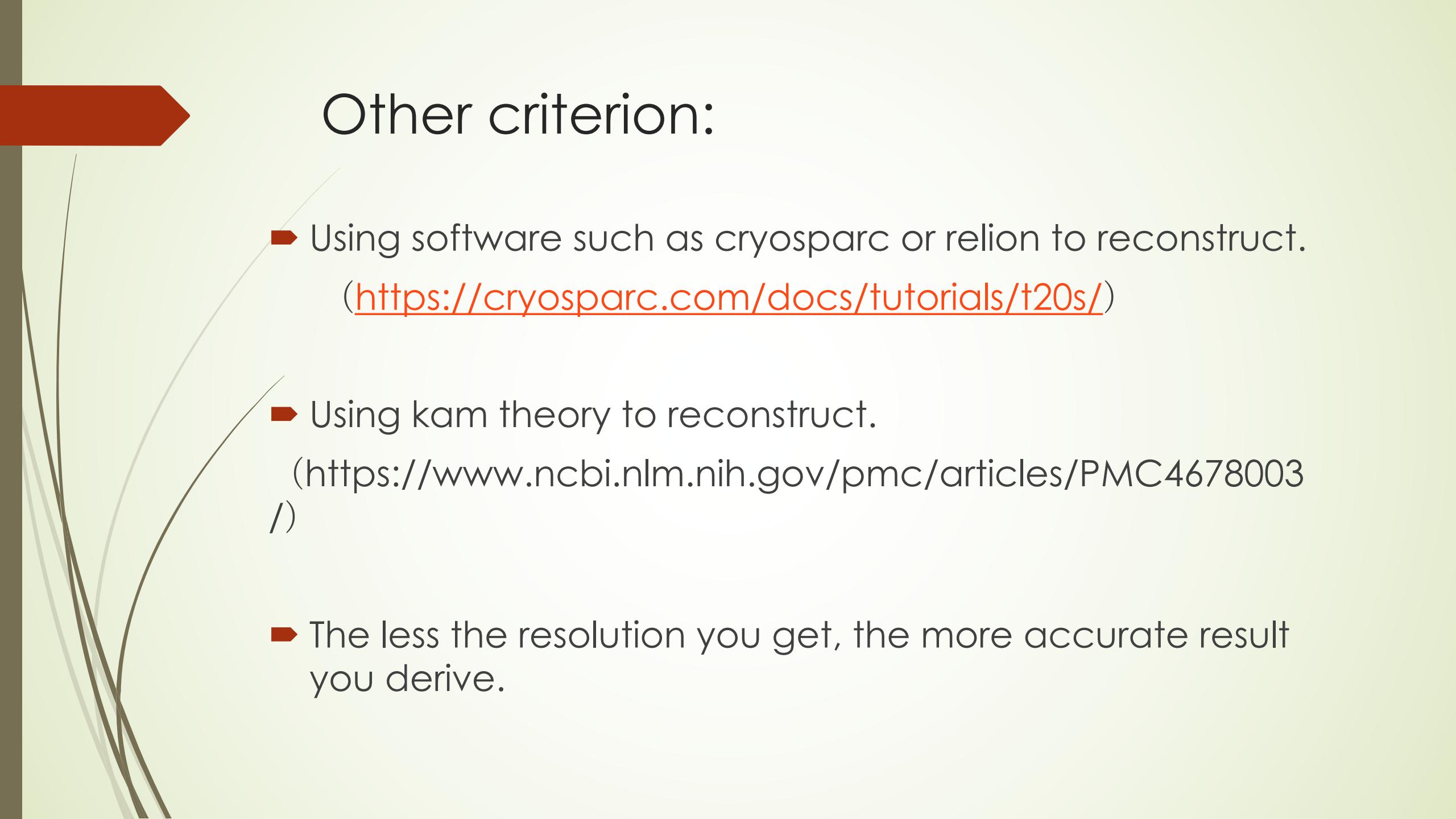
Deconvolution overlap cause
the blur of image



Method

- ▶ A kernel size that is divided by your stride, avoiding the overlap issue.

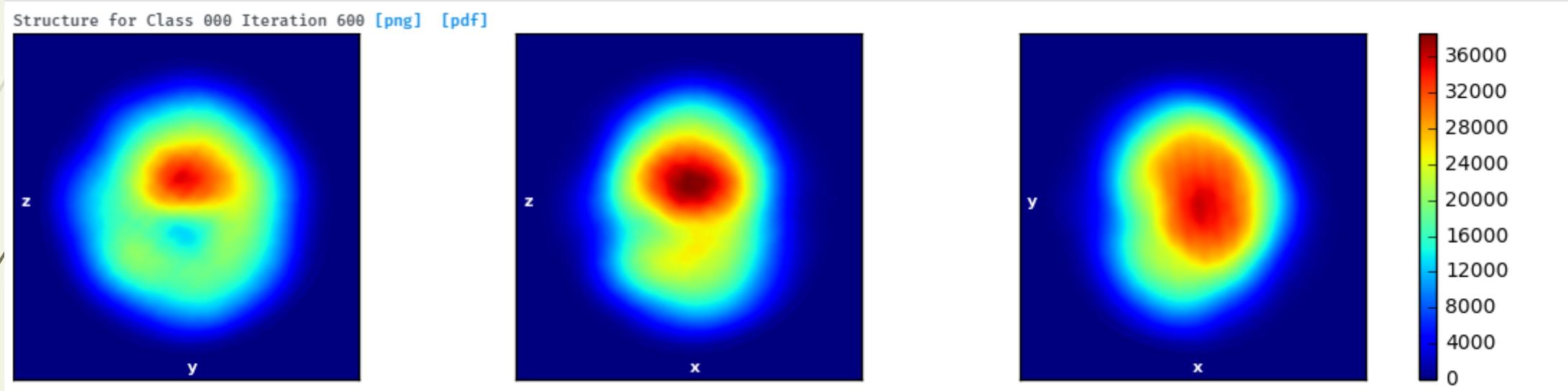




Other criterion:

- ▶ Using software such as cryosparc or relion to reconstruct.
(<https://cryosparc.com/docs/tutorials/t20s/>)
- ▶ Using kam theory to reconstruct.
(<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4678031/>)
- ▶ The less the resolution you get, the more accurate result you derive.

The structure after 600 iteration(using cryosparc)





Discussion

- ▶ Other improvements? Design you own network structure.
- ▶ Other than GAN, using autoencoder or VAE to denoise the experimental data
- ▶ Using the multiple images denoising methods taking using of the connections of images.



Acknowledgment

► Thanks for help from Wei Wang Fredrick and Weizhi Zhu



Thank you!