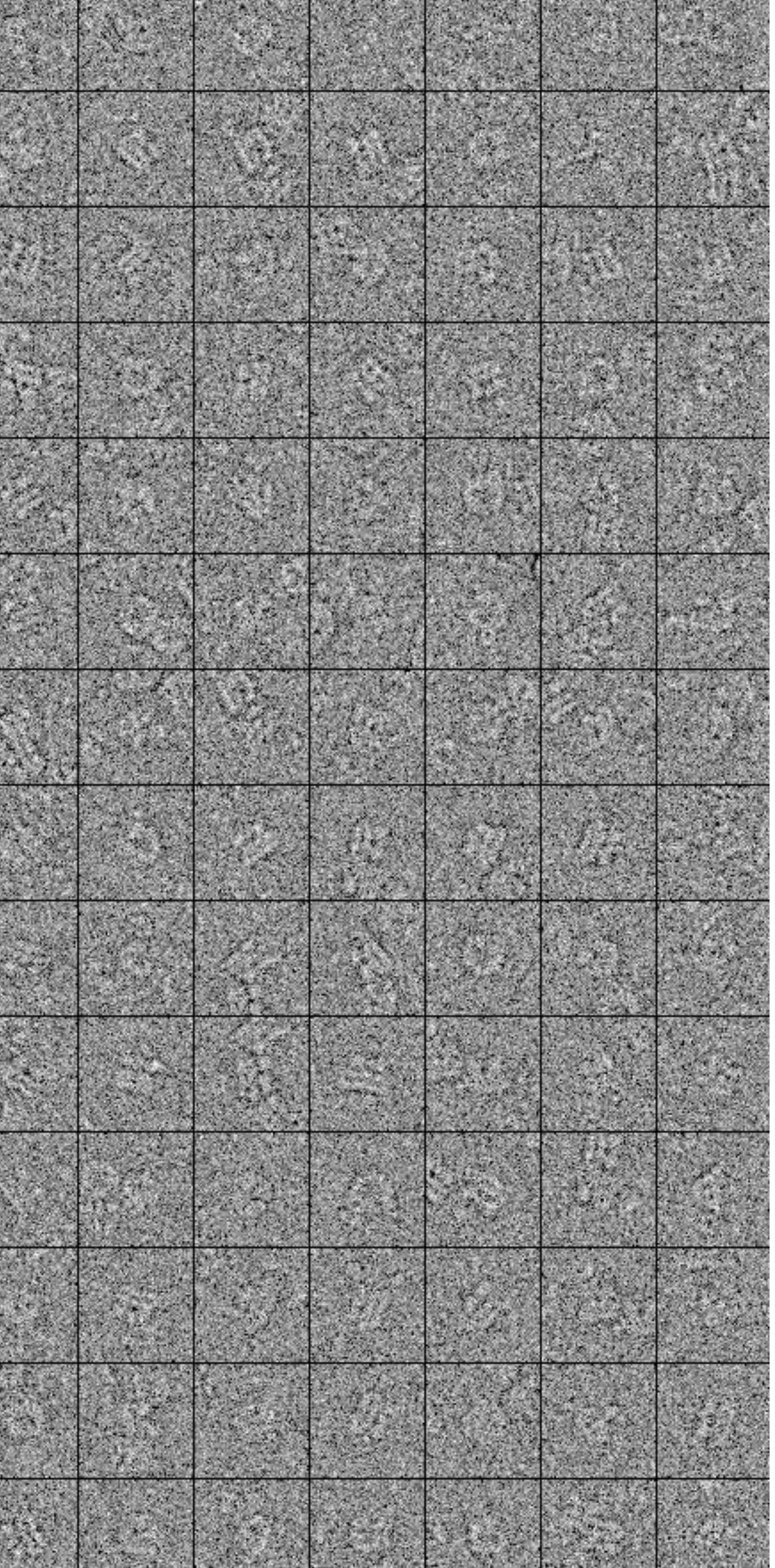
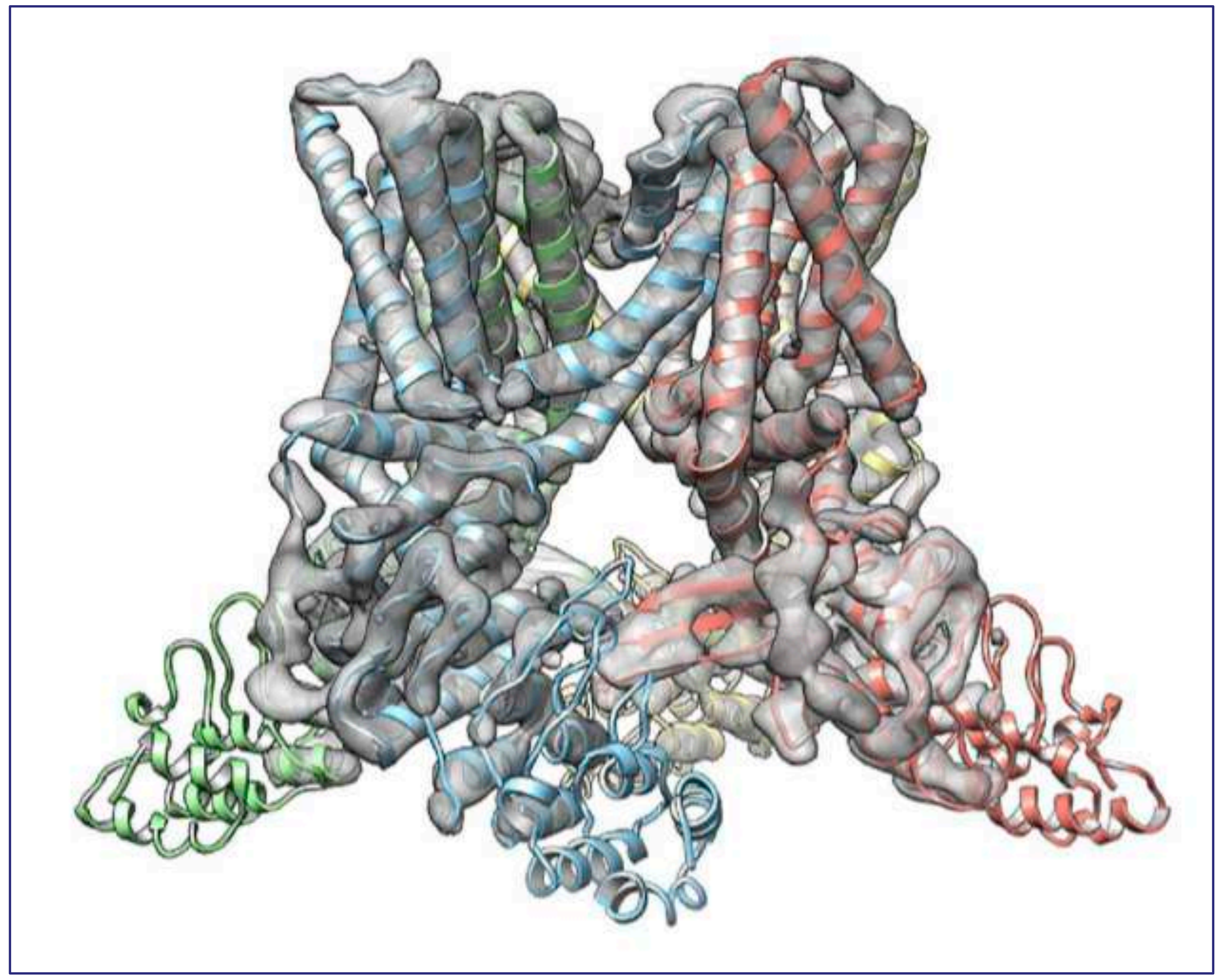
CTF "correction"
Reconstruction
Maximum Likelihood methods

Fred Sigworth
Yale University

National Center for CryoEM Access and Training

Lecture 4b
SPA Short Course
March 2022





3D Reconstruction

CTF “correction”

Single-particle reconstruction

Maximum-likelihood methods

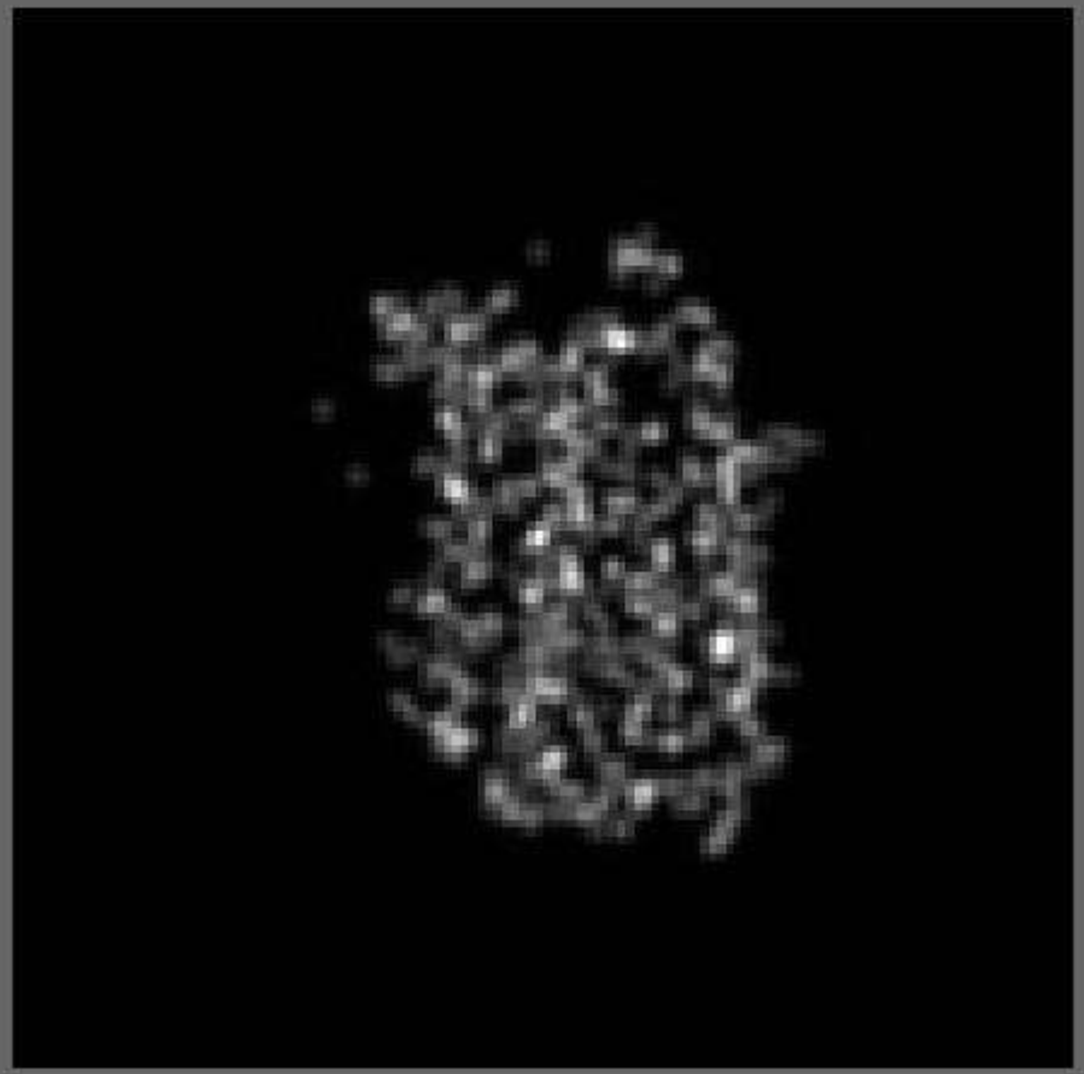
3D Reconstruction

CTF “correction”

Single-particle reconstruction

Maximum-likelihood methods

Every image has a 2D Fourier transform

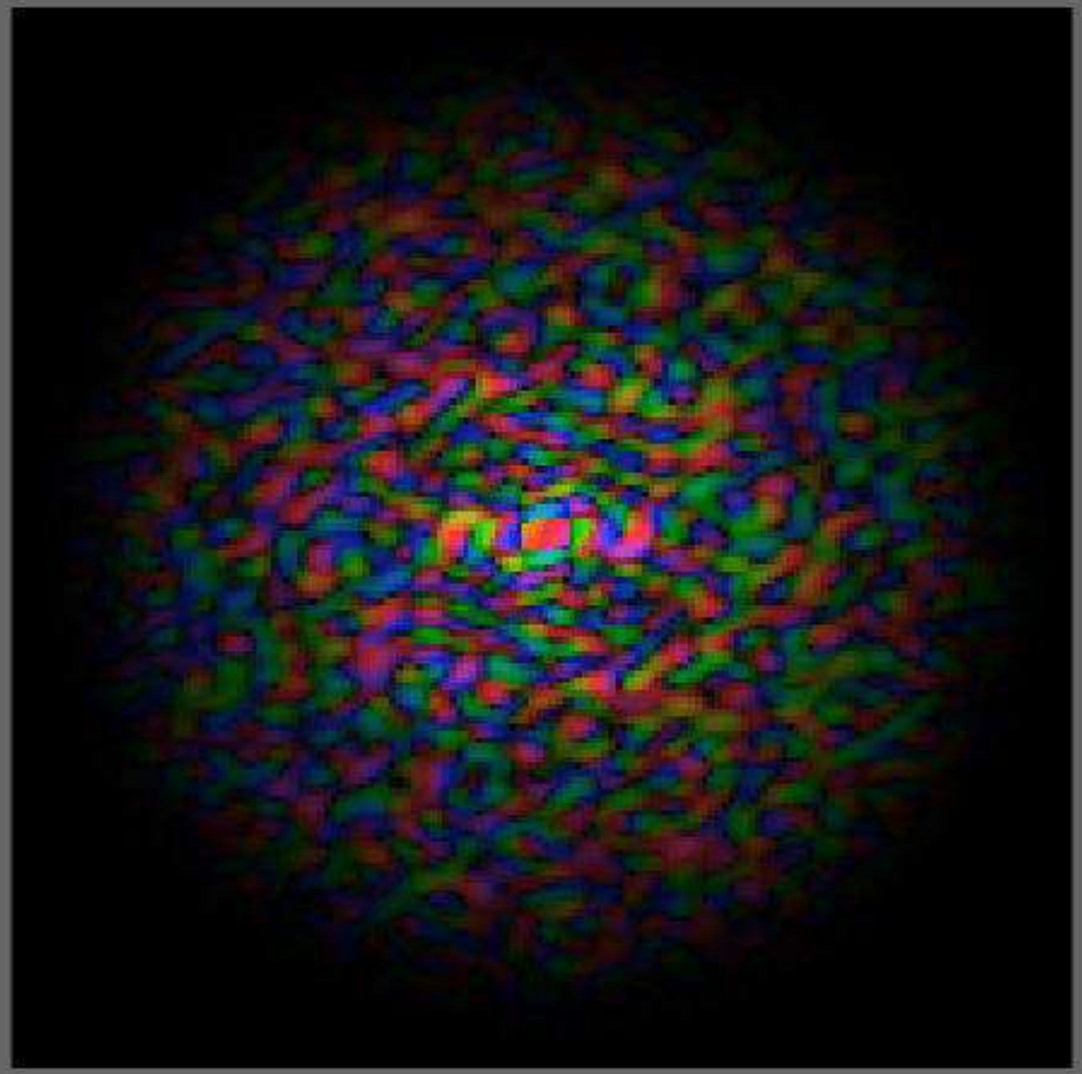


$g(x, y)$

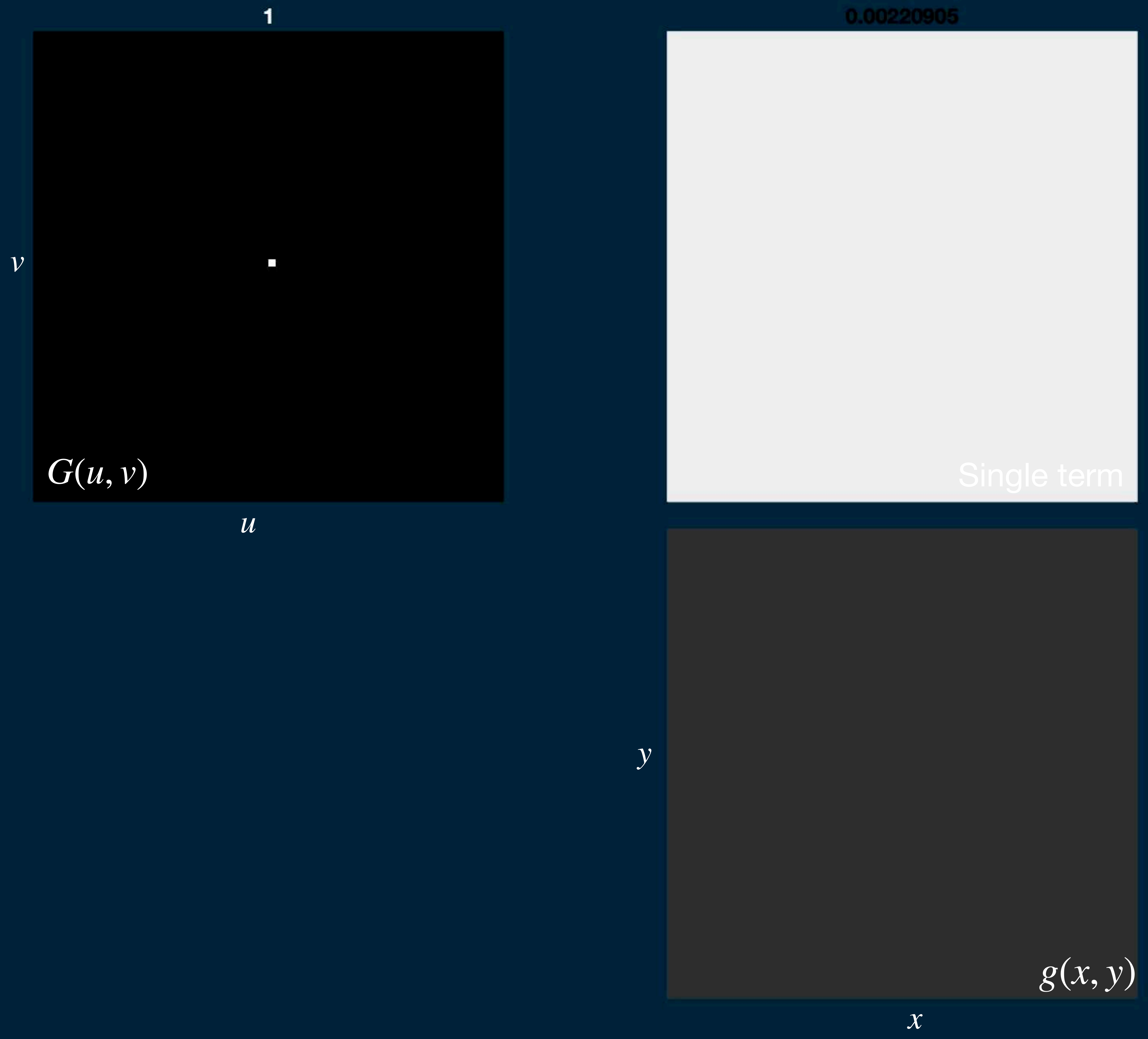
Fourier transform

Inverse FT

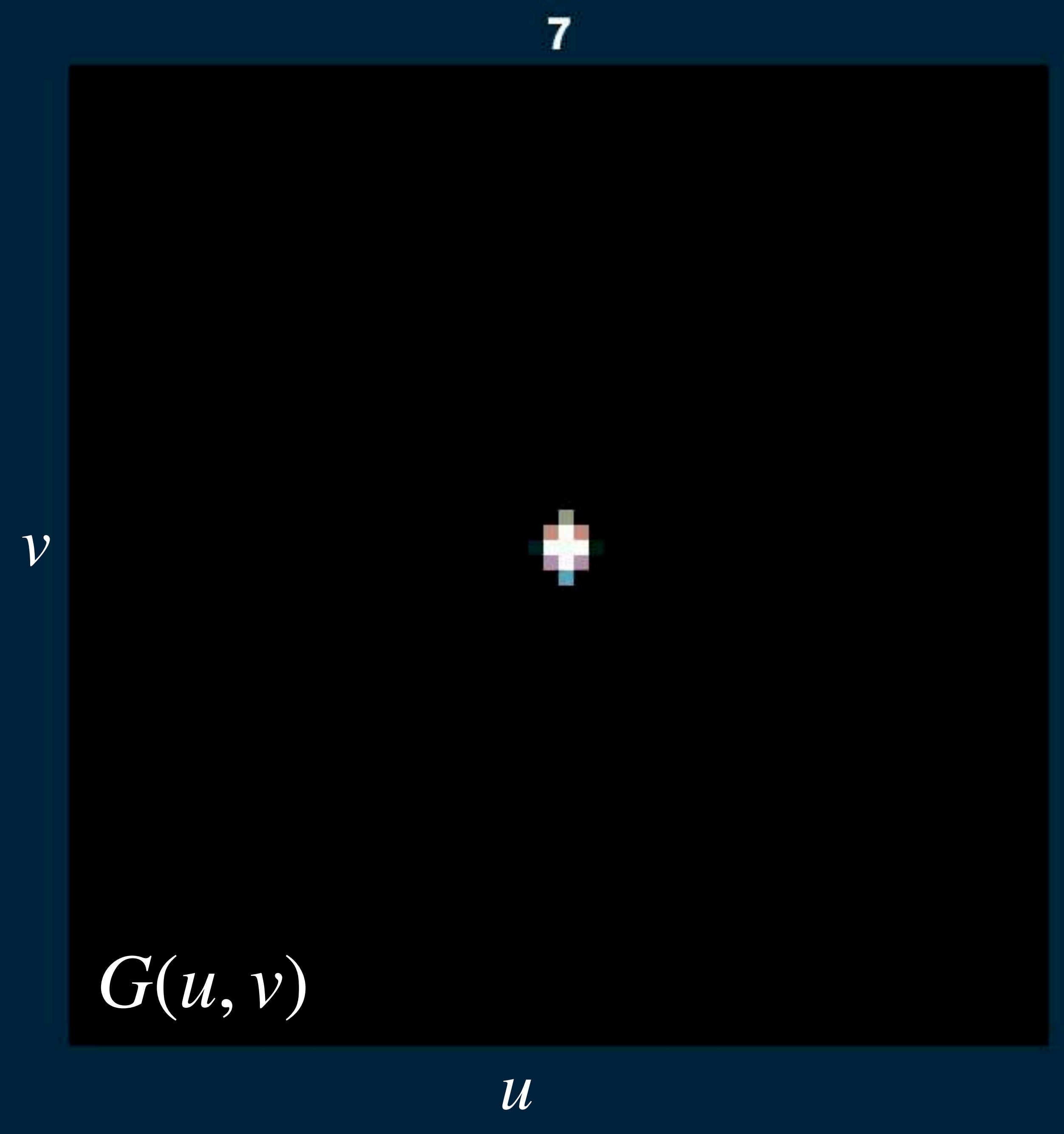
$G(u, v)$

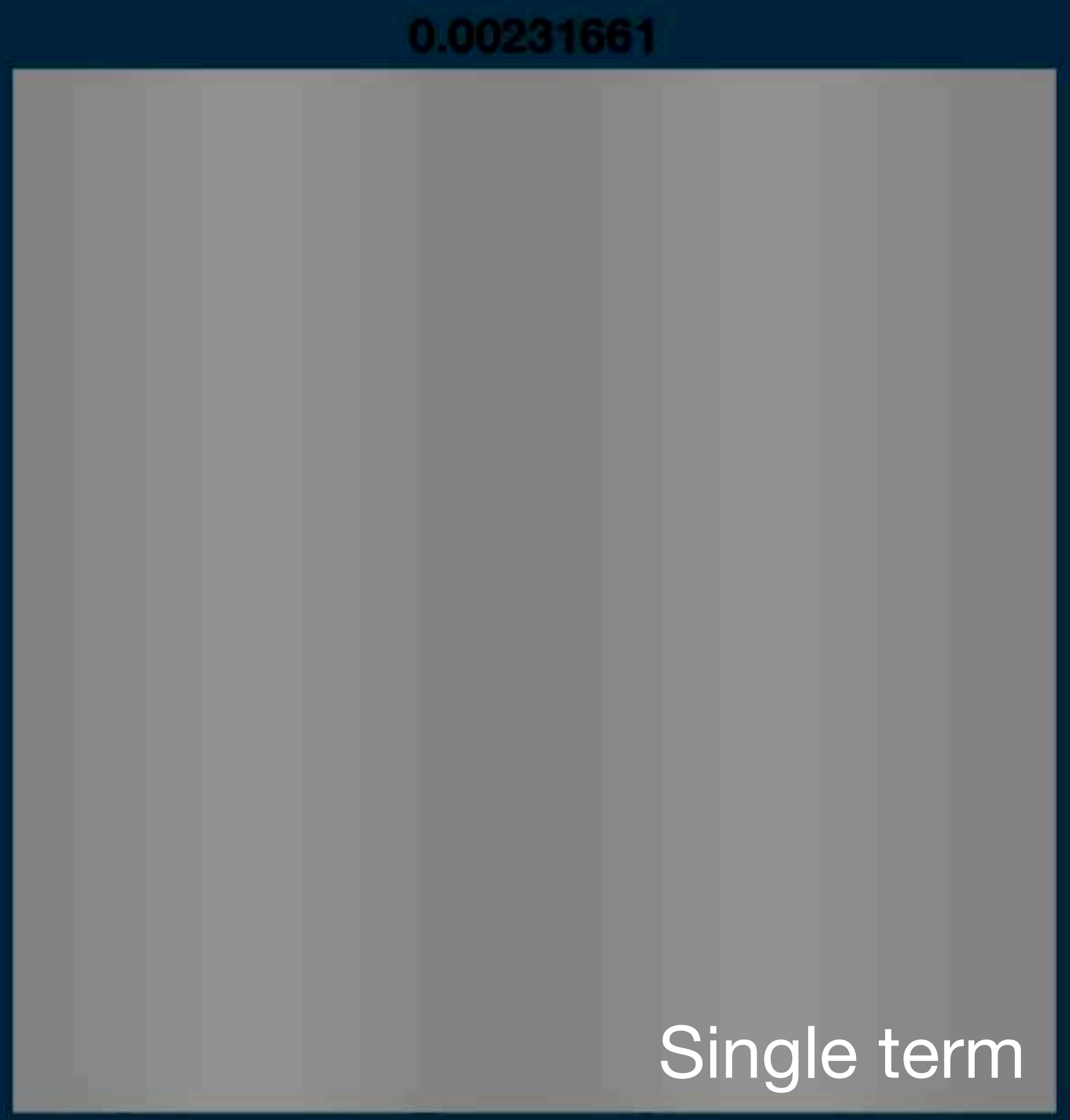


Each point in the FT corresponds to a grating “frequency”



Each point in the FT corresponds to a grating "frequency"





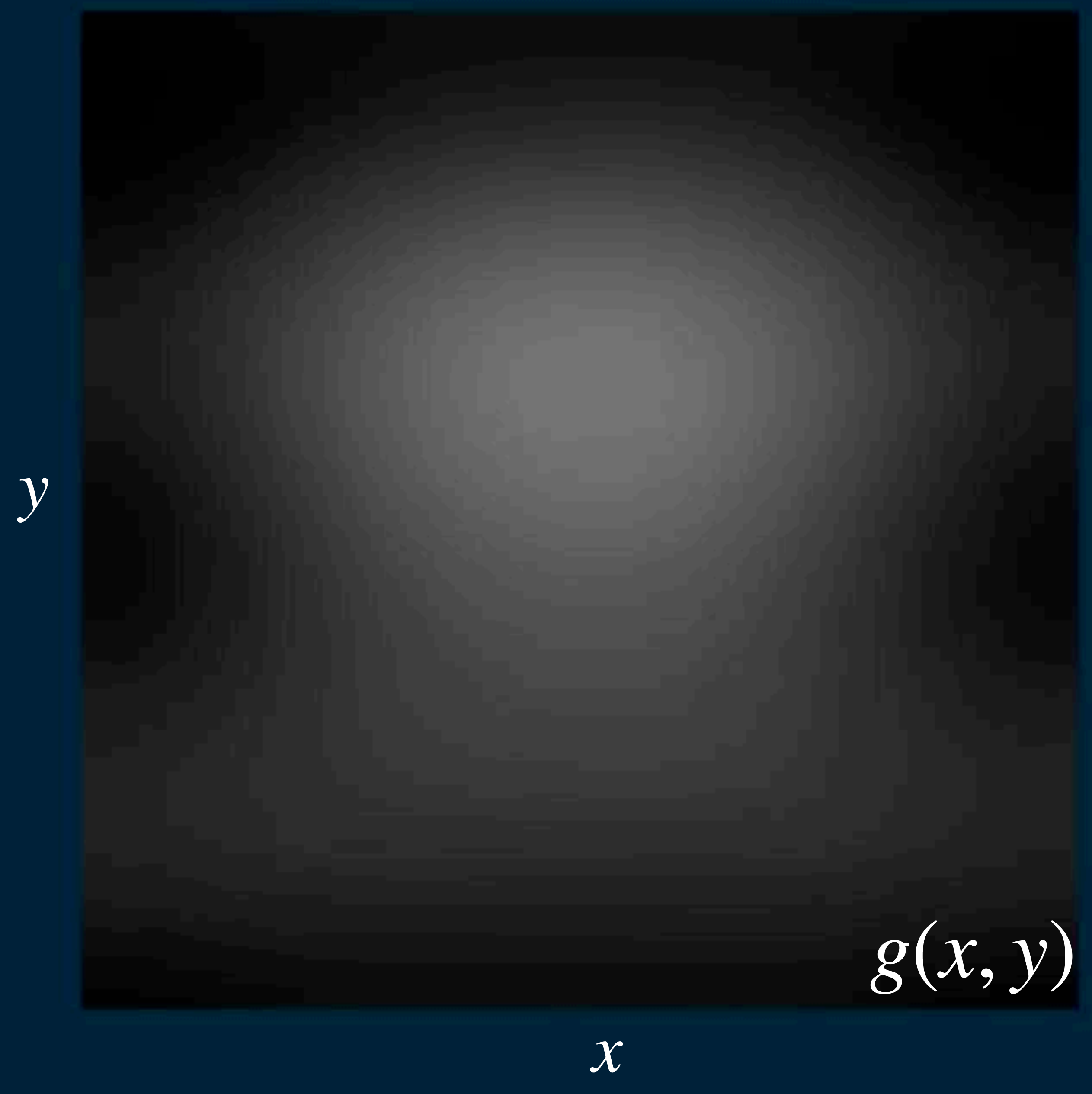
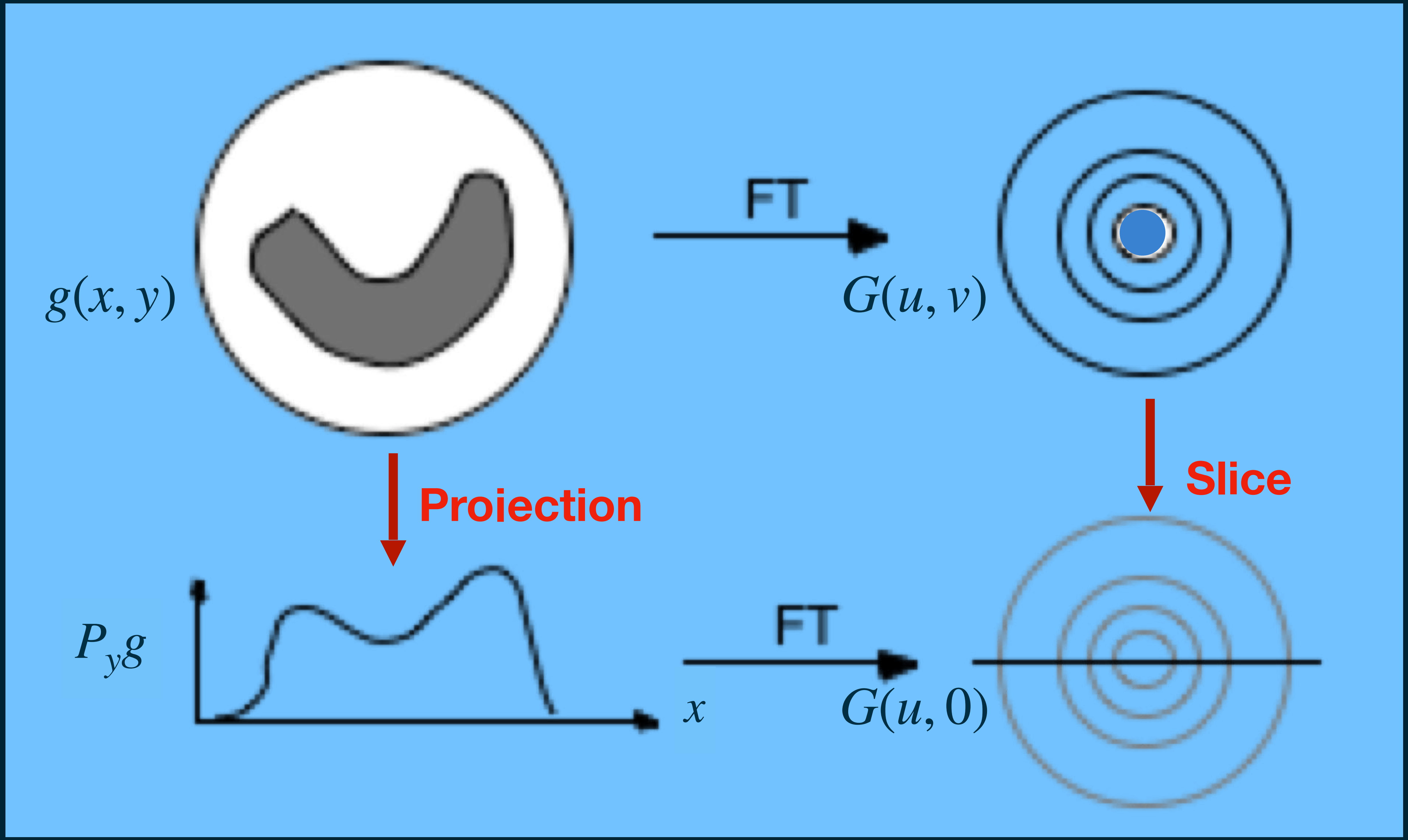


Image processing with Fourier transforms

$g(x, y) \rightarrow G(u, v)$	Fourier Transform
$g * h \rightarrow GH$	Convolution
$g(x', y') \rightarrow G(u', v')$	Rotation
$P_y\, g(x, y) \rightarrow G(u, 0)$	Projection

The Fourier Slice Theorem



2D Fourier Transform

$$G(u, v) = \iint g(x, y) e^{-i2\pi(ux+vy)} dxdy$$

Projection along y

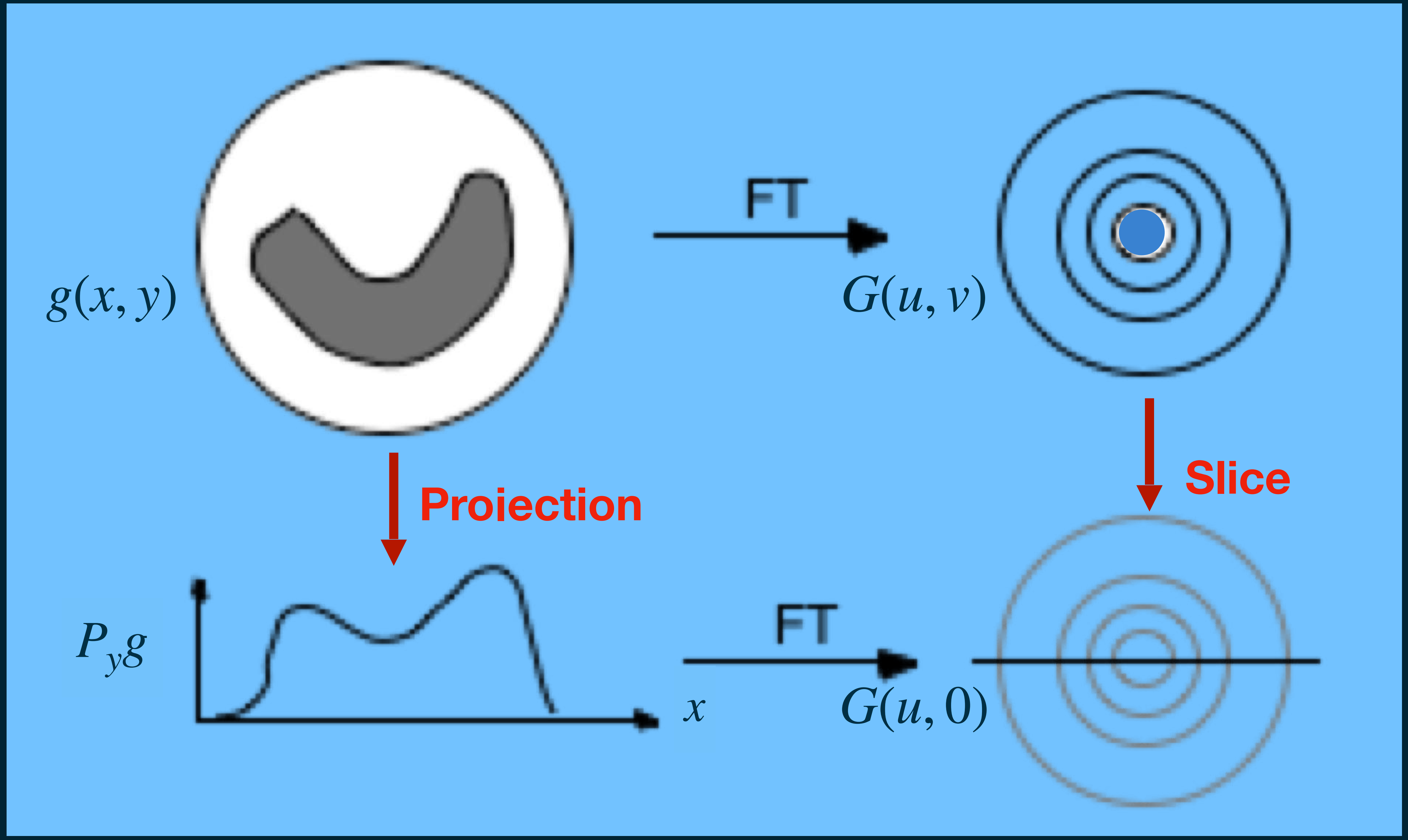
$$P_y g(x, y) = \int g(x, y) dy$$

Values along the u axis

$$G(u, 0) = \int \left(\int g(x, y) dy \right) e^{-i2\pi(ux)} dx$$

$$= \mathcal{F}\{P_y g\}$$

The rotation property allows us to fill in all of $G(u, v)$



2D Fourier Transform

$$G(u, v) = \iint g(x, y) e^{-i2\pi(ux+vy)} dxdy$$

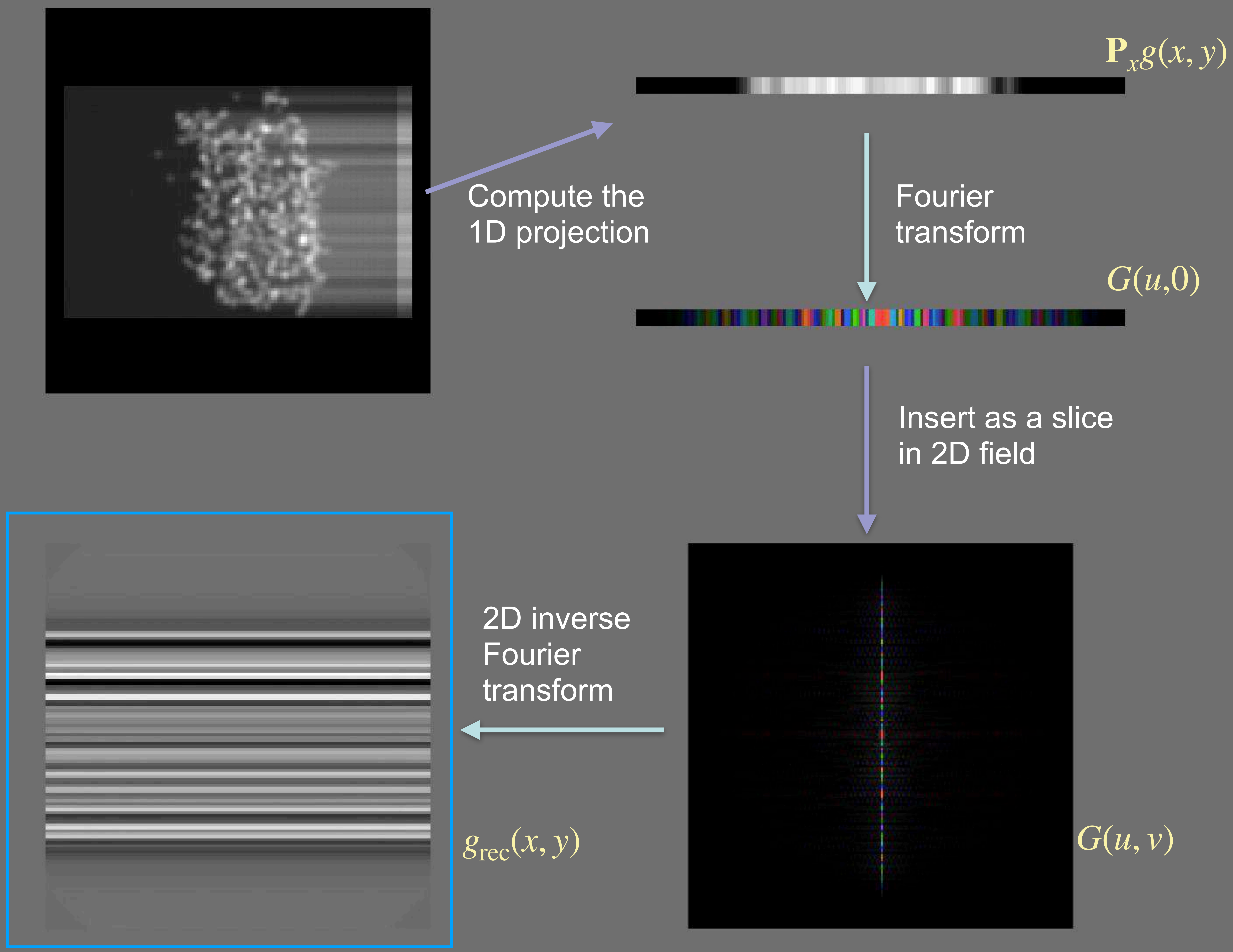
FT using 2D vectors

$$G(\mathbf{u}) = \int\int g(\mathbf{x}) e^{-i2\pi(\mathbf{u}\cdot\mathbf{x})} d^2\mathbf{x}$$

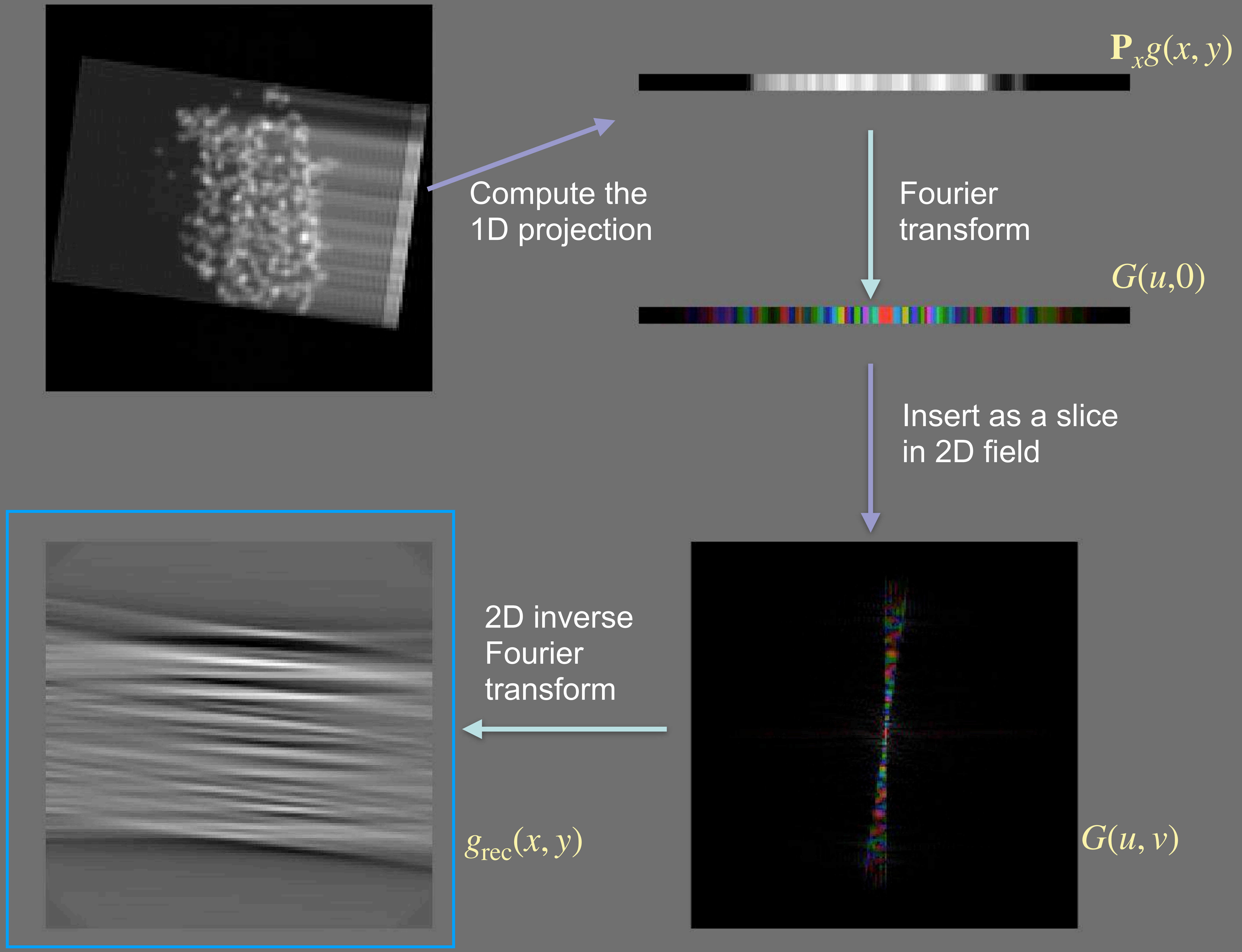
Dot-product is invariant under rotations!

The rotation property says:
If we can collect projections from all directions, we can construct all of $G(u, v)$

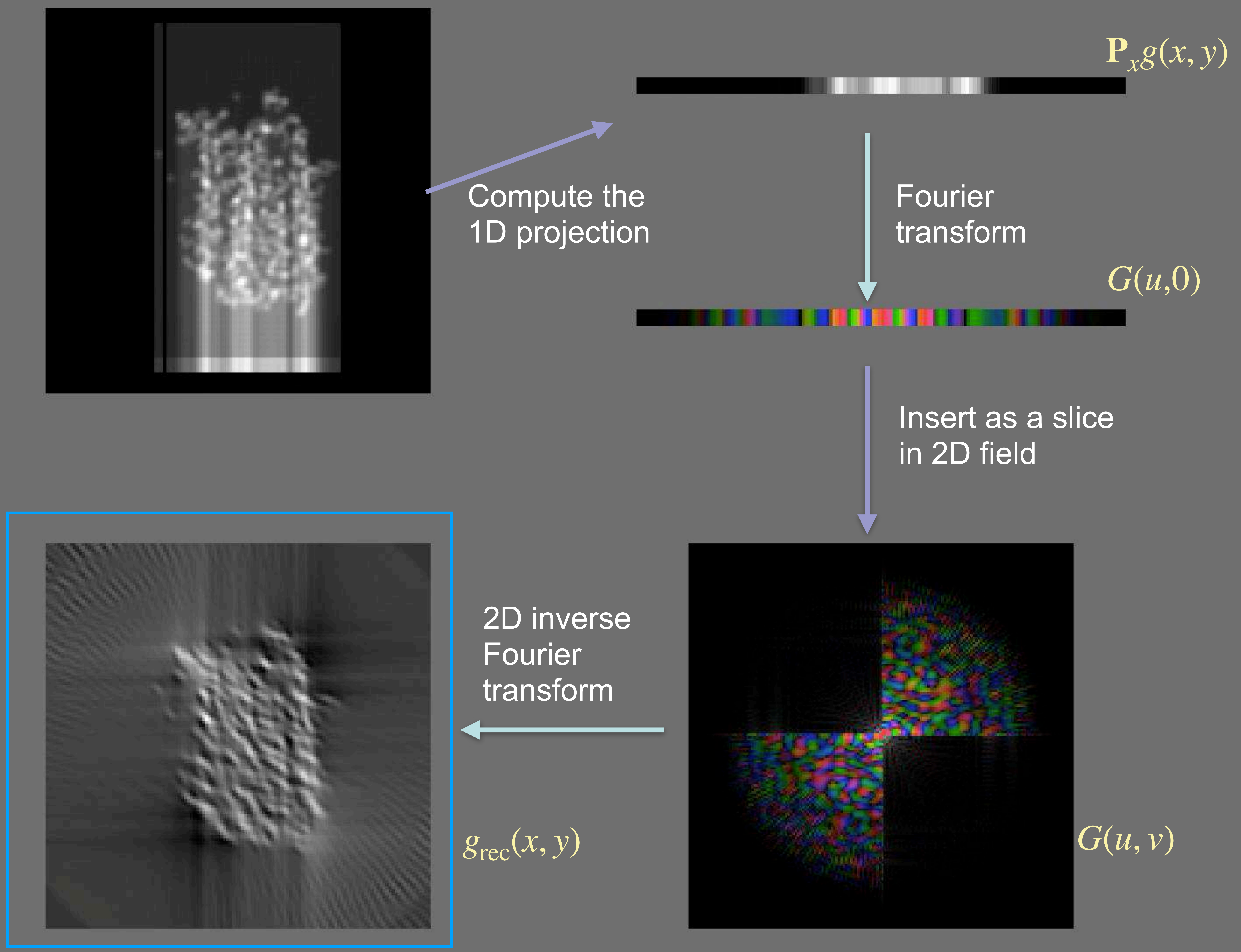
Reconstruction using the Fourier Slice Theorem



Reconstruction using the Fourier Slice Theorem



Reconstruction using the Fourier Slice Theorem



3D Reconstruction

CTF “correction”

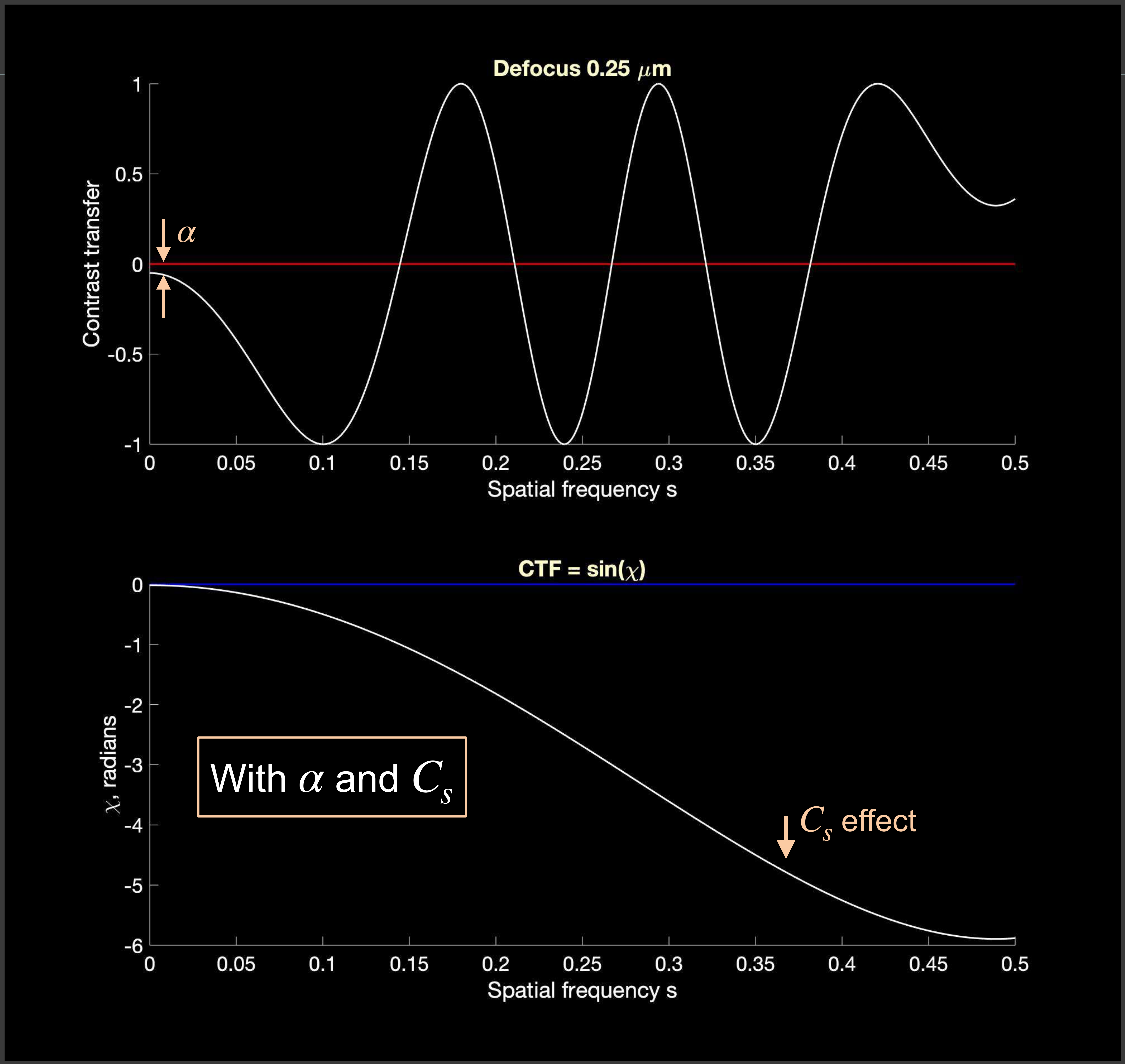
Single-particle reconstruction

Maximum-likelihood methods

Dealing with the contrast-transfer function

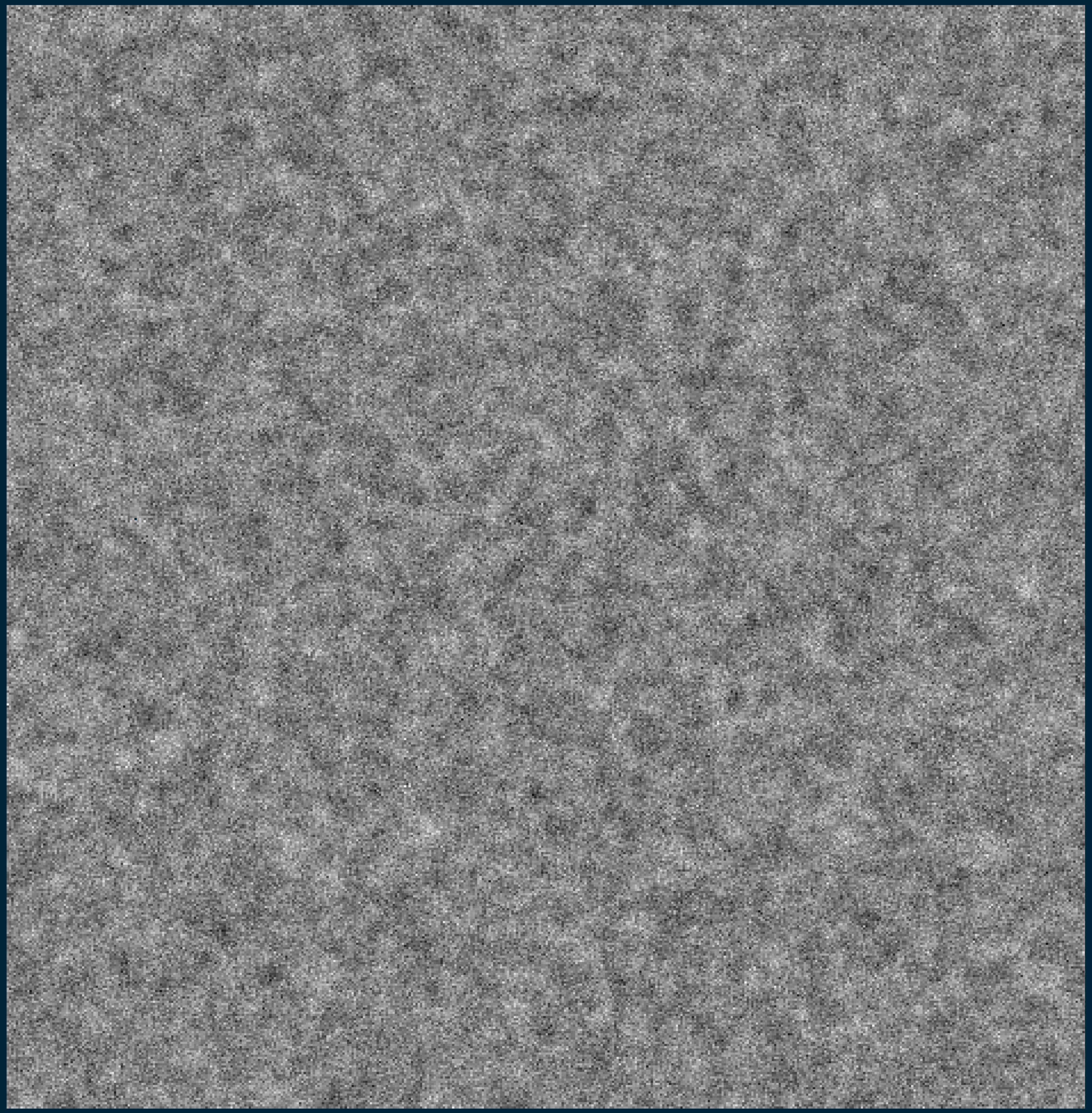
The contrast transfer function is given by

$$\mathrm{CTF} = \sin\left(-\pi\lambda\delta f^2 + \frac{\pi}{2}C_s\lambda^3 f^4 - \alpha\right)$$



The power spectrum describes the magnitude of Fourier components in an image

Image X



Power spectrum $|\mathscr{F}\{X\}|^2$

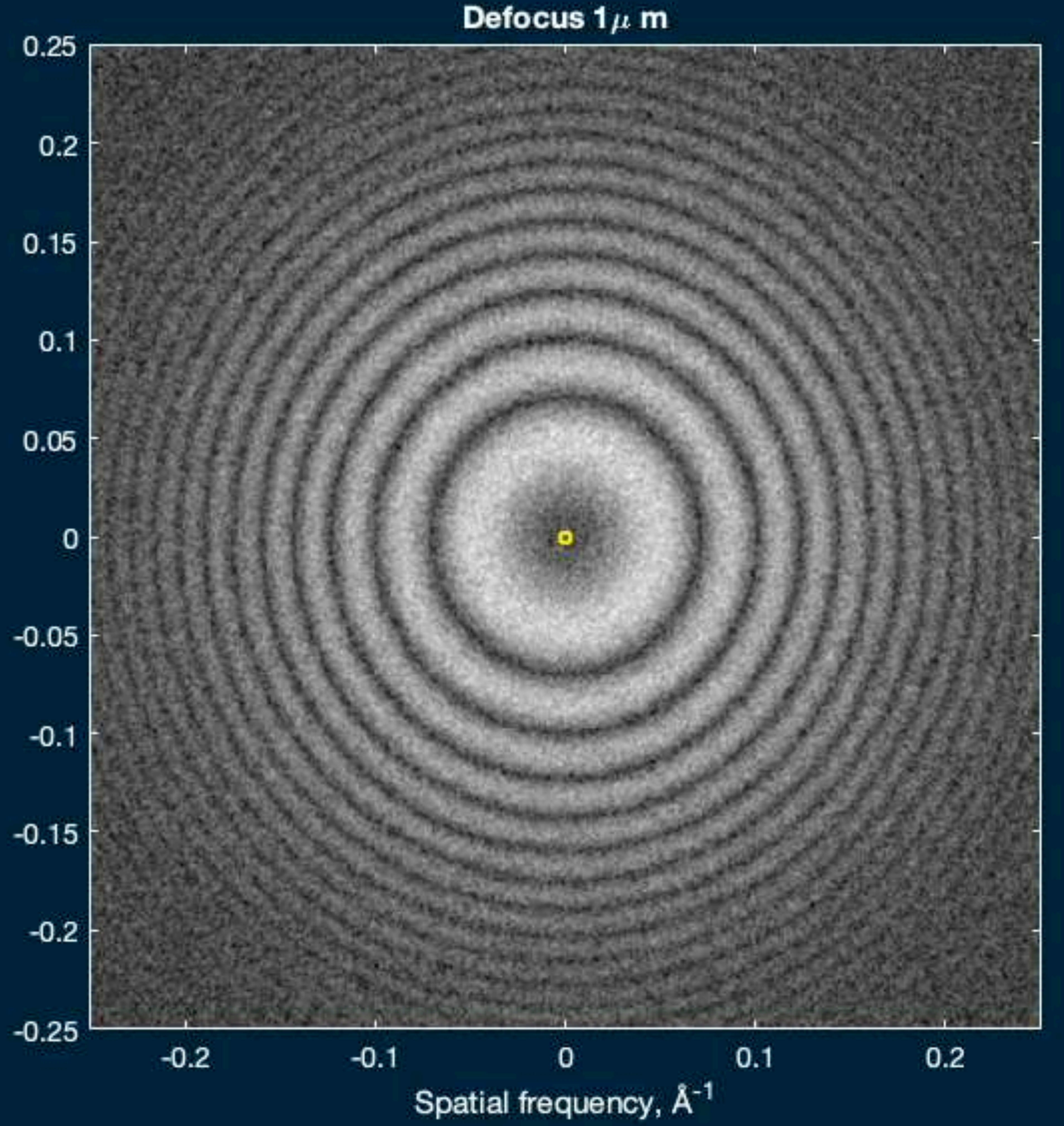


Image processing with Fourier transforms

$g(x, y) \rightarrow G(u, v)$	Fourier Transform
$g * h \rightarrow GH$	Convolution
$g(x', y') \rightarrow G(u', v')$	Rotation
$P_y\, g(x, y) \rightarrow G(u,0)$	Projection

Modeling the CTF effect on an image

Model of an image

$$X = CA + N$$

Can we do the deconvolution

$$\hat{A} \approx X/C \text{ ??}$$

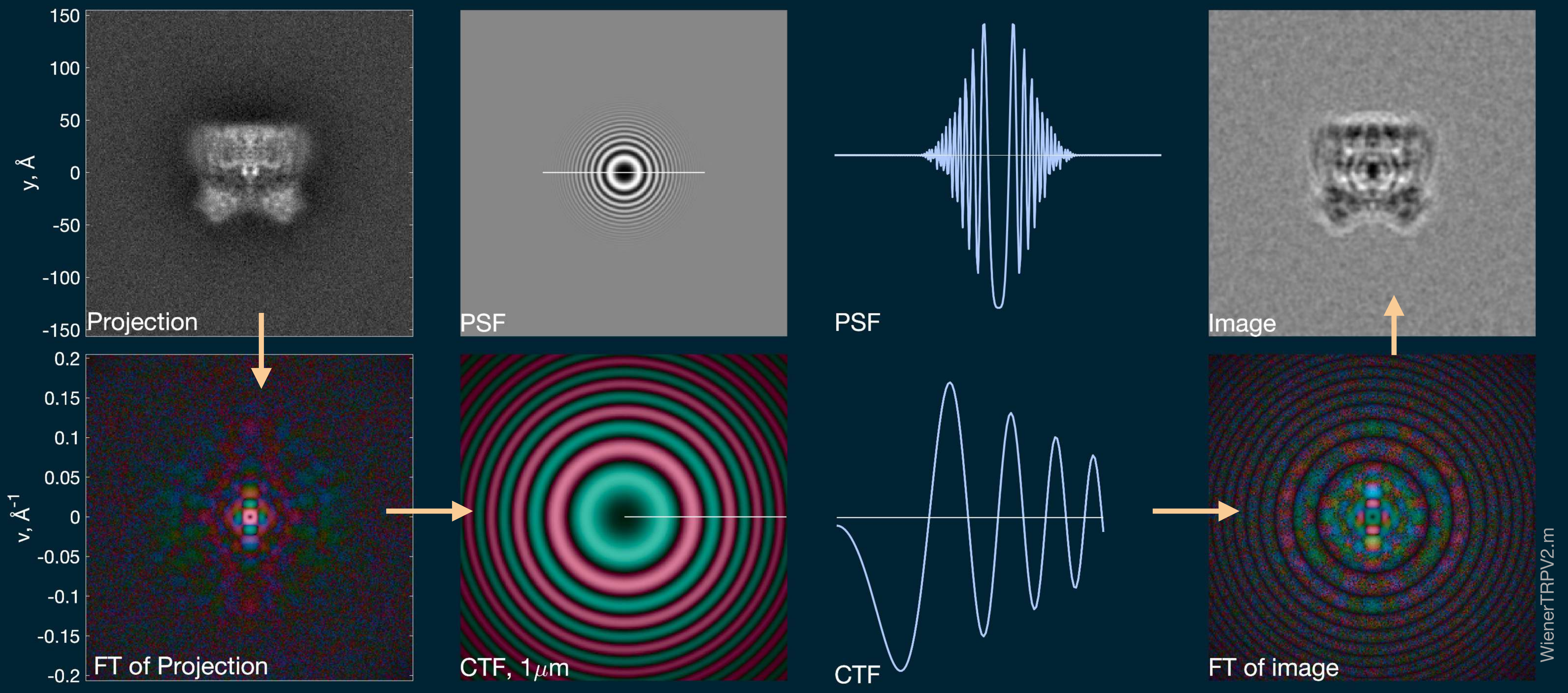
A	"true" image
C	contrast-transfer function
N	noise image

We can interpret *C* as either the CTF operator (*x*,*y* space), or just the multiplicative CTF factor (*u*,*v* space)

Modeling the CTF effect on an image

Model of an image

$X = CA + N$



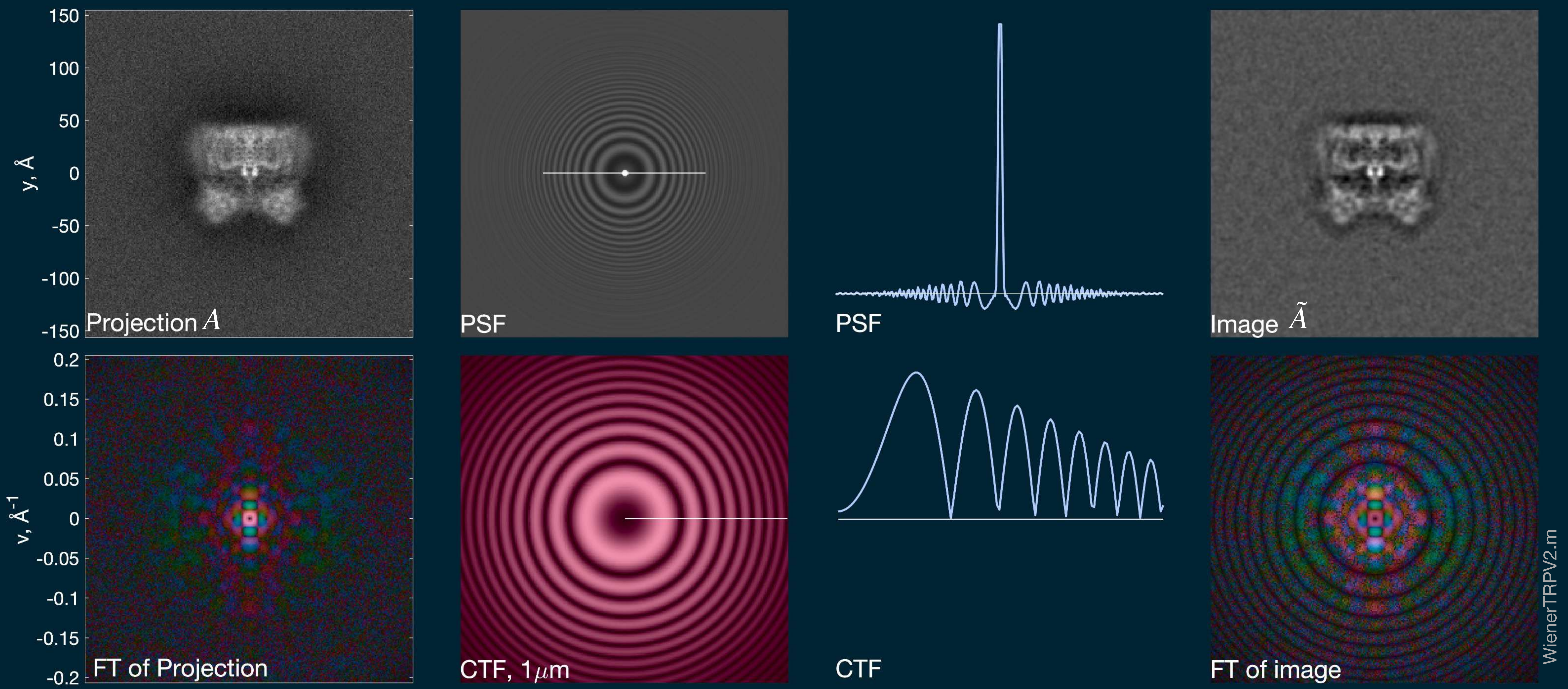
Can we do the deconvolution

$\hat{A} = X/C$??

How to undo the CTF effects?

1. **Phase flipping**

$$\hat{A} = \mathrm{sgn}(C)X$$



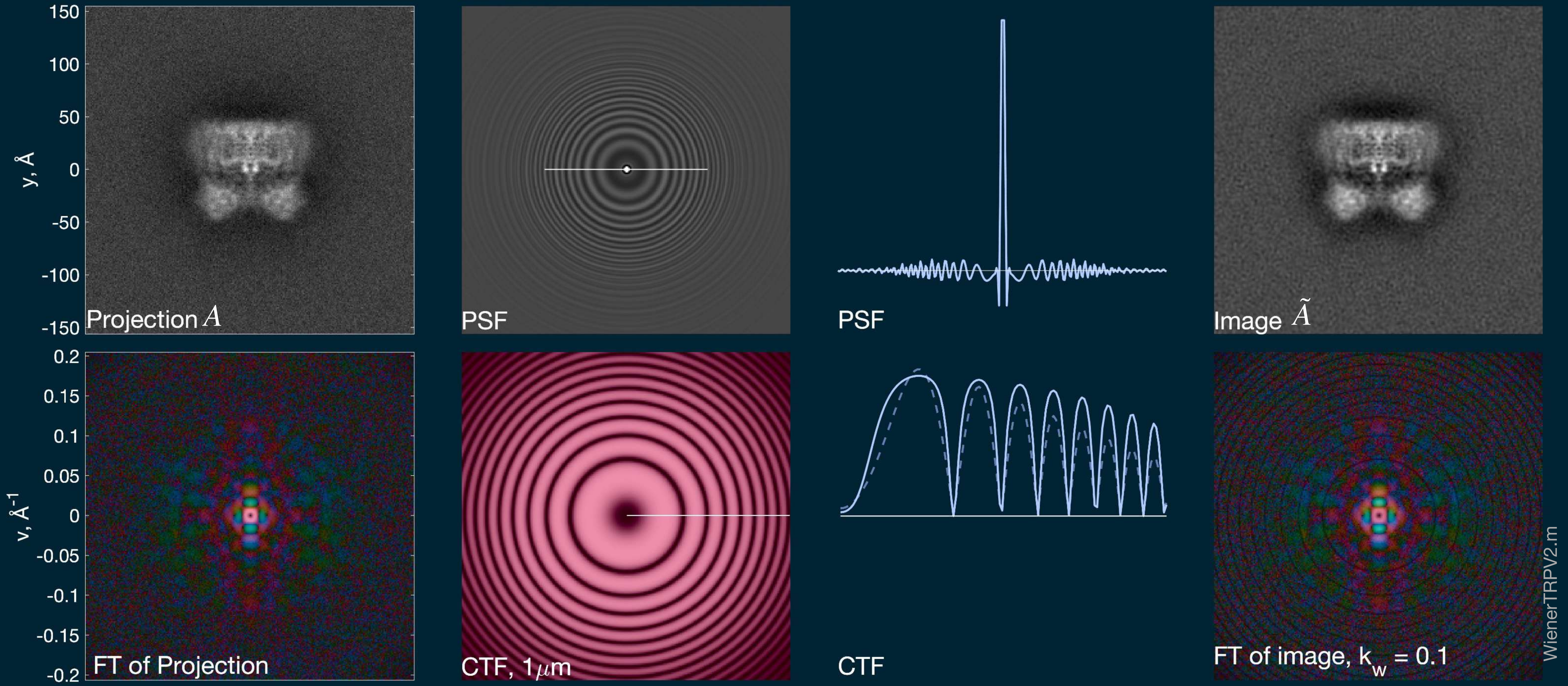
How to undo the CTF effects?

1. **Phase flipping**

$$\hat{A} = \mathrm{sgn}(C)X$$

2. **Wiener filter**

$$\hat{A} = \frac{CX}{C^2 + k}$$



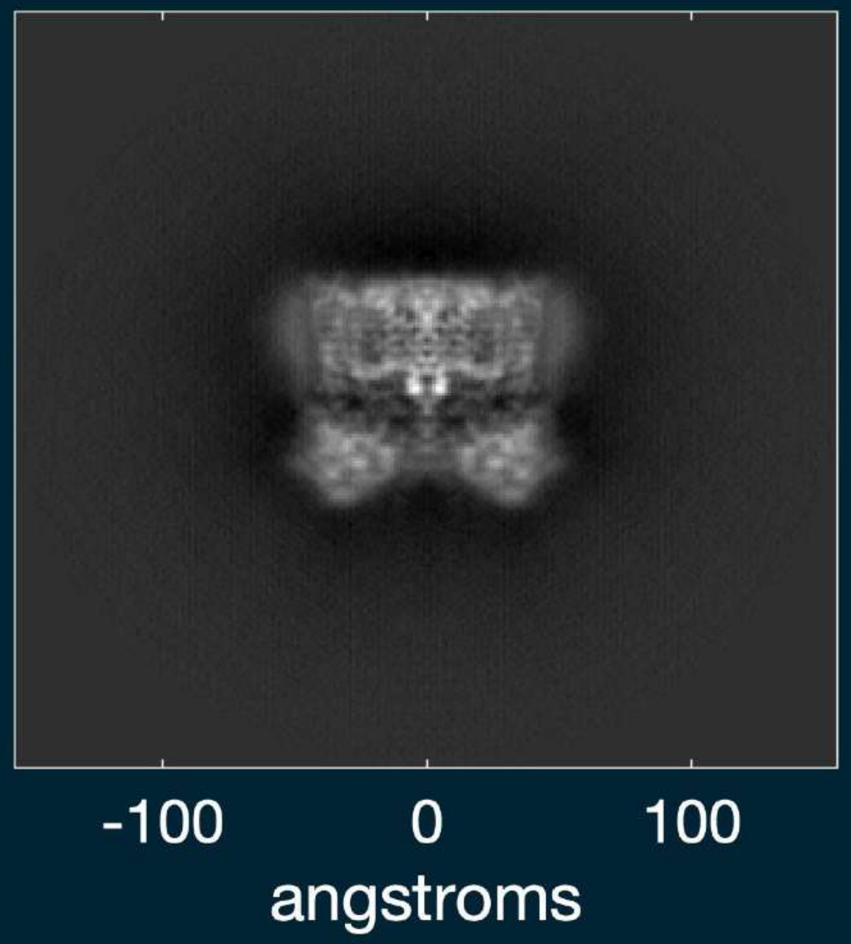
How to undo the CTF effects in noisy images?

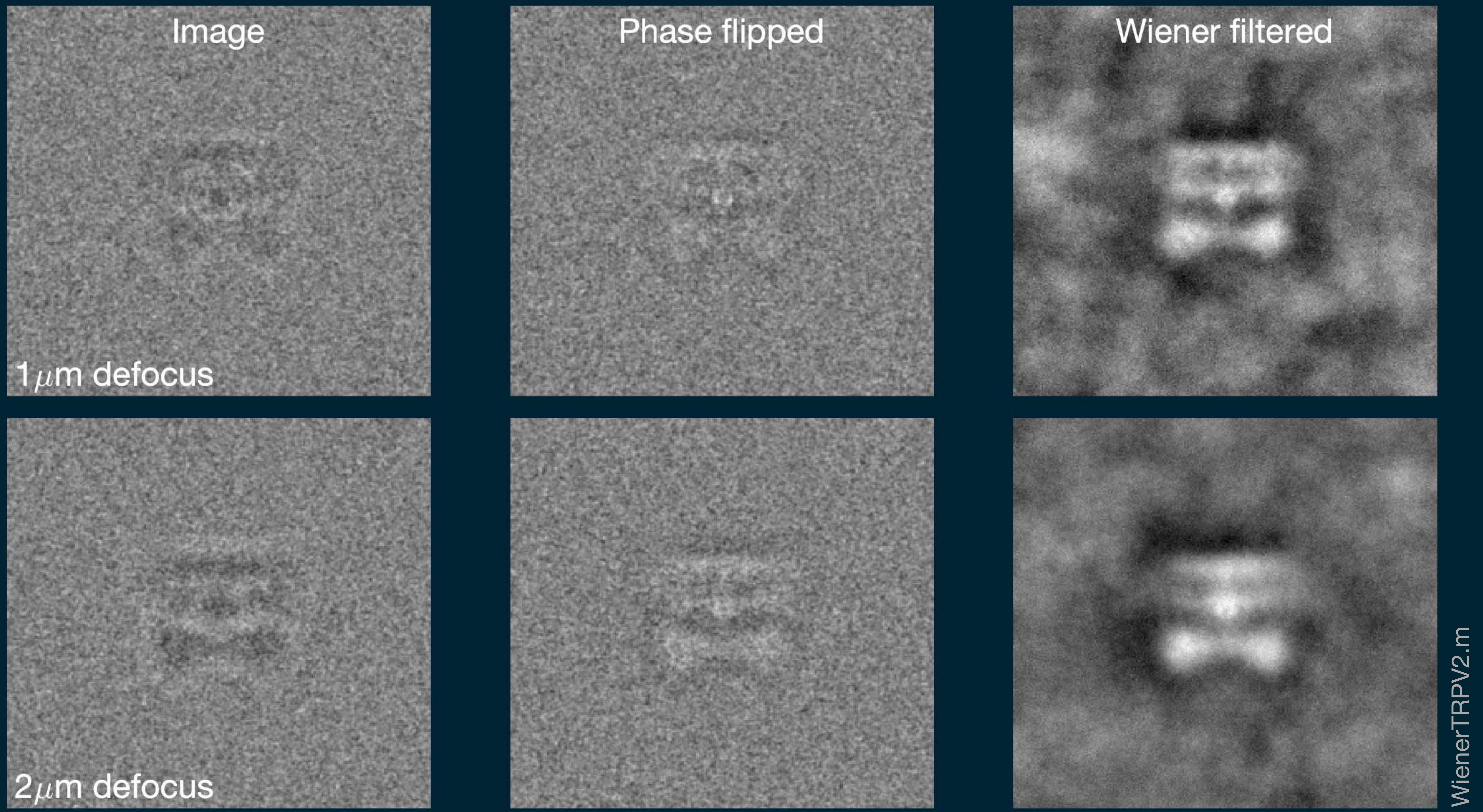
1. **Phase flipping**

$$\hat{A} = \text{sgn}(C)X$$

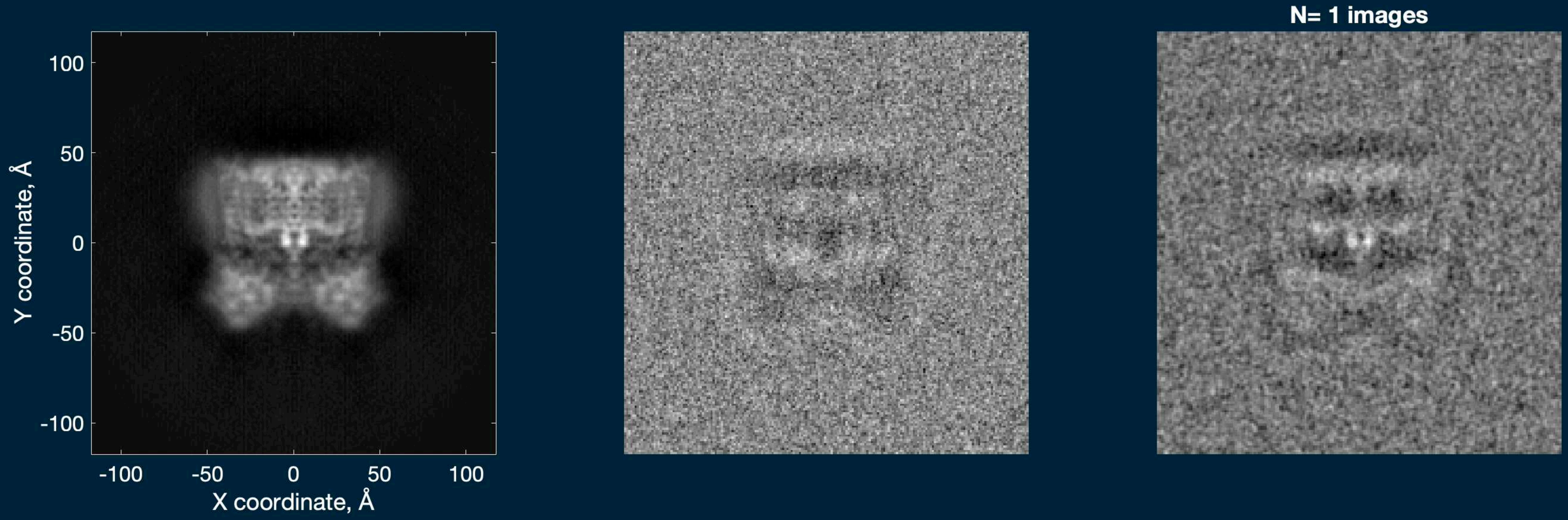
2. **Wiener filter**

$$\hat{A} = \frac{CX}{C^2 + k}$$





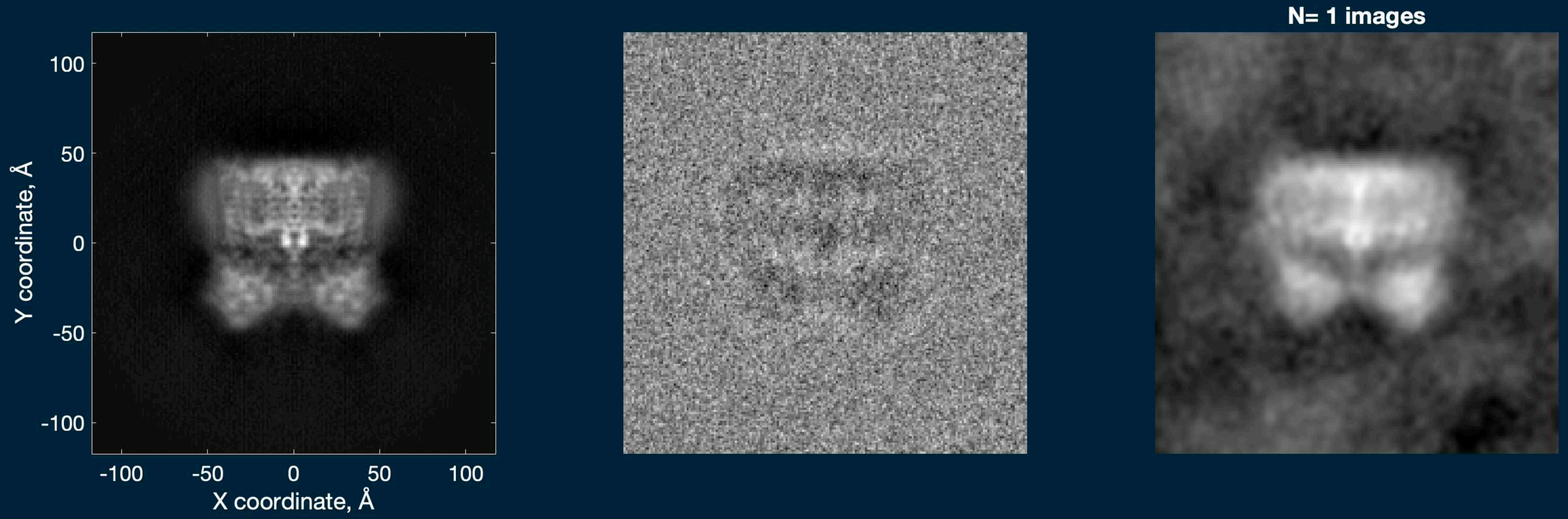
How to undo the CTF effects in noisy images?



3. Wiener from multiple images

$$\hat{A} = \frac{\sum_i^N C_i X_i}{k + \sum_i^N C_i^2}$$

How to undo the CTF effects in noisy images?



3. Wiener from multiple images

$$\hat{A} = \frac{\sum_i^N C_i X_i}{k_w(s) + \sum_i^N C_i^2}$$

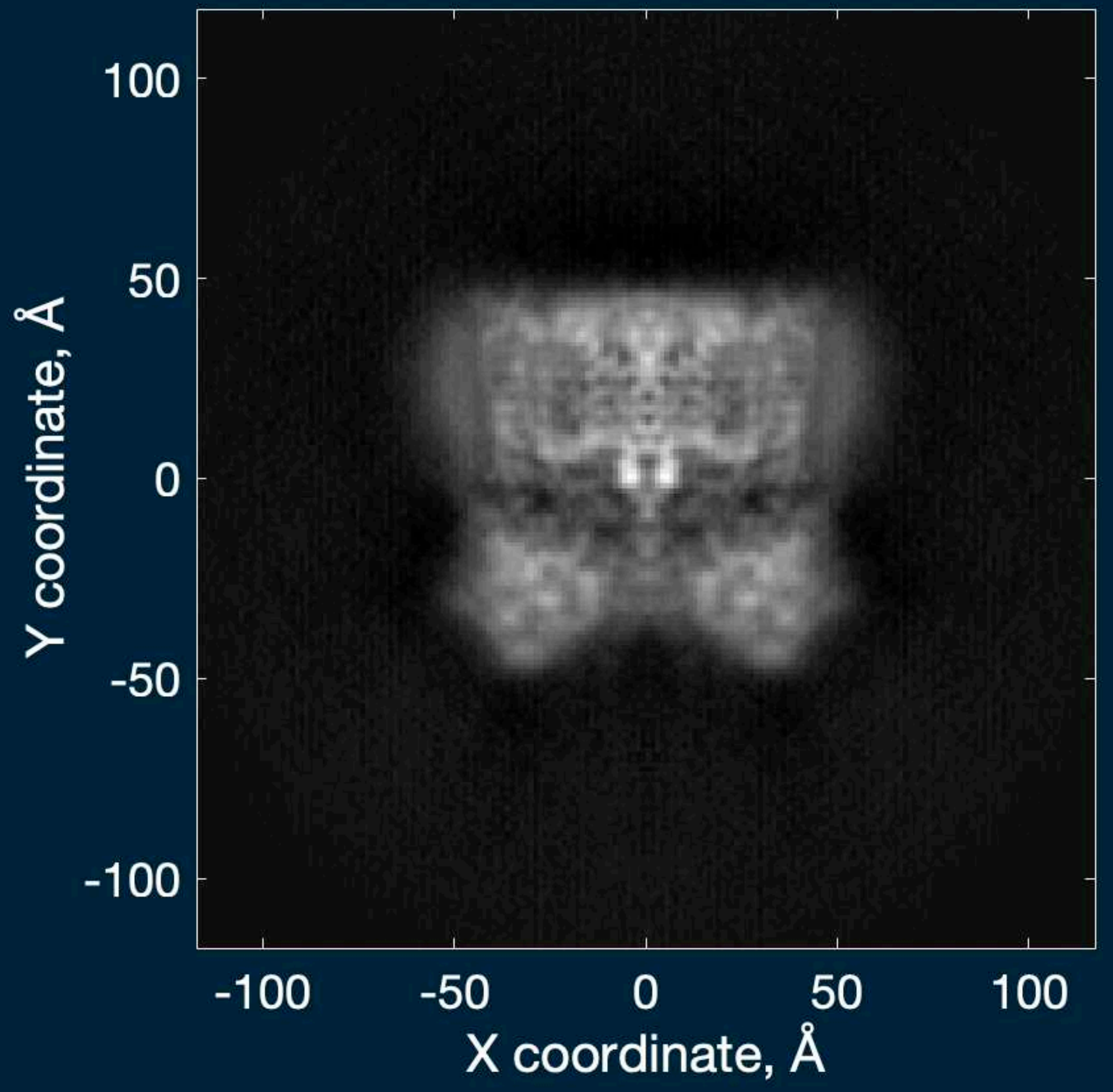
$$k_w(s) = 1/\mathrm{SNR} = \frac{\|N\|^2}{\|A\|^2}$$

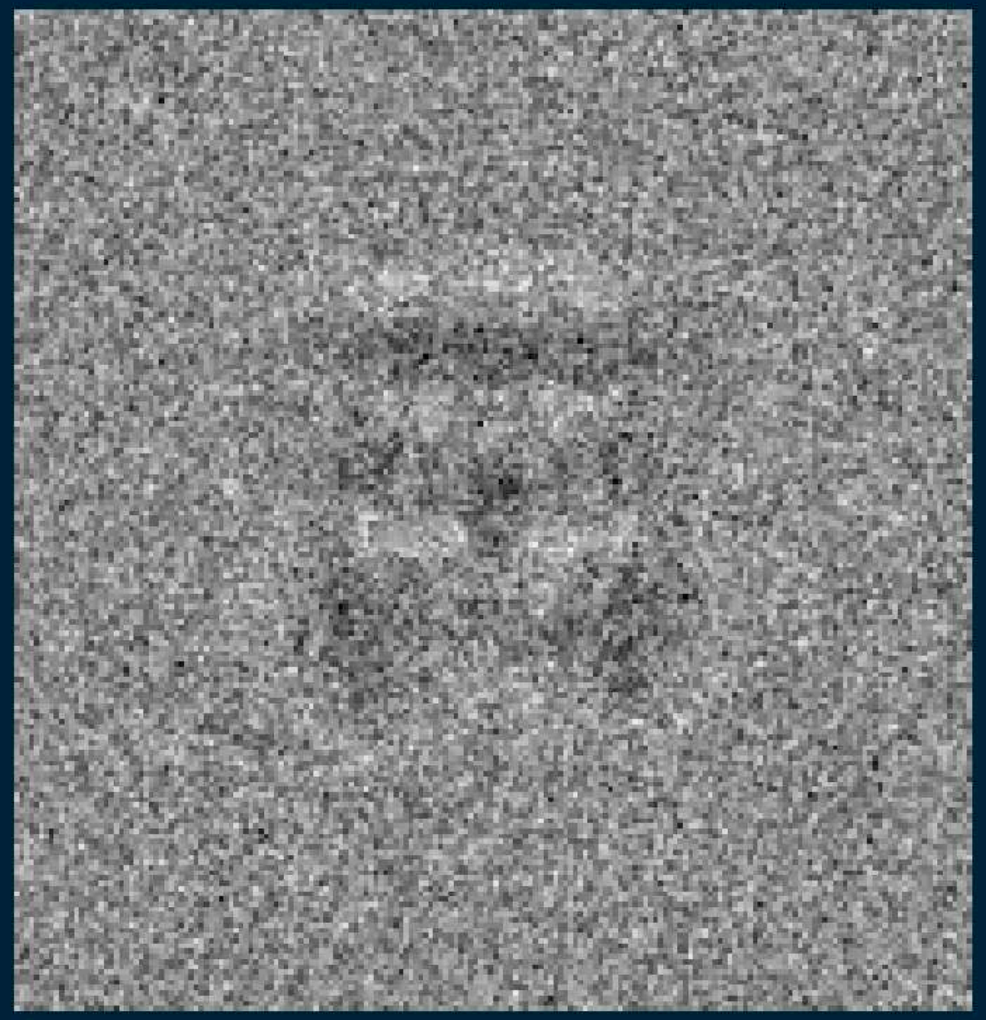
Image restoration when spectral SNR is known

Restoration
from multiple images

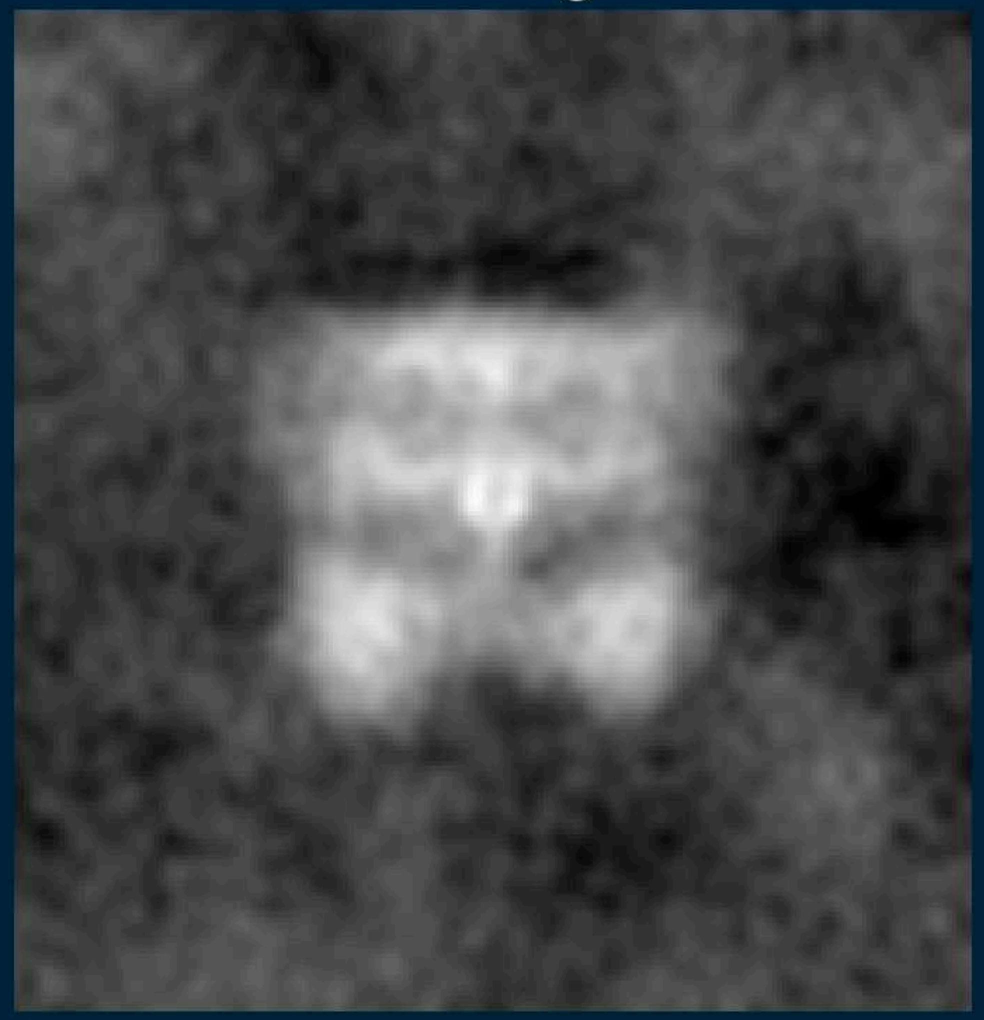
$$\hat{A} = \frac{\sum_i^N C_i X_i}{k_w(s) + \sum_i^N C_i^2}$$

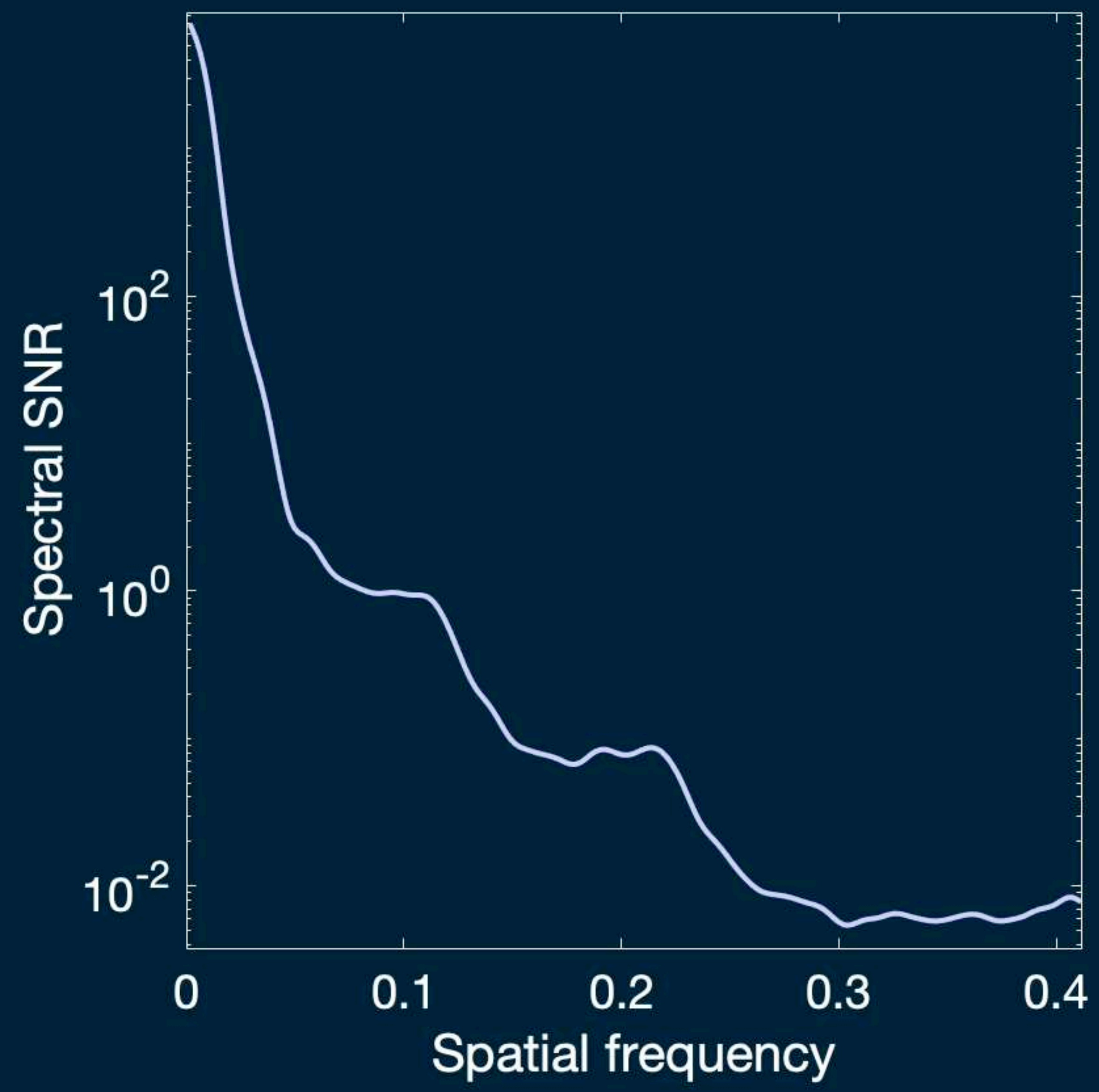
The defocus varies
to fill in CTF zeros





N= 1 images





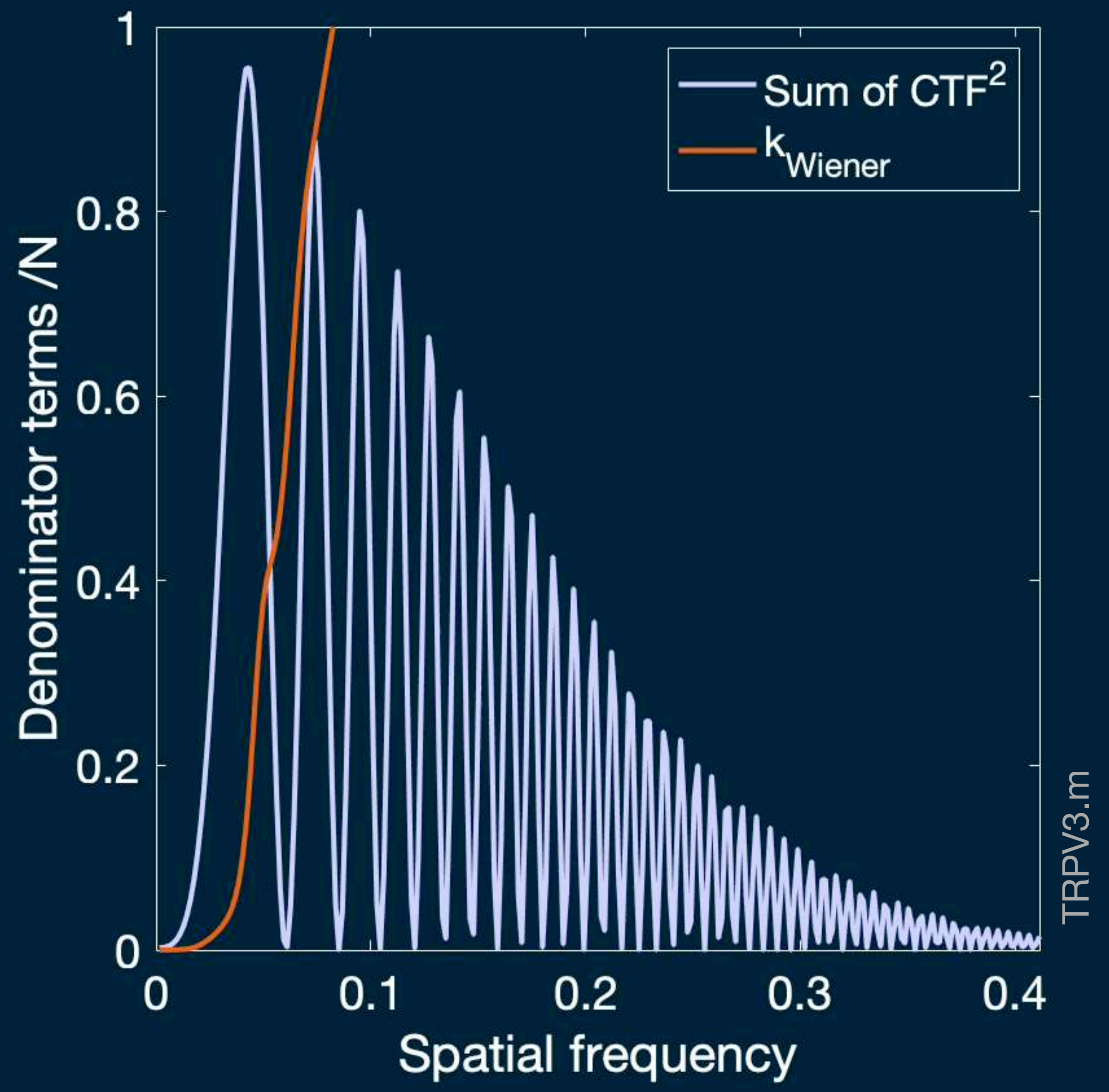
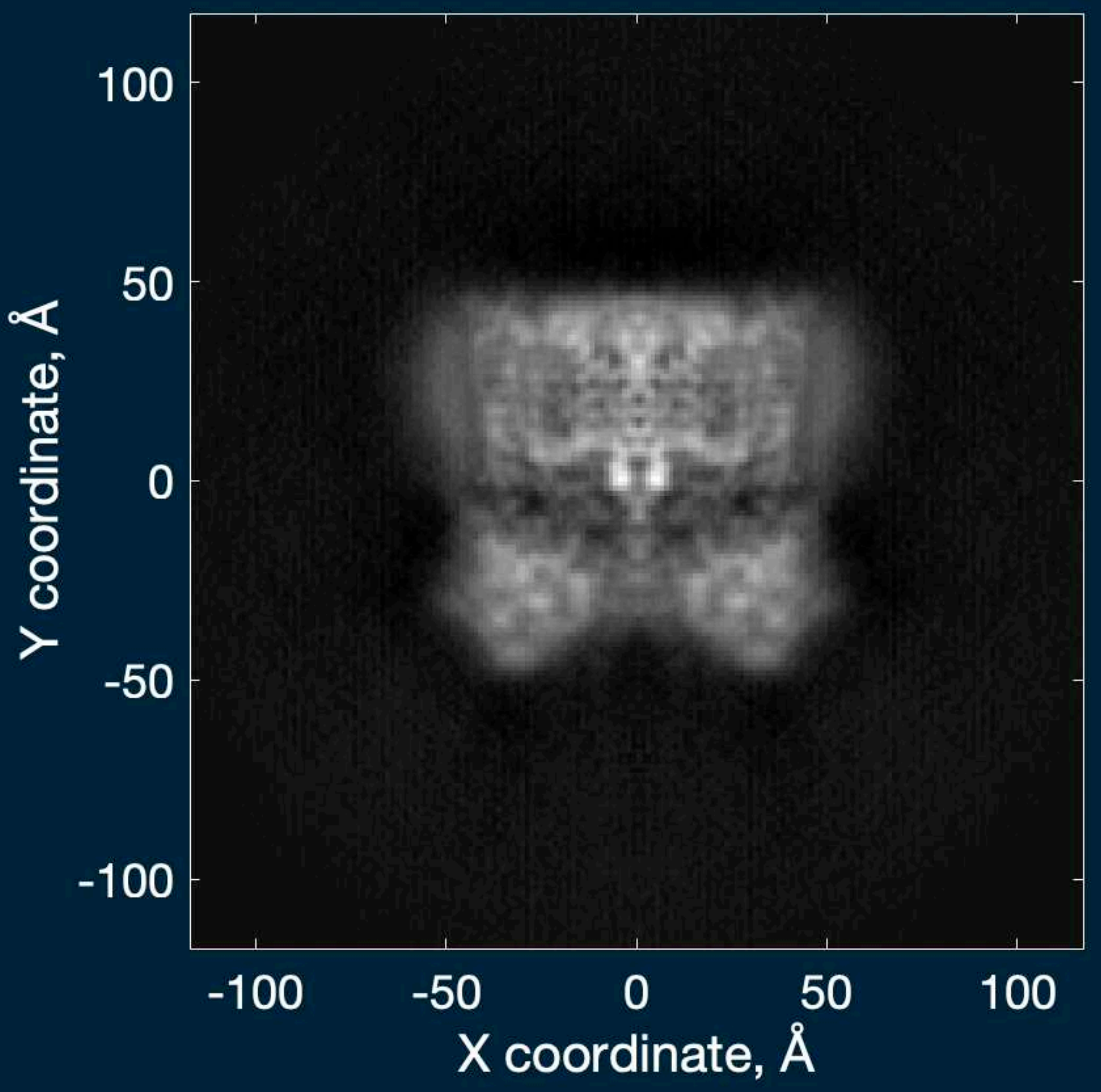


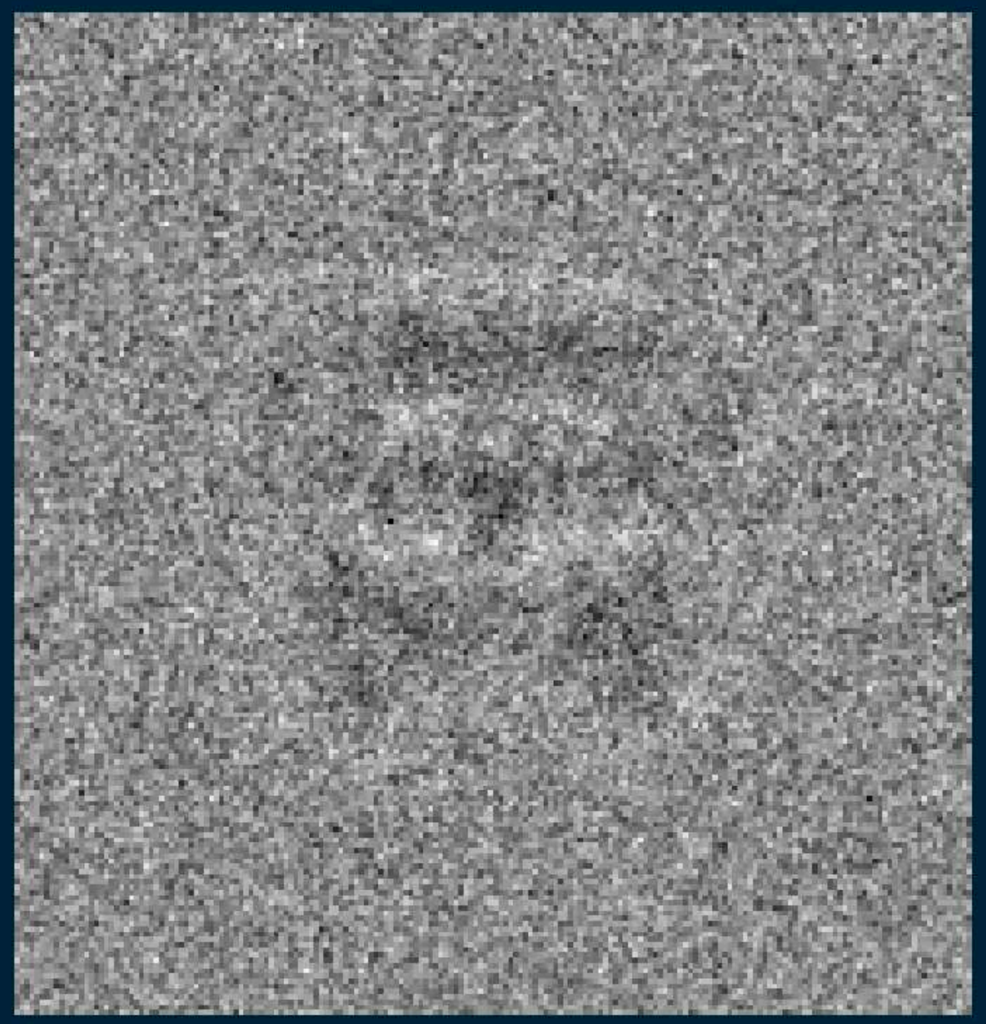
Image restoration when spectral SNR is known

Restoration
from multiple images

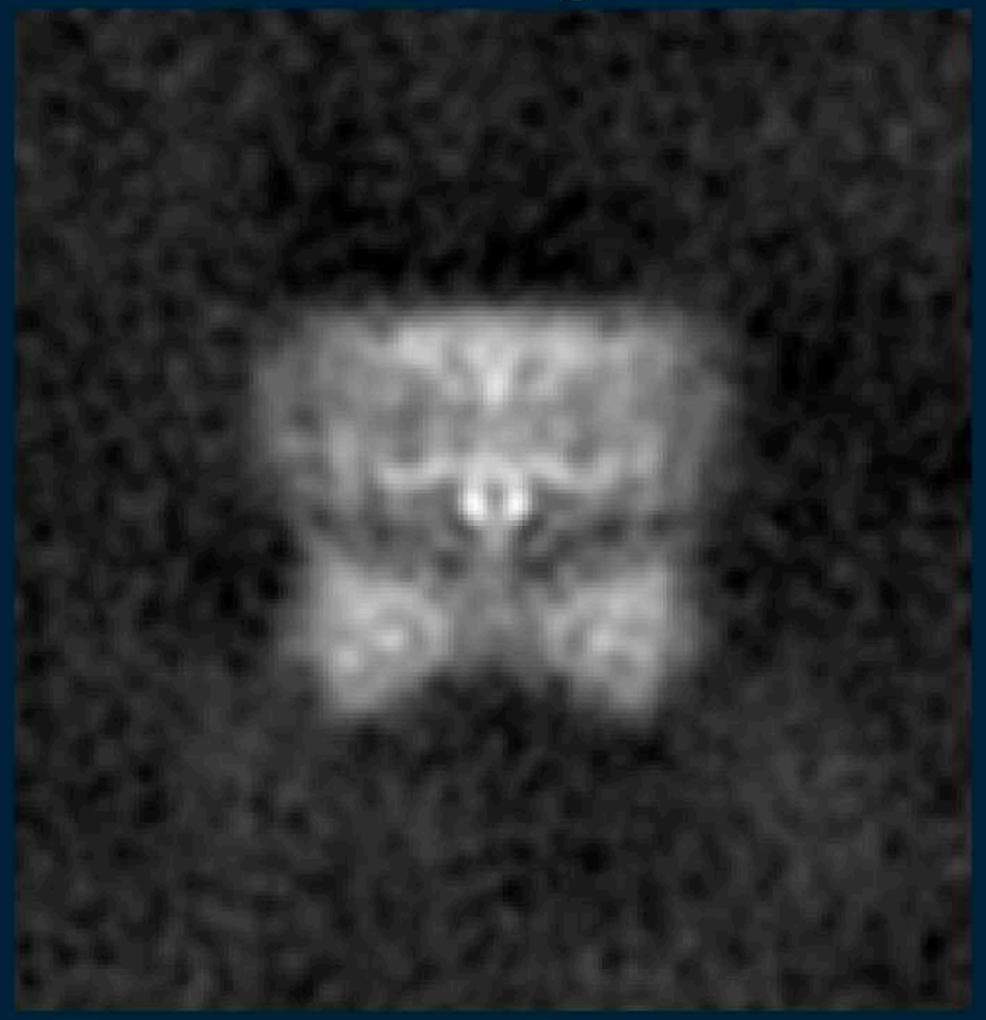
$$\tilde{A} = \frac{\sum_i^N C_i X_i}{k_w(s) + \sum_i^N C_i^2}$$

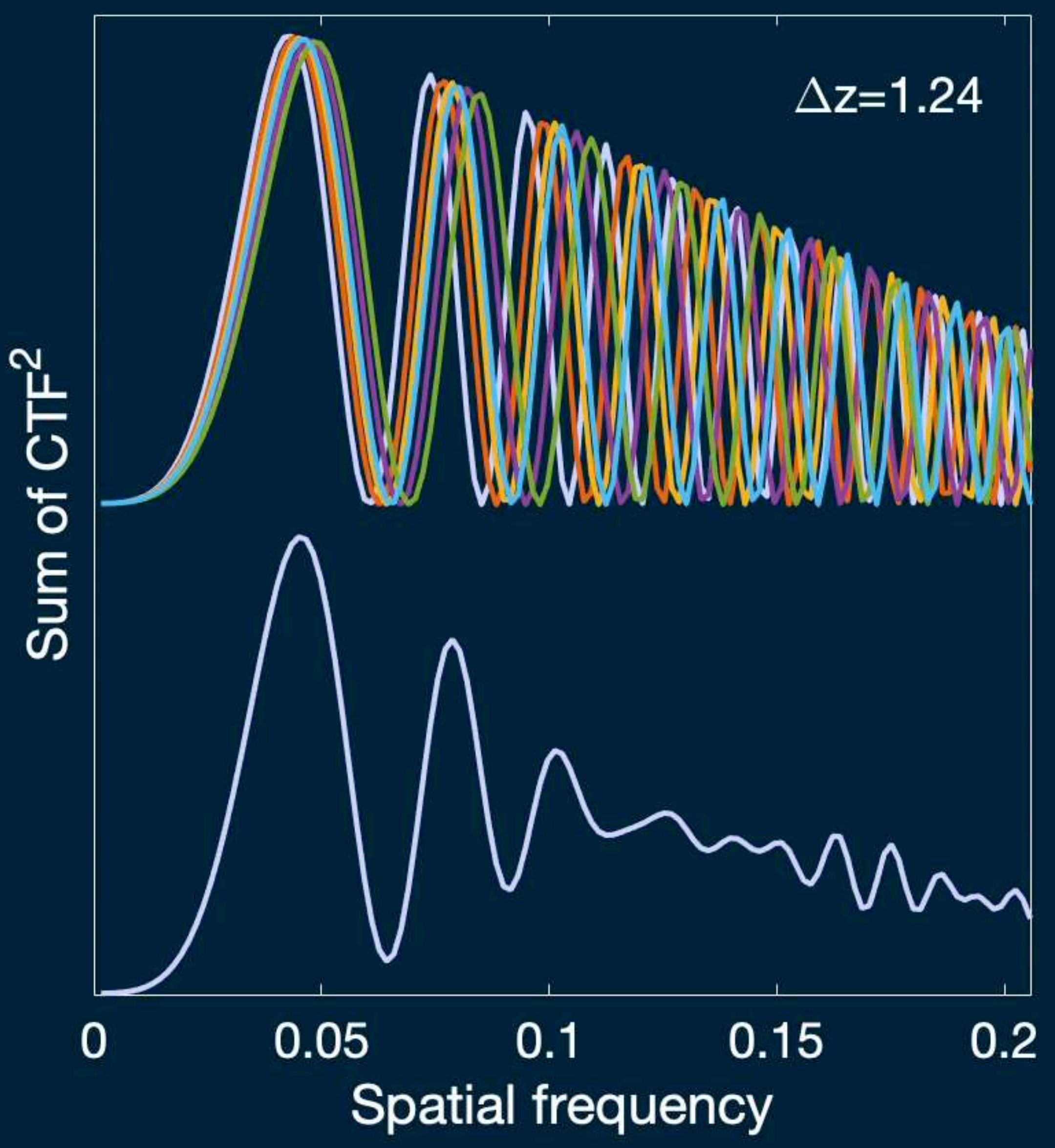
The defocus varies
to fill in CTF zeros

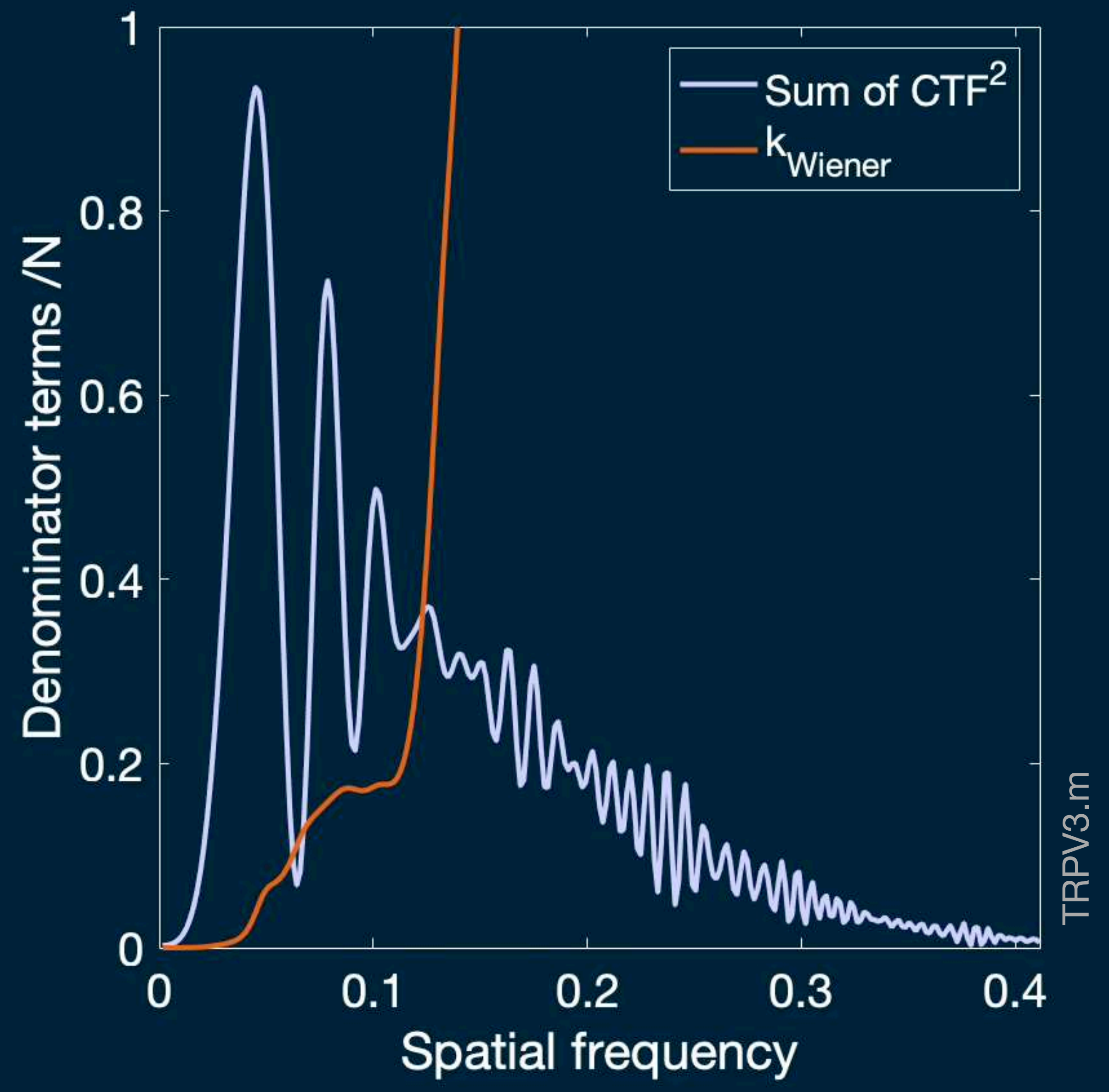




N= 6 images







Even the small defocus range
1–1.5 μm is sufficient.

3D Reconstruction

Correlation and particle picking

CTF “correction”

Single-particle reconstruction

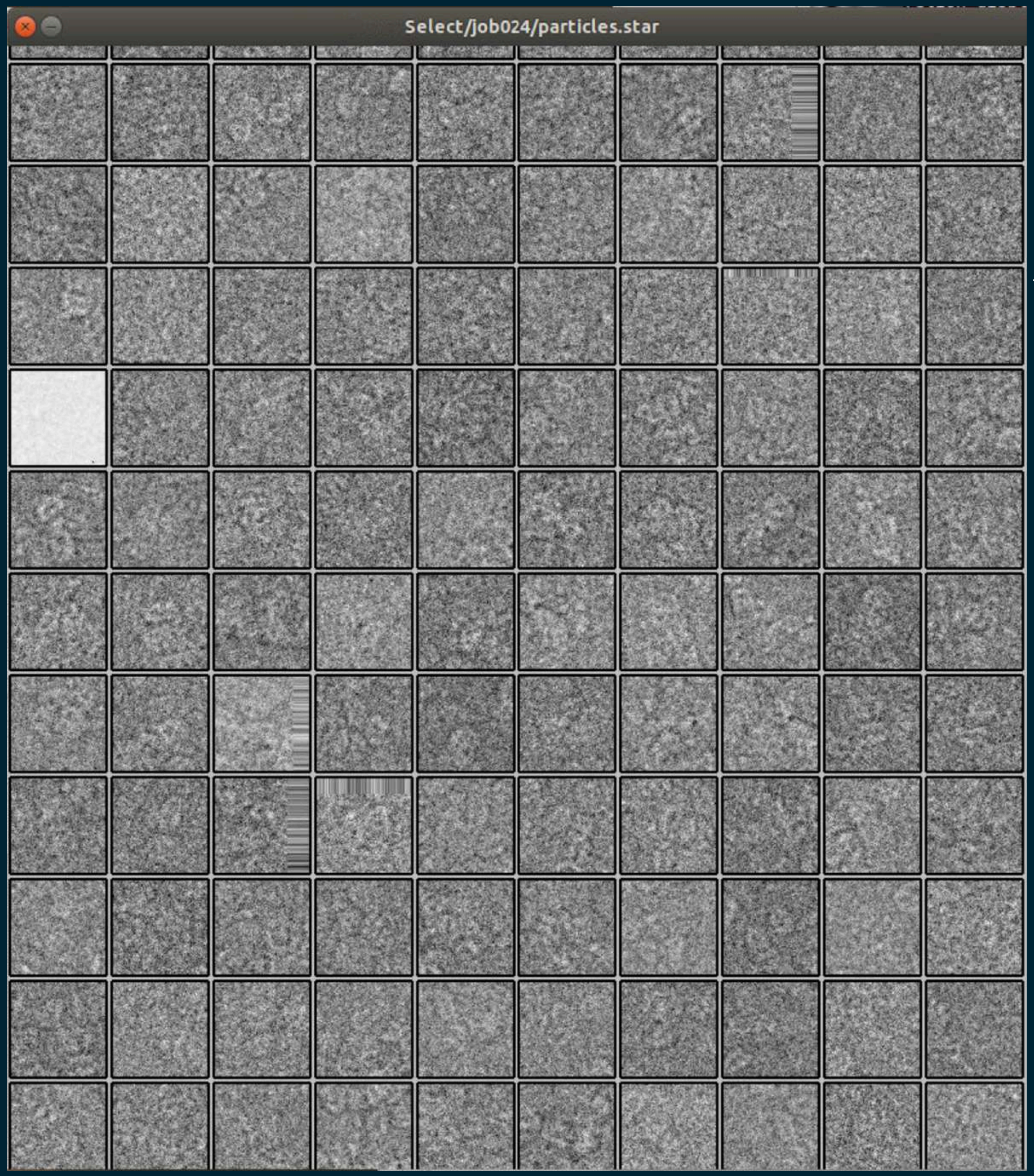
Maximum-likelihood methods

Single-particle reconstruction

We assume that image X_i comes from a projection in direction ϕ_i of volume V according to

$$X_i = C_i \mathbf{P}_{\phi_i} V + N_i$$

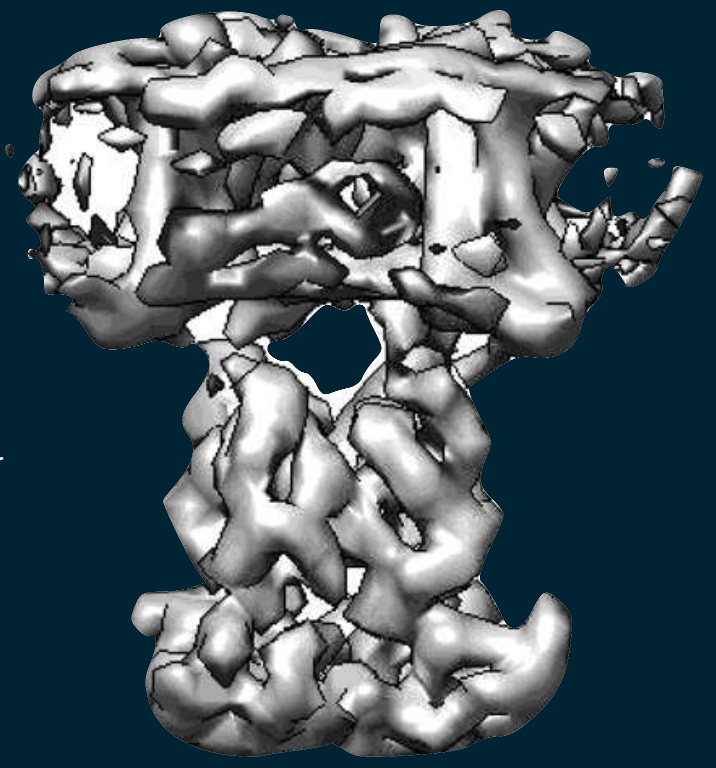
The goal is to discover the volume V



X_i

Project along ϕ_i

V



3D reconstruction in FREALIGN—it's like a Wiener filter

A Frealign iteration, refining $V^{(n)}$ to $V^{(n+1)}$, consists of two steps:

1. Vary the projection direction ϕ_i to find the projection image $R_i = C_i \mathbf{P}_{\phi_i} V^{(n)}$ that maximizes the correlation coefficient for each image X_i,

$$\mathrm{CC} = \frac{X_i \cdot R_i}{|X_i||R_i|}.$$

2. Knowing the best projection direction ϕ_i for each image X_i, update the volume according to

$$V^{(n+1)} = \frac{\sum_i^N \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i X_i}{k + \sum_i^N \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i^2}$$

Notes

1. C_i is the CTF corresponding to the image X_i.
2. The projection operator $\mathbf{P}_\phi$ also includes translations. So ϕ consists of five variables: $\phi = \{\alpha, \beta, \gamma, t_x, t_y\}$.
3. $\mathbf{P}_{\phi_i}^{\mathbf{T}}$ is the corresponding back projection operator. In Fourier space it yields a volume that is all zeros except for values along a slice.

3D reconstruction in FREALIGN—iterations

1. Start with a preliminary structure $V^{(n)}, n = 1$

2. For each particle image X_i find the projection angles ϕ_i that gives the best match, so $X_i \approx C_i \mathbf{P}_{\phi_i} V^{(n)}$

3. Use the Frealign iteration to produce a new 3D volume $V^{(n+1)}$

Iterate

3D Reconstruction

Correlation and particle picking

CTF “correction”

Single-particle reconstruction

Maximum-likelihood methods

There are various ways to compare images

Define the "reference" as the true image A modified by the CTF C:

$$R = CA$$

We wish to compare a data image X with it.

Squared difference

$$\|X - R\|^2 = \sum_j (\mathrm{X}_j - \mathrm{R}_j)^2$$
$$= \|X\|^2 - 2X \cdot R + \|R\|^2$$

Correlation

$$\mathrm{Cor} = X \cdot R$$
$$= \sum_j \mathrm{X}_j \mathrm{R}_j$$

Correlation coefficient

$$\mathrm{CC} = \frac{X \cdot R}{|X|\,|R|}$$

Notation used here:

A single pixel in the image X:

X_j —the j^{th} pixel (out of J pixels total)

The i^{th} image in the dataset $\mathbf{X}$:

X_i

Probabilities, another way to compare images

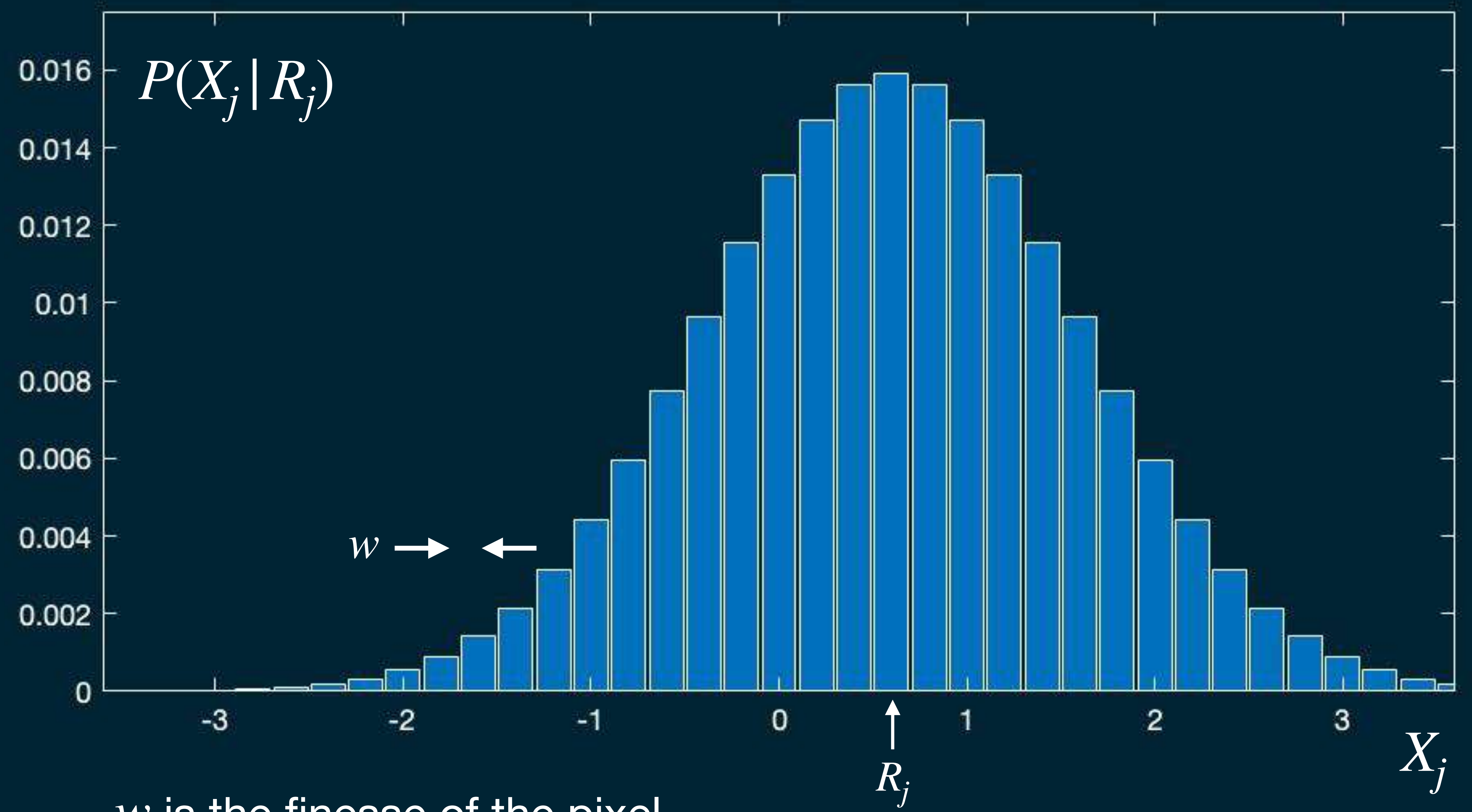
$$X = R + N$$

<u>Probability of a pixel value:</u>

$$P(X_j | R_j) = \frac{\cancel{w}\ 1}{\sqrt{2\pi\sigma^2}} e^{-(X_j - R_j)^2/2\sigma^2}$$

<u>Probability of observing an image that comes from R:</u>

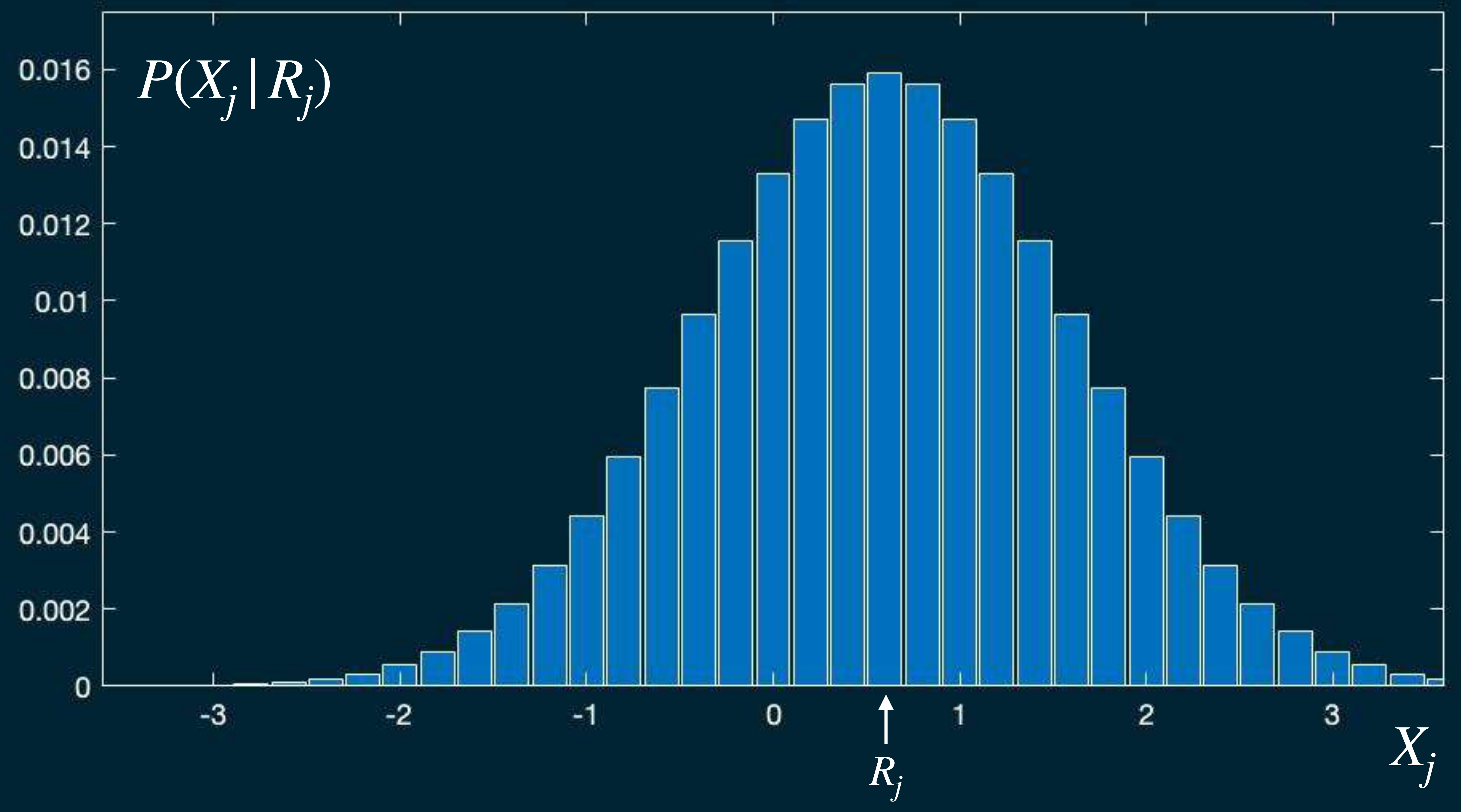
$$P(X | R) = \frac{\cancel{w^J}\ 1}{(2\pi\sigma^2)^{J/2}} e^{-||X - R||^2/2\sigma^2}$$



w is the finesse of the pixel intensity measurements. We'll ignore it (set it to 1).

Probabilities, another way to compare images

$$X = R + N$$



Probability of observing an image that comes from R:

$$P(X | R) = c\, e^{-||X-R||^2/2\sigma^2}$$

(The normalization factor c we'll treat as a constant and ignore it.)

The Likelihood

Let $\mathbf{X} = \{X_1 \ldots X_N\}$ be our "stack" of particle images. We'd like to find the best 3D volume consistent with these data, say maximizing

$$P(V \mid \mathbf{X}).$$

According to Bayes' theorem,

$$P(V \mid \mathbf{X}) = P(\mathbf{X} \mid V) \frac{P(V)}{P(\mathbf{X})}.$$

prior → **Experiment** → *posterior*

1. $P(\mathbf{X})$ doesn't depend on V so we can ignore it.
2. $P(V)$ is called the prior probability. It reflects any knowledge about V that we have before considering the data set.
3. $P(\mathbf{X} \mid V)$ is something we can calculate. It's called the likelihood of V.

$$\mathrm{Lik}(V) = P(\mathbf{X} \mid V)$$

We know how to compute the likelihood

To get the likelihood for one image we just integrate over all the ϕ's:

$$P(X\,|\,V) = \int P(X\,|\,V,\phi)\,d\phi$$

We know that

$$P(X\,|\,V,\phi) = c\,e^{-\|X-\mathbf{CP}_\phi V/\|^2/2\sigma^2}$$

To get the likelihood for the whole dataset we compute the product over all the images,

$$P(\mathbf{X}\,|\,V) = \prod_i^N \int P(X_i\,|\,V,\phi)\,d\phi,$$

or for numerical sanity, we compute the log likelihood,

$$L = \sum_i^N \ln\left(\int P(X_i\,|\,V,\phi)\,d\phi\right).$$

Maximum-likelihood reconstruction is finding V that maximizes L.

Maximum-likelihood estimation is asymptotically unbiased

If the size of the dataset grows without bounds
(and the number of parameters to be estimated do not)
ML converges to the right answer.

$$L = \sum_{i}^{N} \ln\left(\int P(X_i \,|\, V, \phi)\, d\phi \right).$$

To maximize the likelihood, we'll need a probability function $\Gamma(\phi)$

A projection

$$A = \mathbf{P}_\phi V$$

Probability of observing an image X_i

$$P(X_i \,|\, V, \phi) = c\, e^{-||X_i - \mathbf{CP}_\phi V||^2 / 2\sigma^2}$$

Probability of a projection direction

$$\Gamma_i(\phi) = P(\phi \,|\, X_i, V) = \frac{P(X_i \,|\, V, \phi)}{\int P(X_i \,|\, V, \phi) d\phi}$$

The E-M algorithm finds a local maximum of the likelihood

The Expectation-Maximization (E-M) algorithm has this iteration, guaranteed to increase the likelihood:

$$V^{(n+1)} = \frac{\sum_i \int \Gamma_i^{(n)}(\phi) \mathbf{P}_\phi^\mathbf{T} C_i X_i \, d\phi}{\frac{\sigma^2}{T\tau^2} + \sum_i \int \Gamma_i^{(n)}(\phi) \mathbf{P}_\phi^\mathbf{T} C_i^2 \, d\phi}$$

For comparison, here is Frealign's iteration:

1. Find the best orientation ϕ_i for each particle image X_i
2. Update the volume according to

$$V^{(n+1)} = \frac{\sum_i \mathbf{P}_{\phi_i}^\mathbf{T} C_i X_i}{k + \sum_i \mathbf{P}_{\phi_i}^\mathbf{T} C_i^2}$$

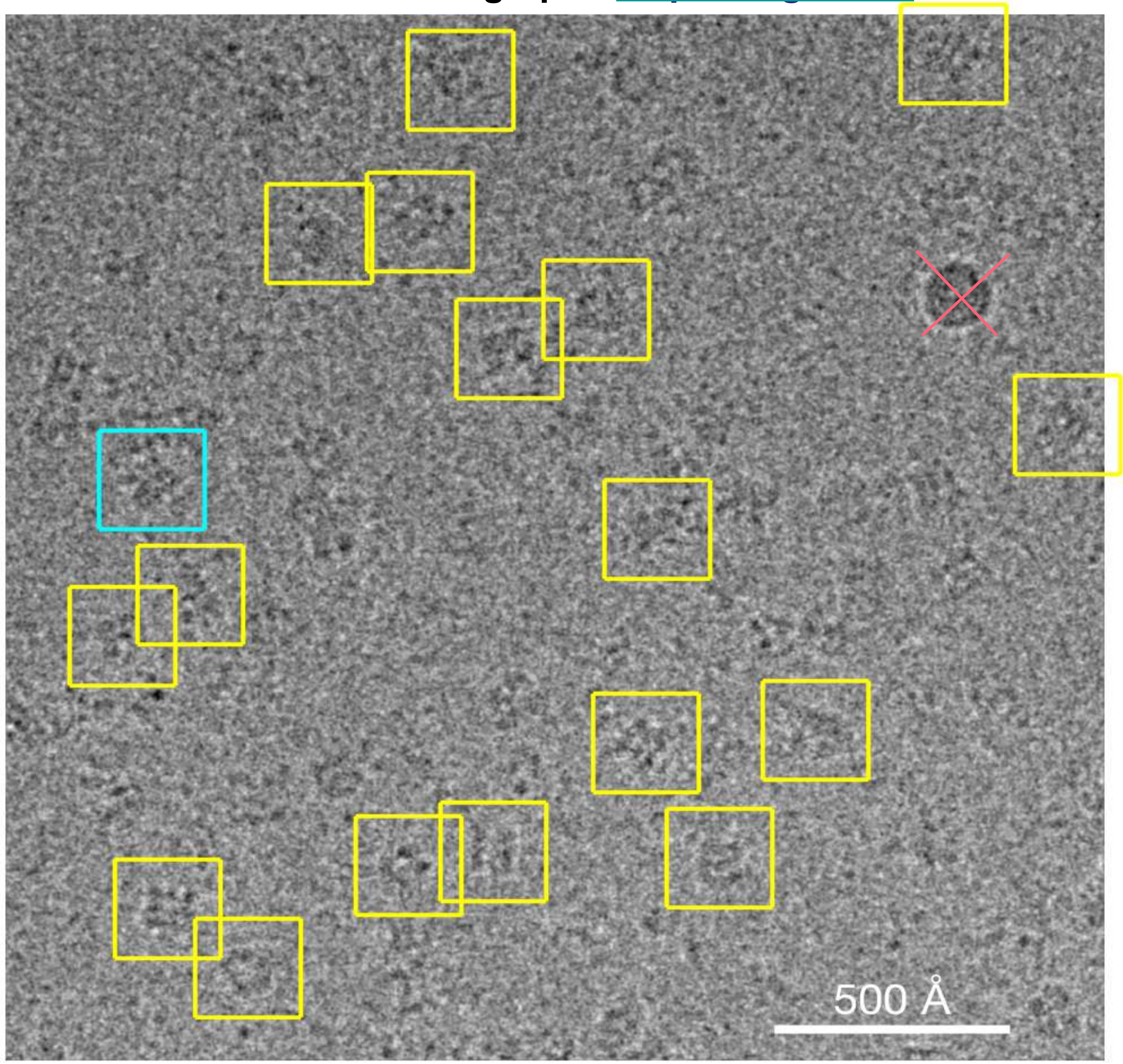
…Relion's compute-intensive "Expectation" step is basically the evaluation of $\Gamma_i(\phi)$ for each image X_i

Determining the orientation angles: example from the TRPV1 dataset

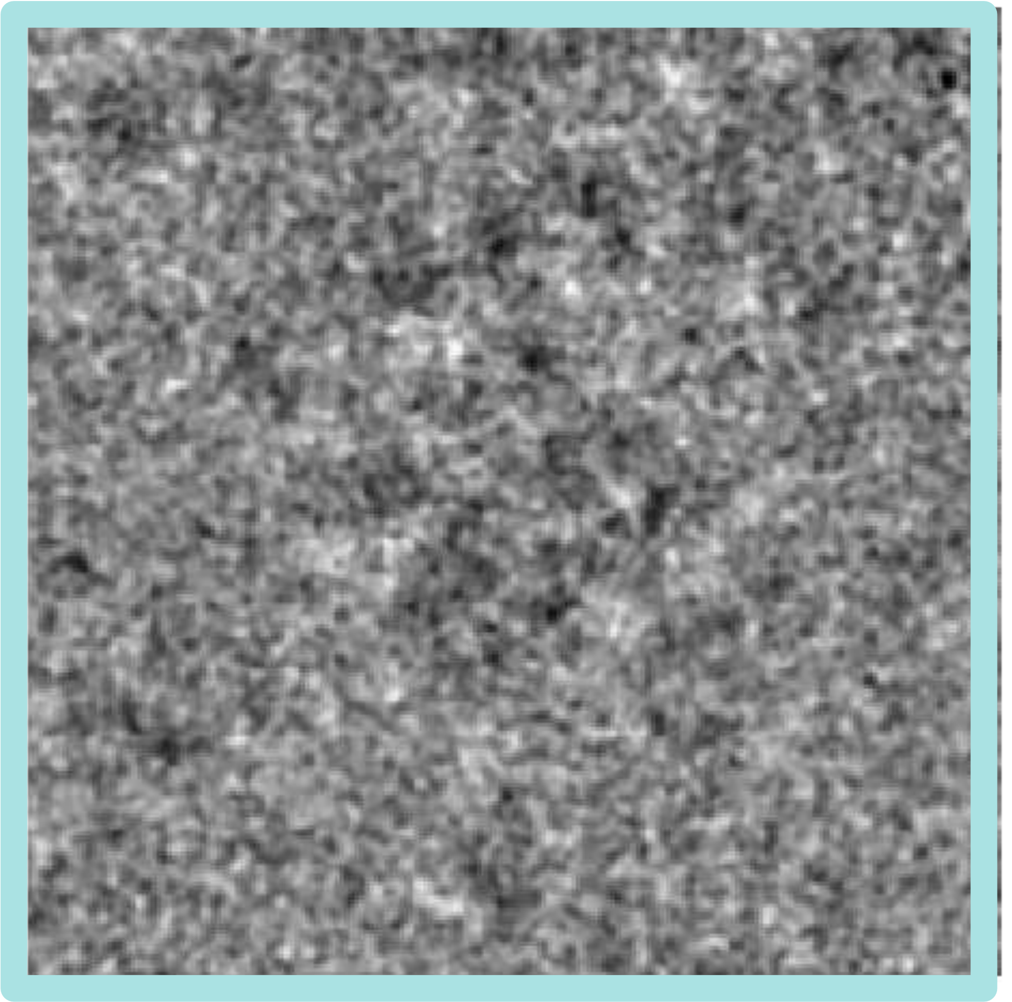
Structure of the TRPV1 ion channel determined by electron cryo-microscopy

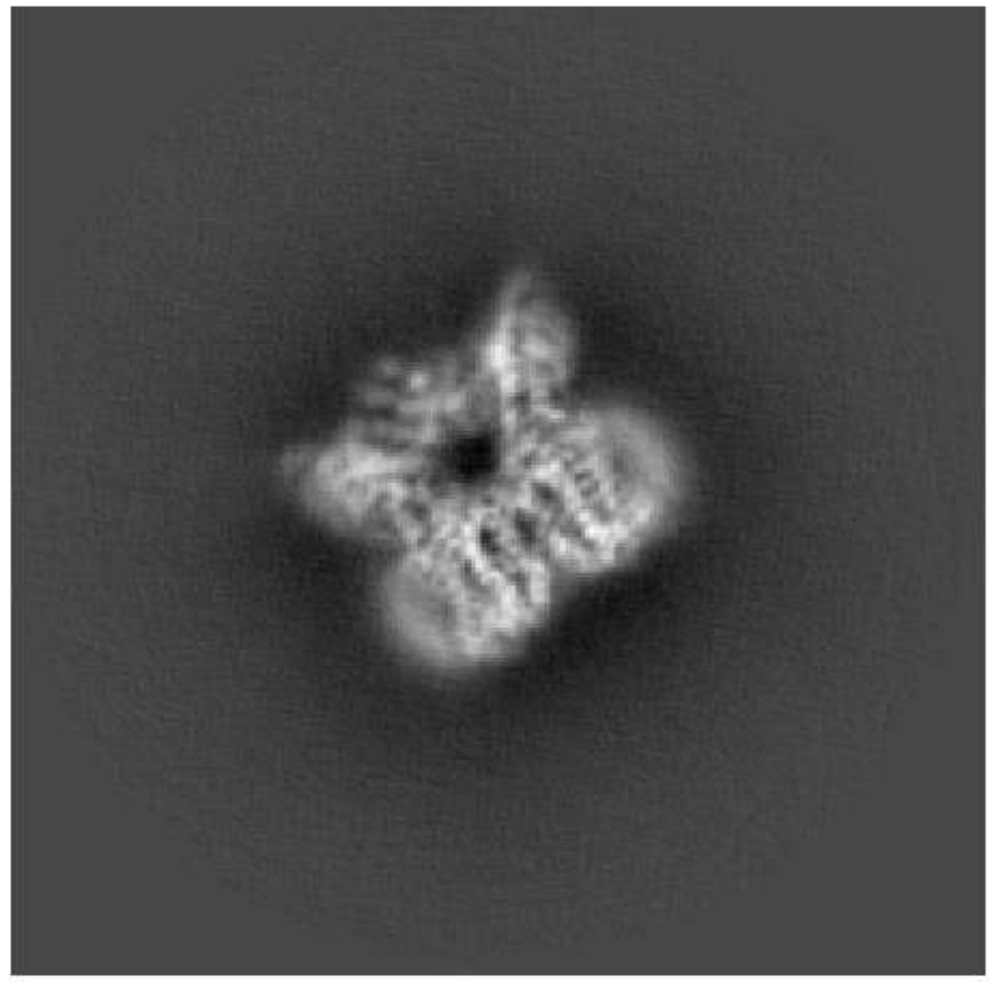
Maofu Liao[1]*, Erhu Cao[2]*, David Julius[2] & Yifan Cheng[1]

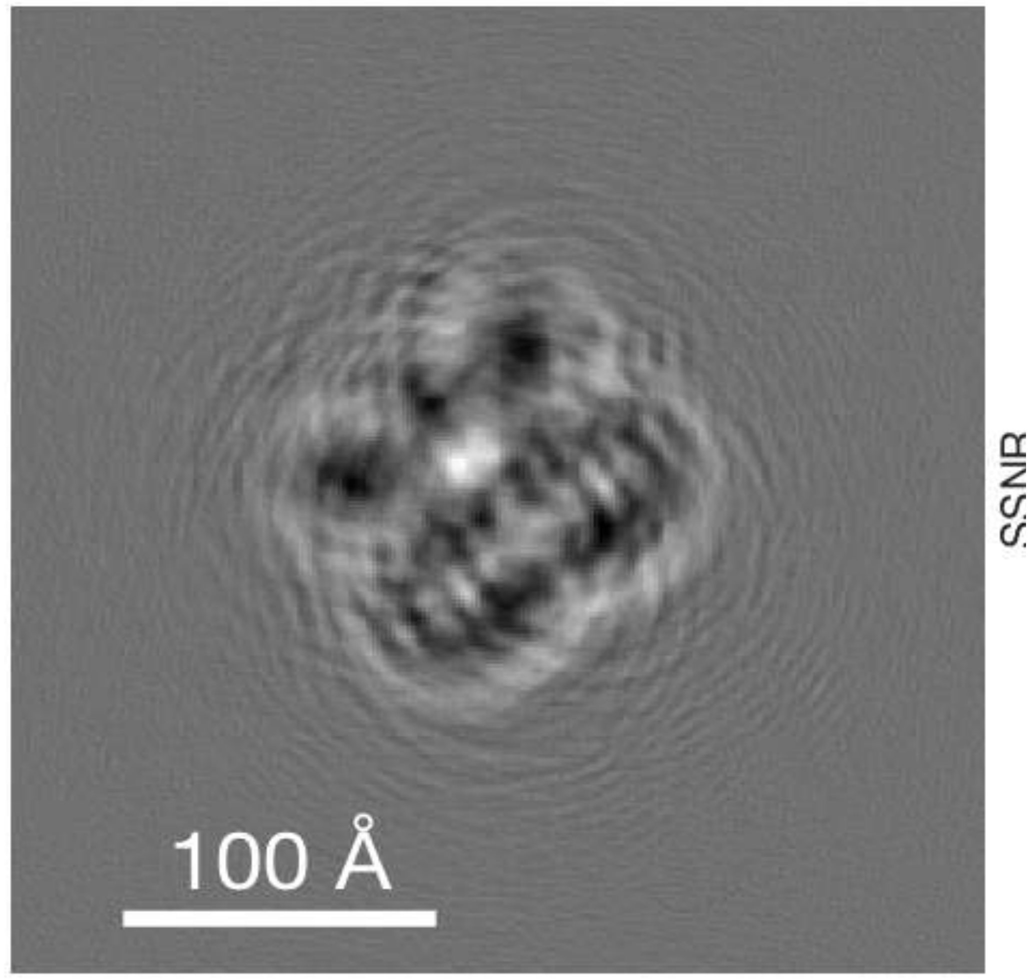
1/4 of a micrograph - empiar.org/10005

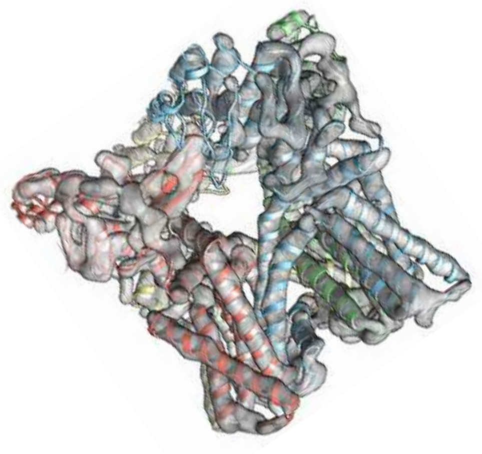


One particle image



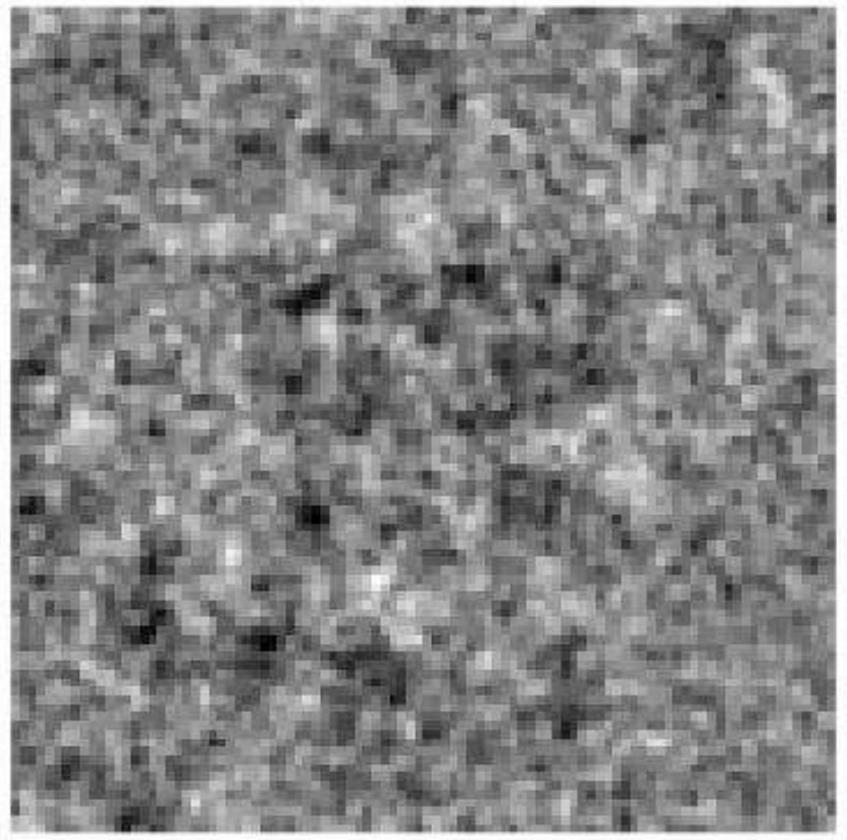




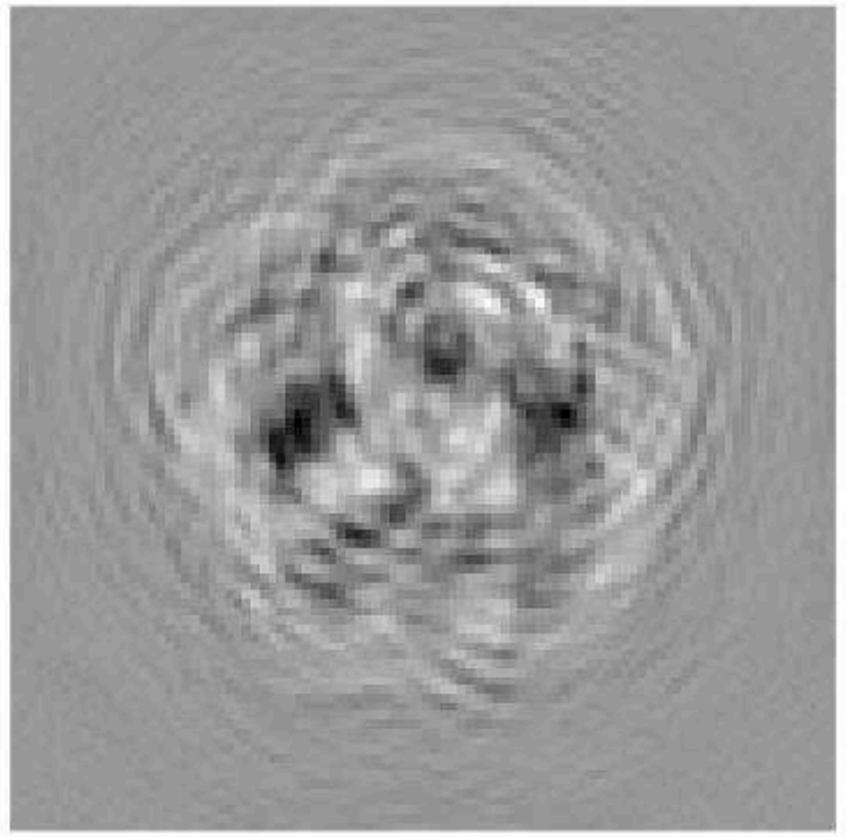


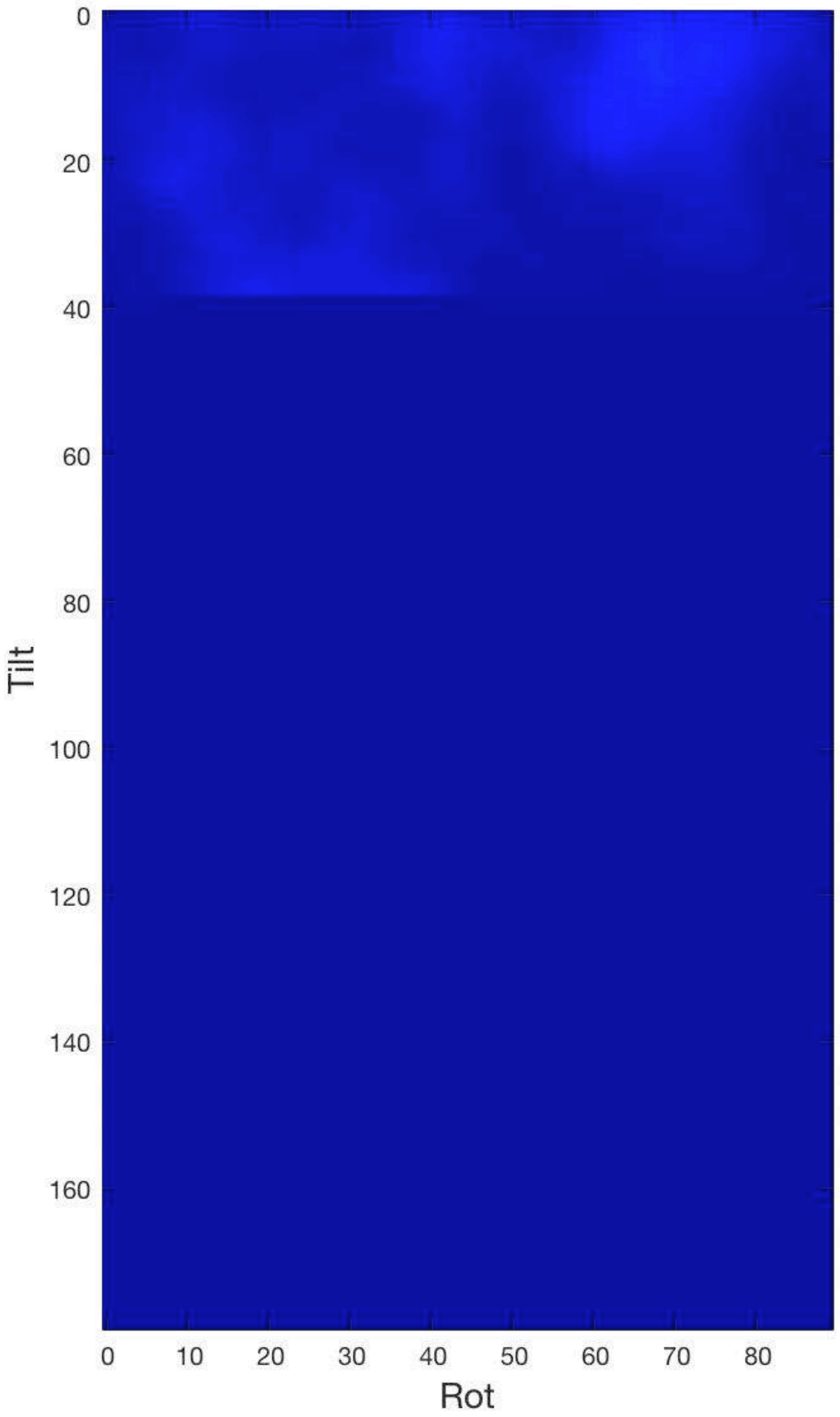
The probability of orientations $P(\phi \mid X, V)$ is remarkably sharp

Particle image

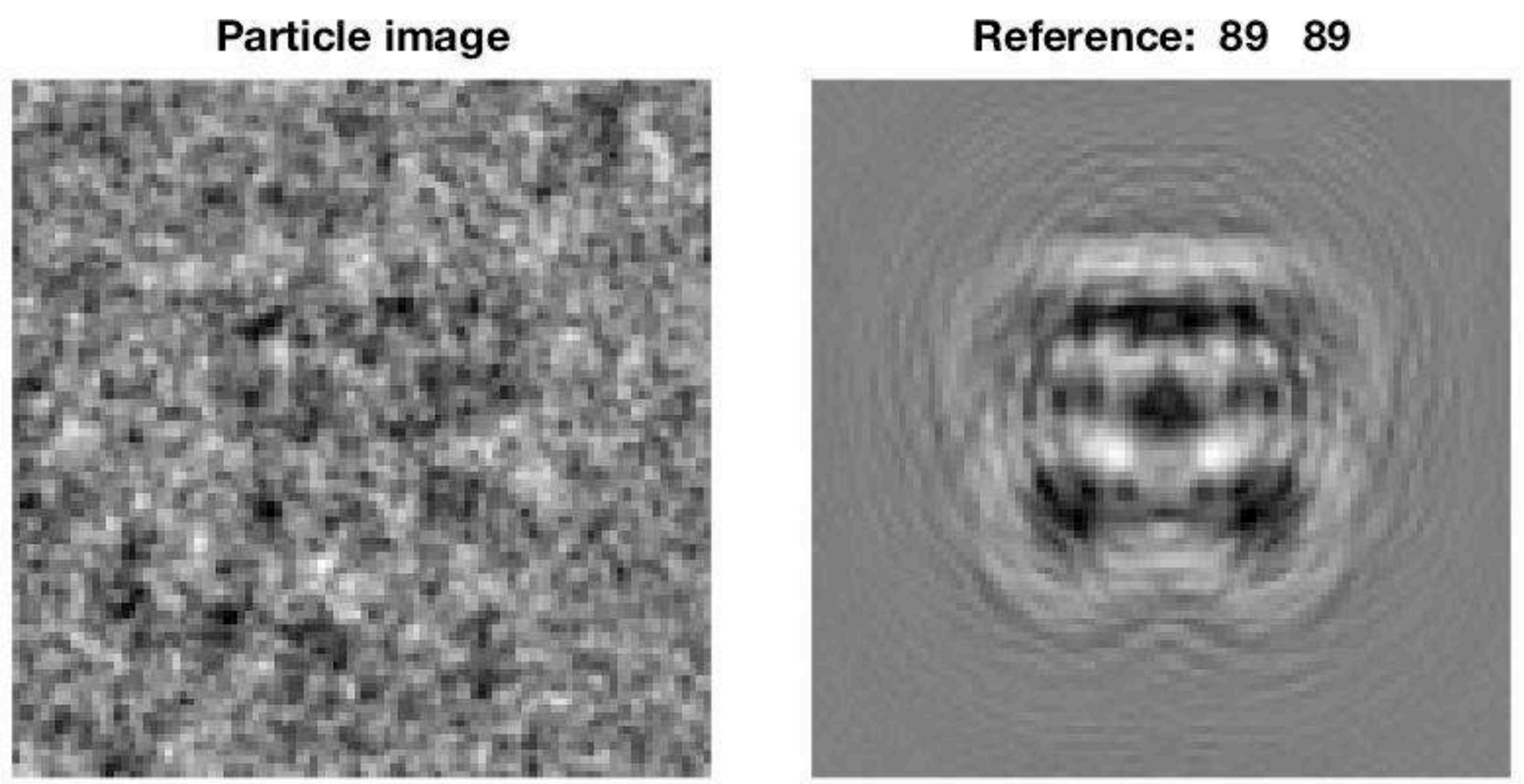


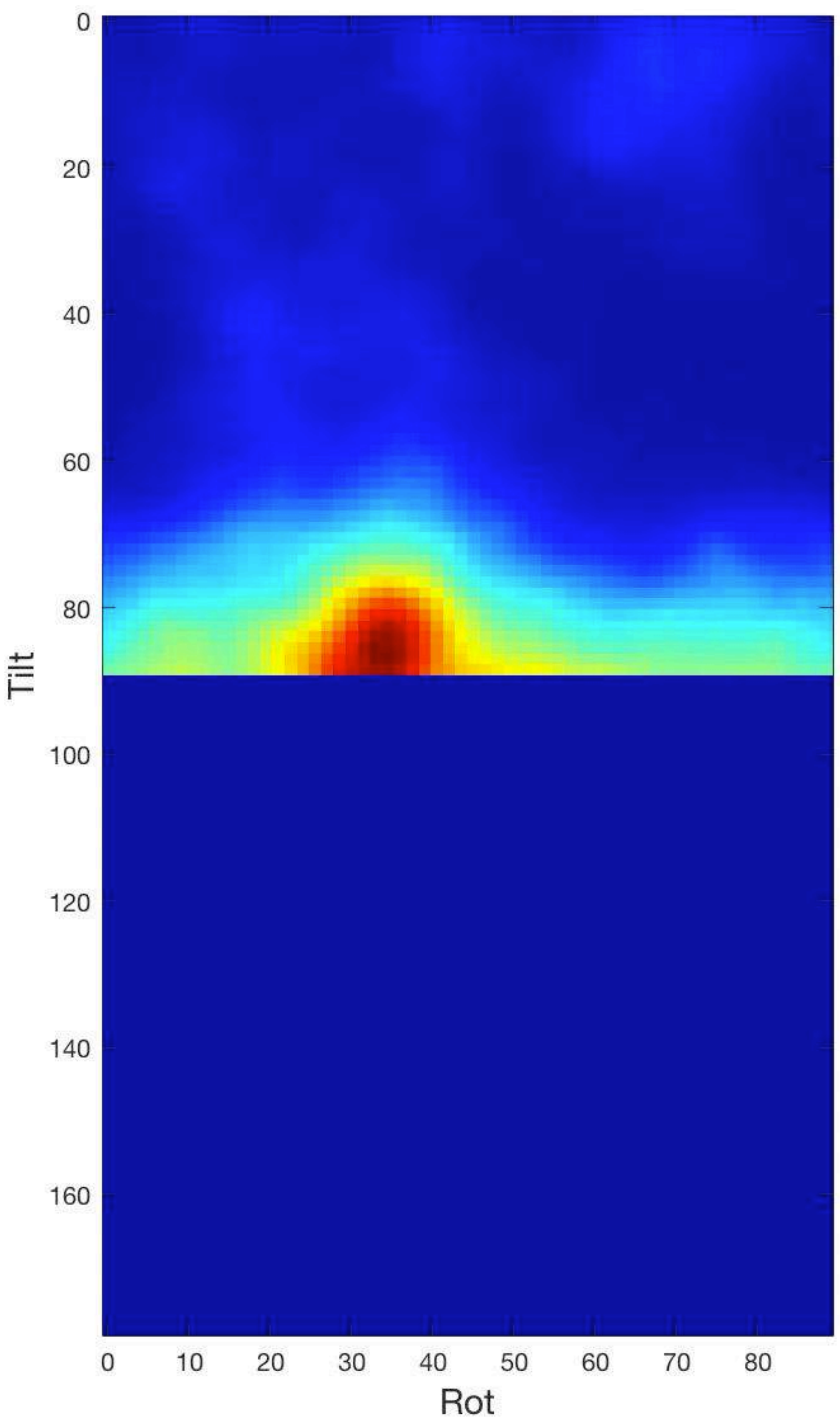
Reference: 86 38





The probability of orientations $P(\phi | X, V)$ is remarkably sharp





The orientation determination is the most expensive step

$$\text{No. operations} \approx \underbrace{\frac{\pi^3}{8} t^2 n^5 N}_{\text{finding orientations}} + \underbrace{\pi n^4 + N n^2}_{\text{3D reconstruction}}$$

The orientation determination is the most expensive step

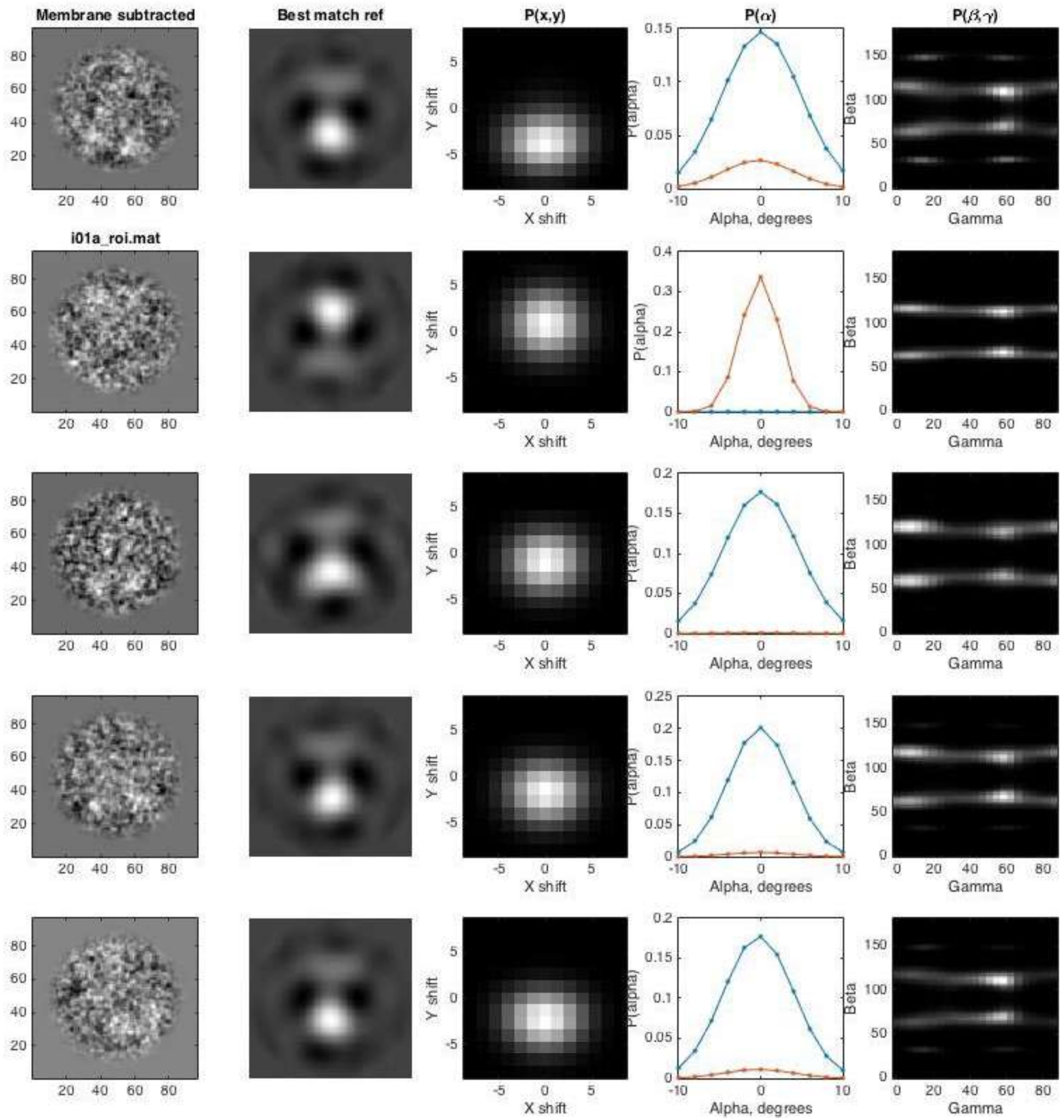
$$\text{No. operations} \approx \underbrace{\frac{\pi^3}{8}t^2n^5N}_{\substack{\text{finding}\\\text{orientations}}} + \underbrace{\pi n^4 + Nn^2}_{\substack{\text{3D recon-}\\\text{struction}}}$$

e.g. $N=10^5$, $n=128$, $t=7$

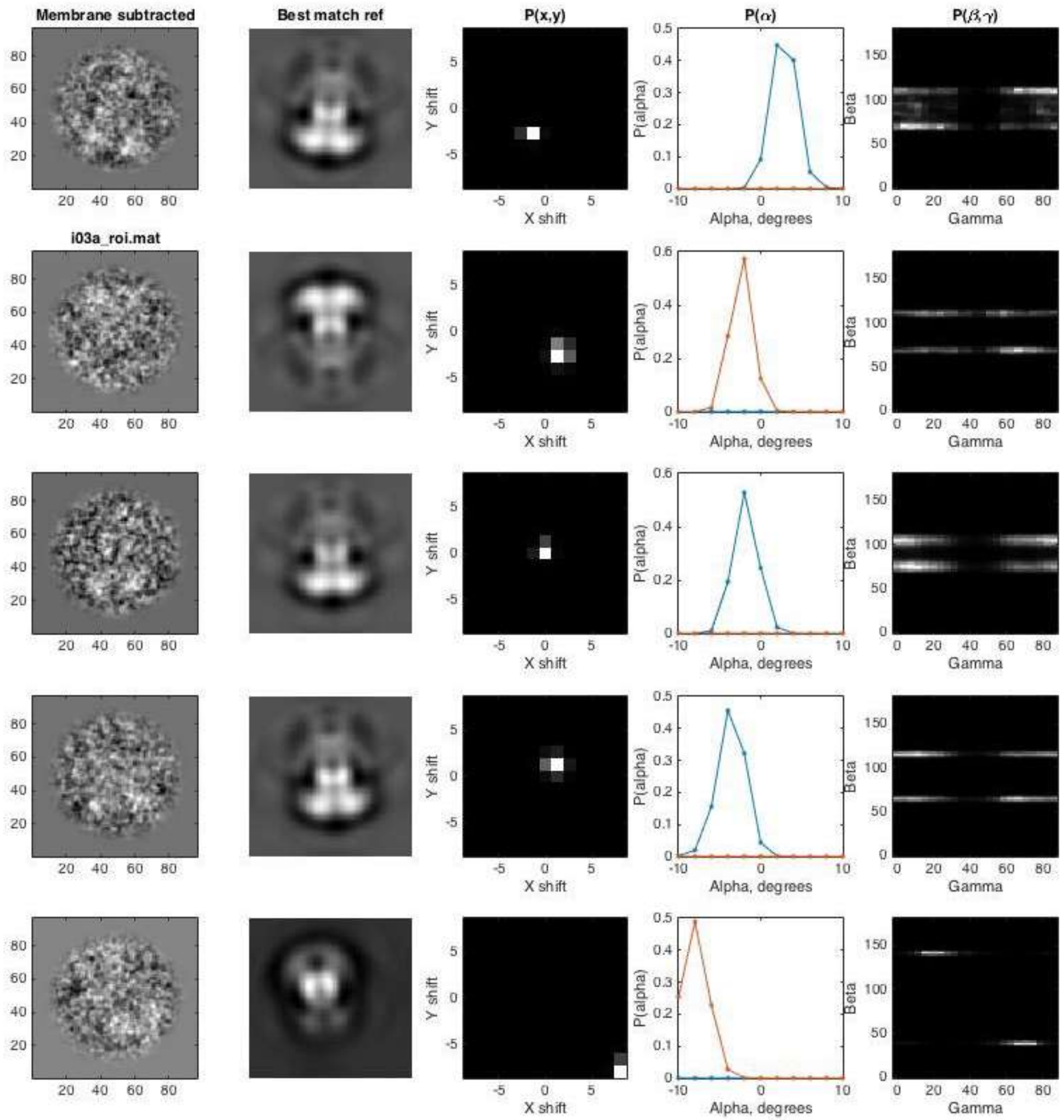
No. operations ≈ 6×10^{17} ≈ 19 CPU-years

With efficient programs, ~ 1 CPU-month

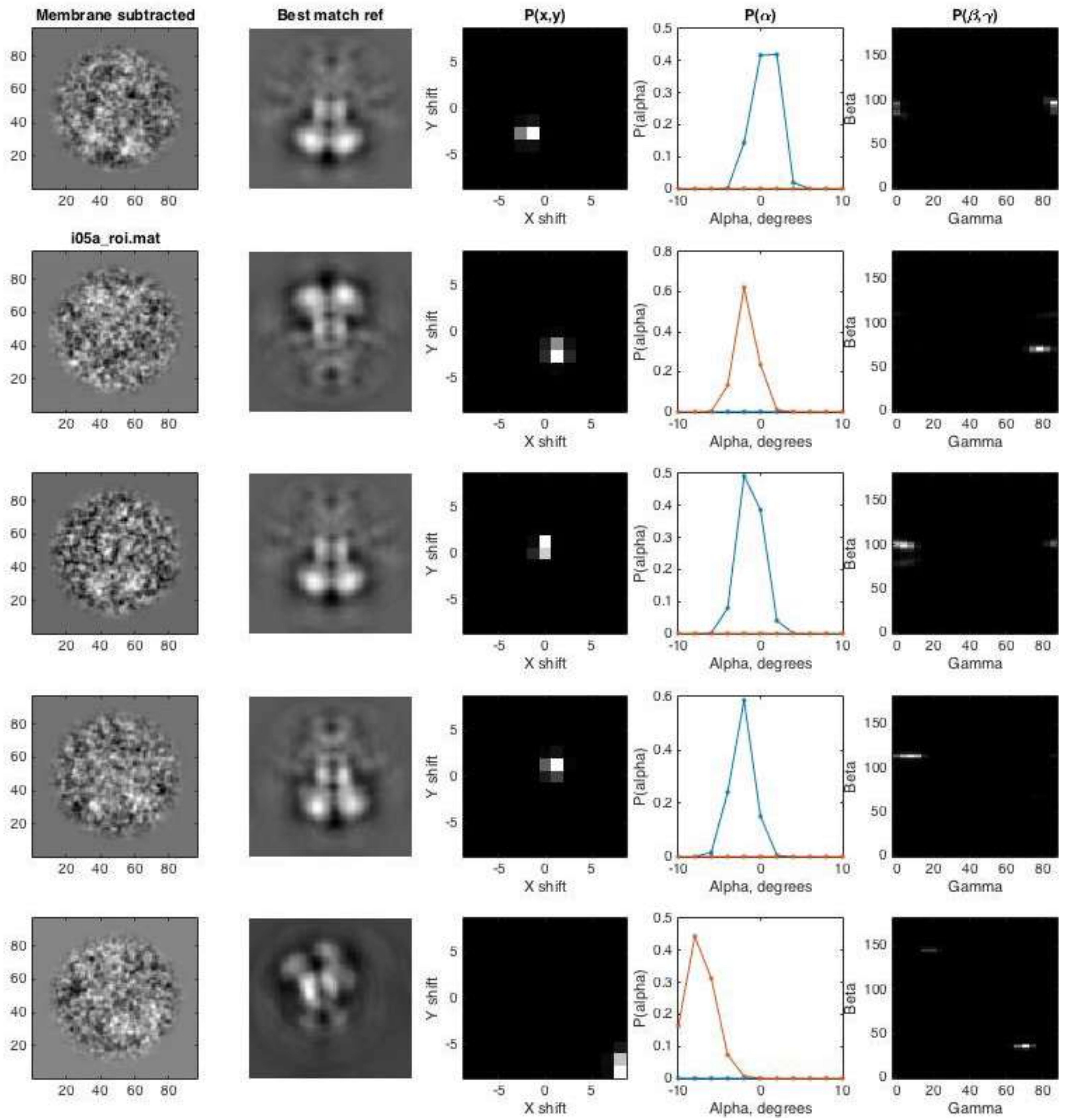
Reconstruction: on the first EM iteration, angle assignments mainly arise from geometry



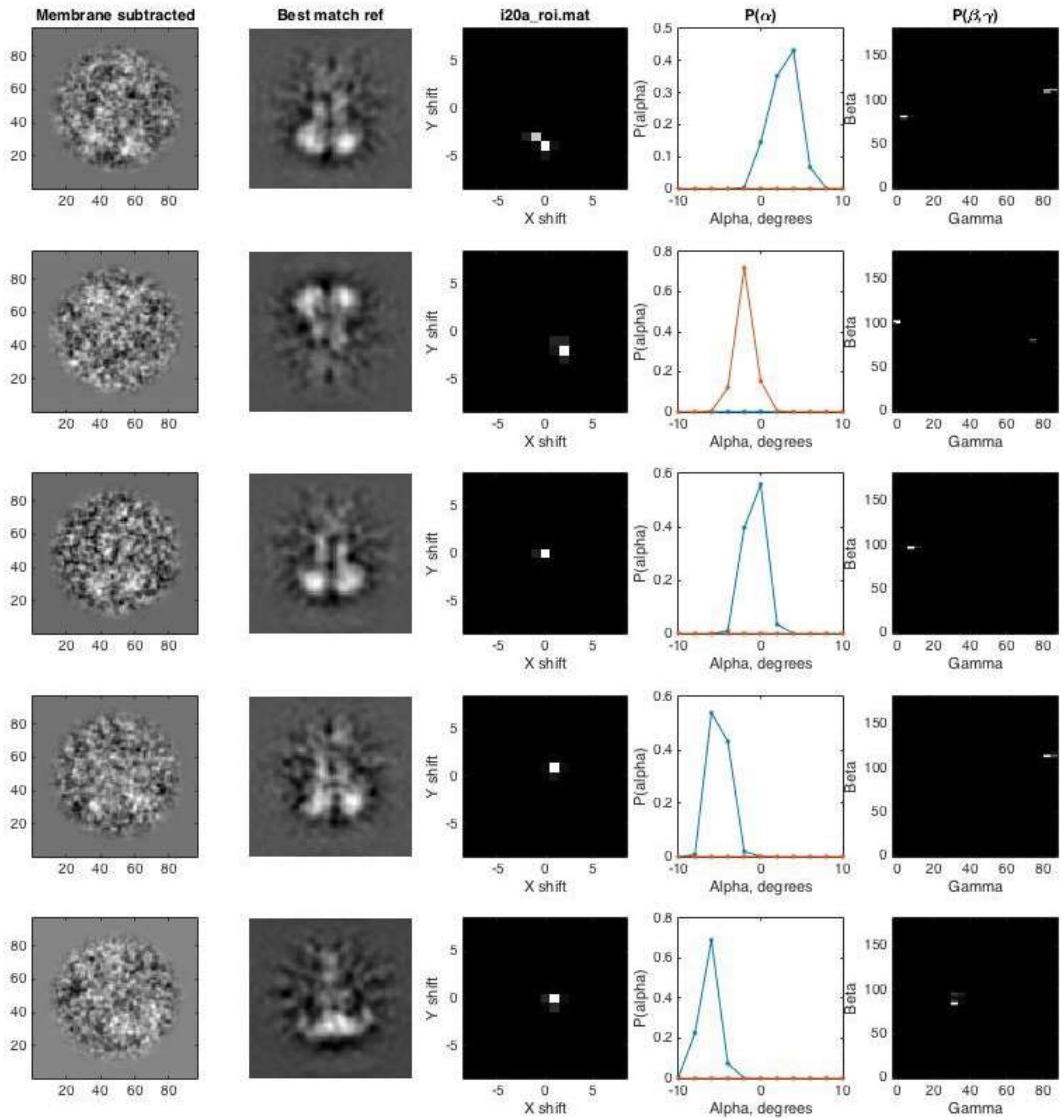
Iteration 3


Membrane subtracted
Best match ref
P(x,y)
P(α)
P(β,γ)
i03a_roi.mat
X shift
Y shift
Alpha, degrees
P(alpha)
Gamma
Beta

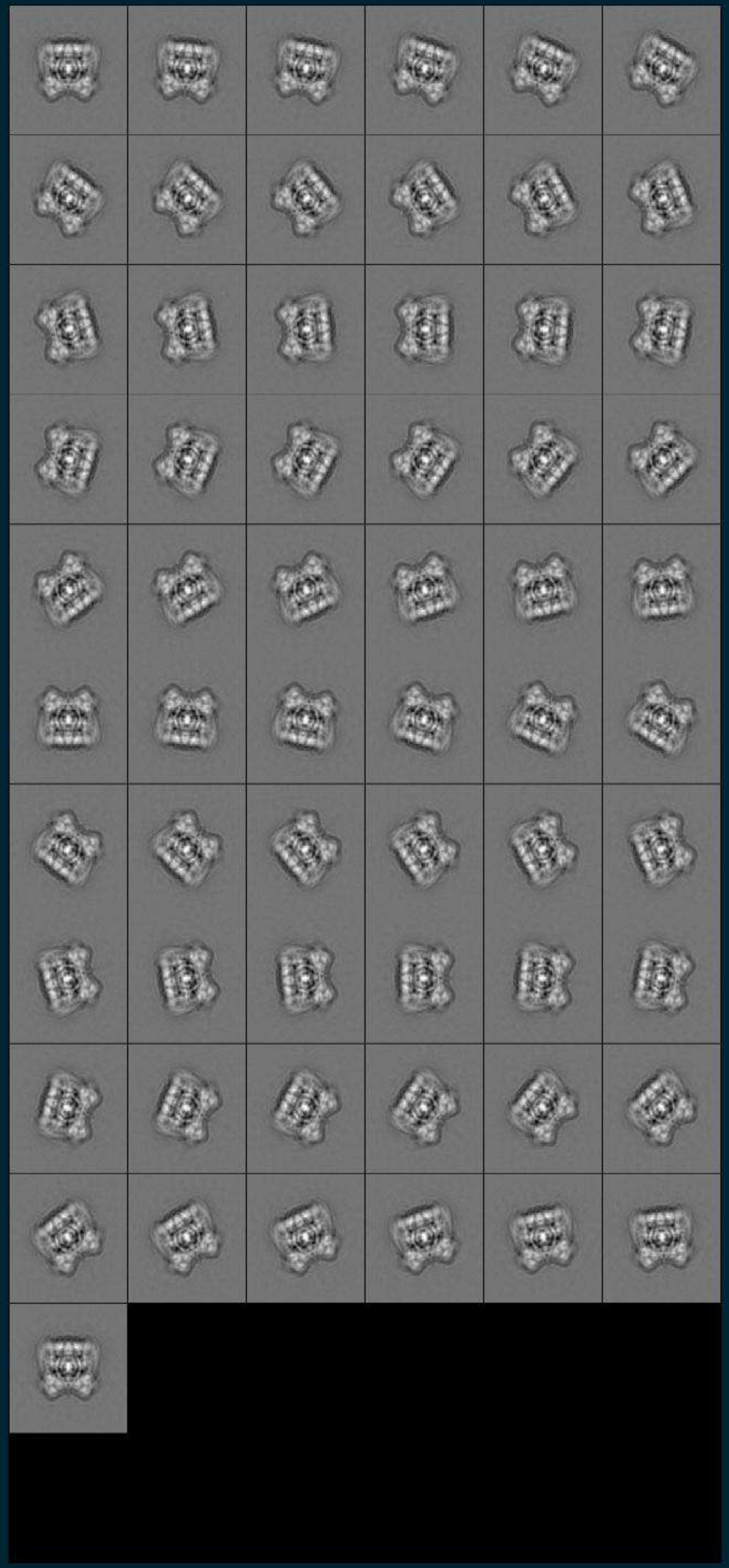
Iteration 5

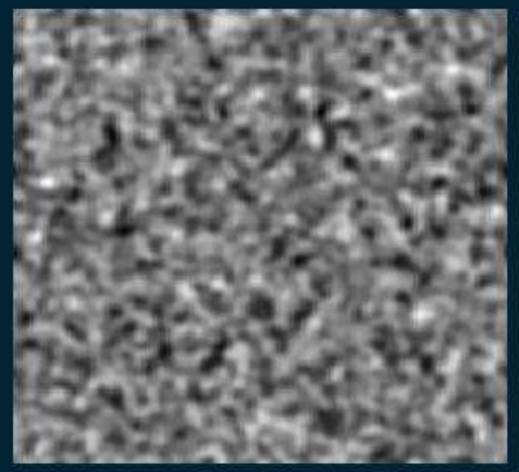

Membrane subtracted
Best match ref
P(x,y)
P(α)
P(β,γ)
i05a_roi.mat
X shift
Y shift
Alpha, degrees
P(alpha)
Gamma
Beta

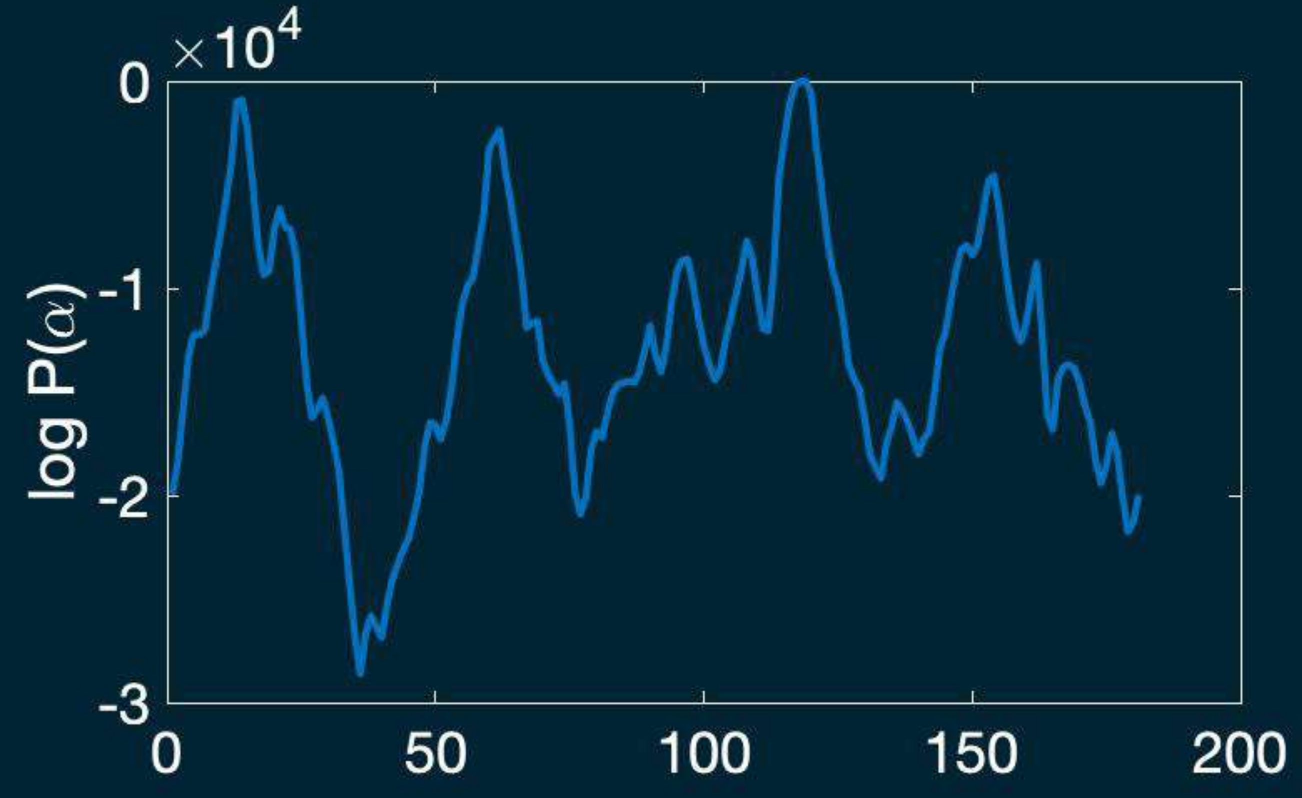
Iteration 14, near convergence: distributions are becoming sharp

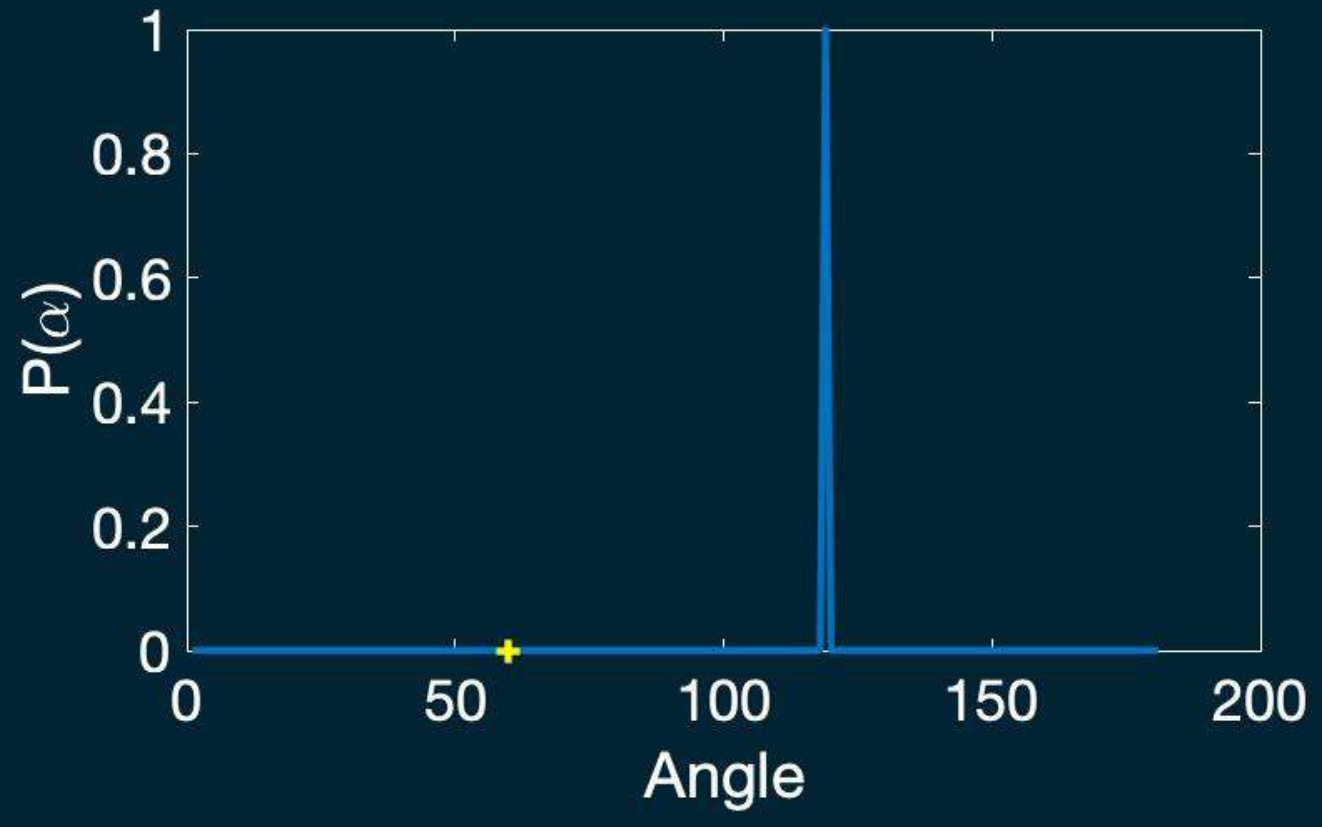


Evaluating Γ_ϕ is expensive: one of 5 parameters

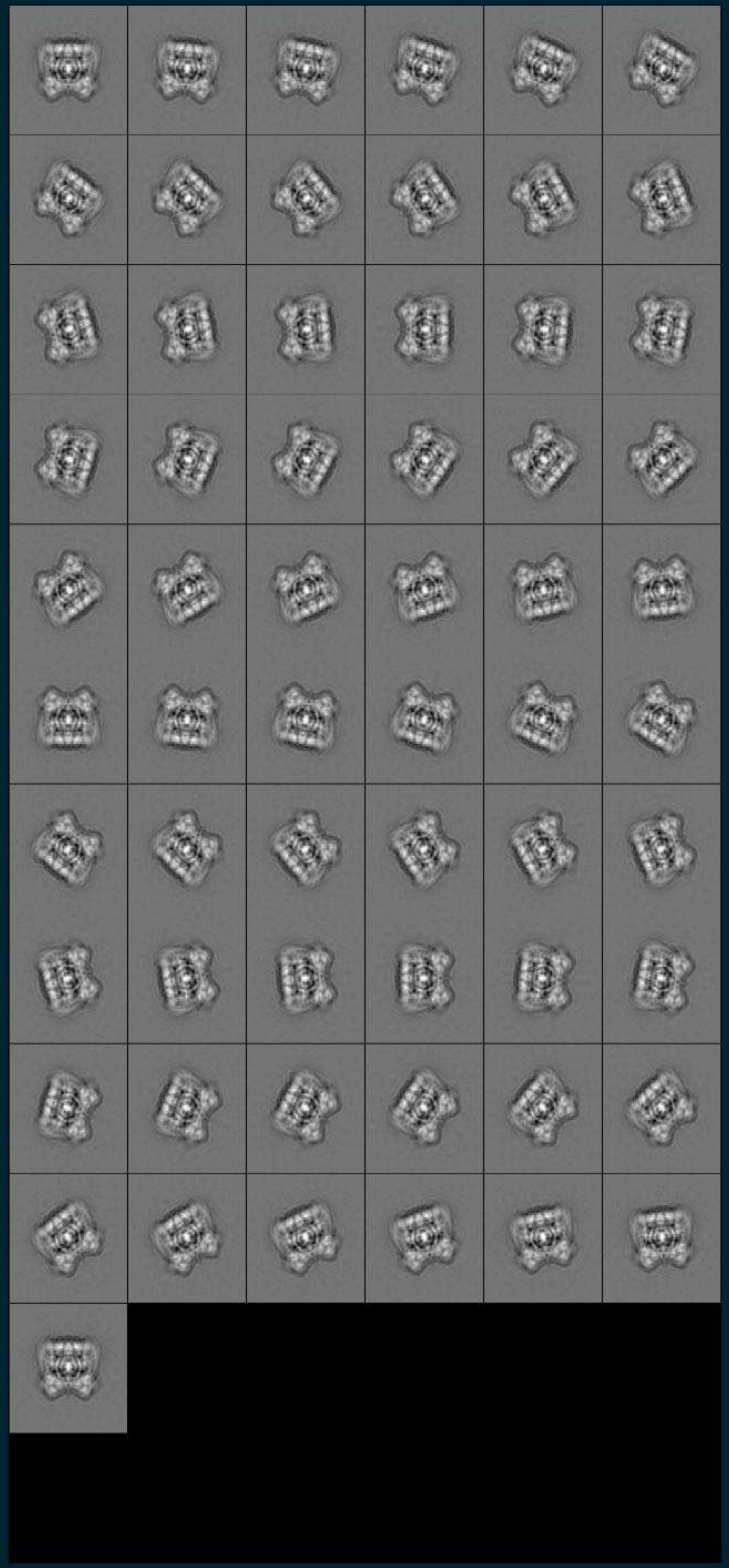


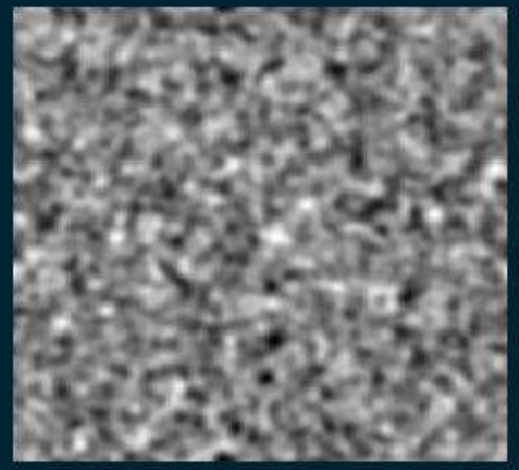


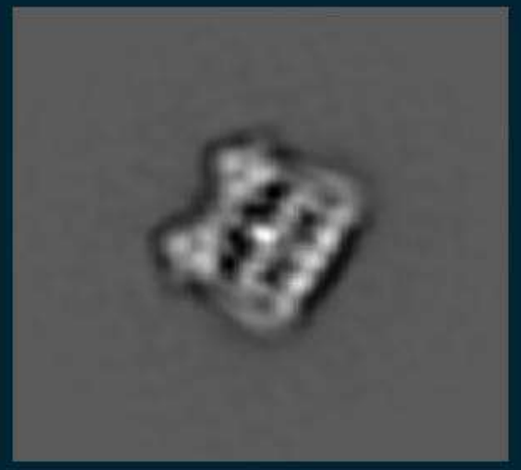


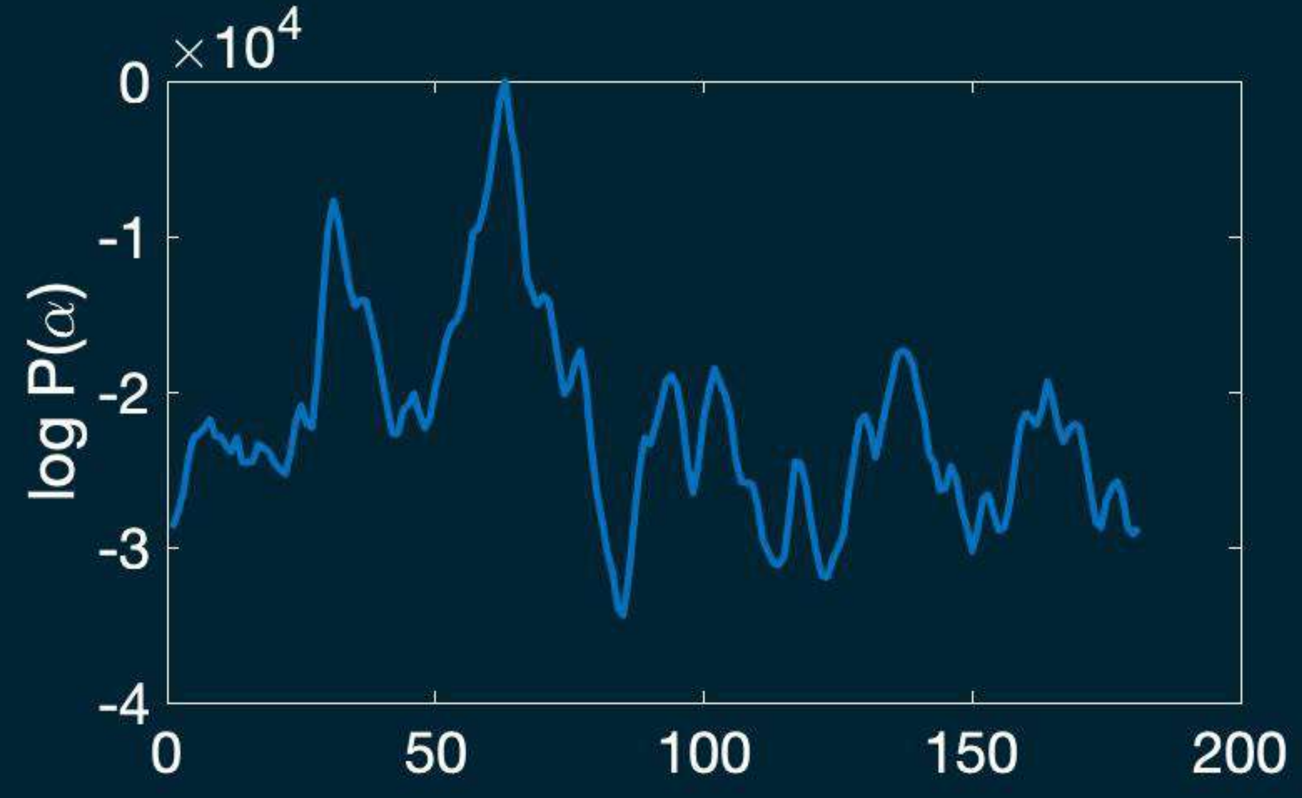


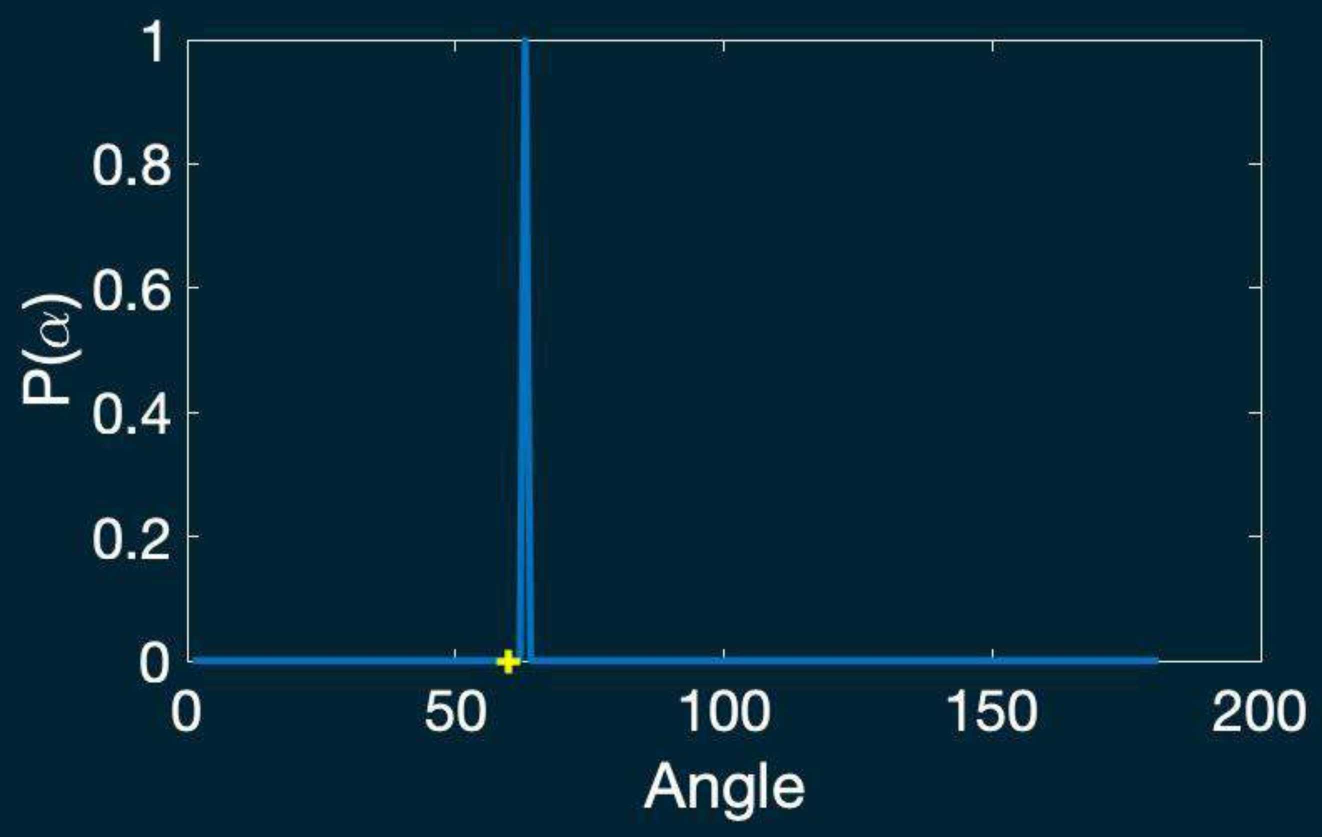
Evaluating Γ_ϕ is expensive: one of 5 parameters



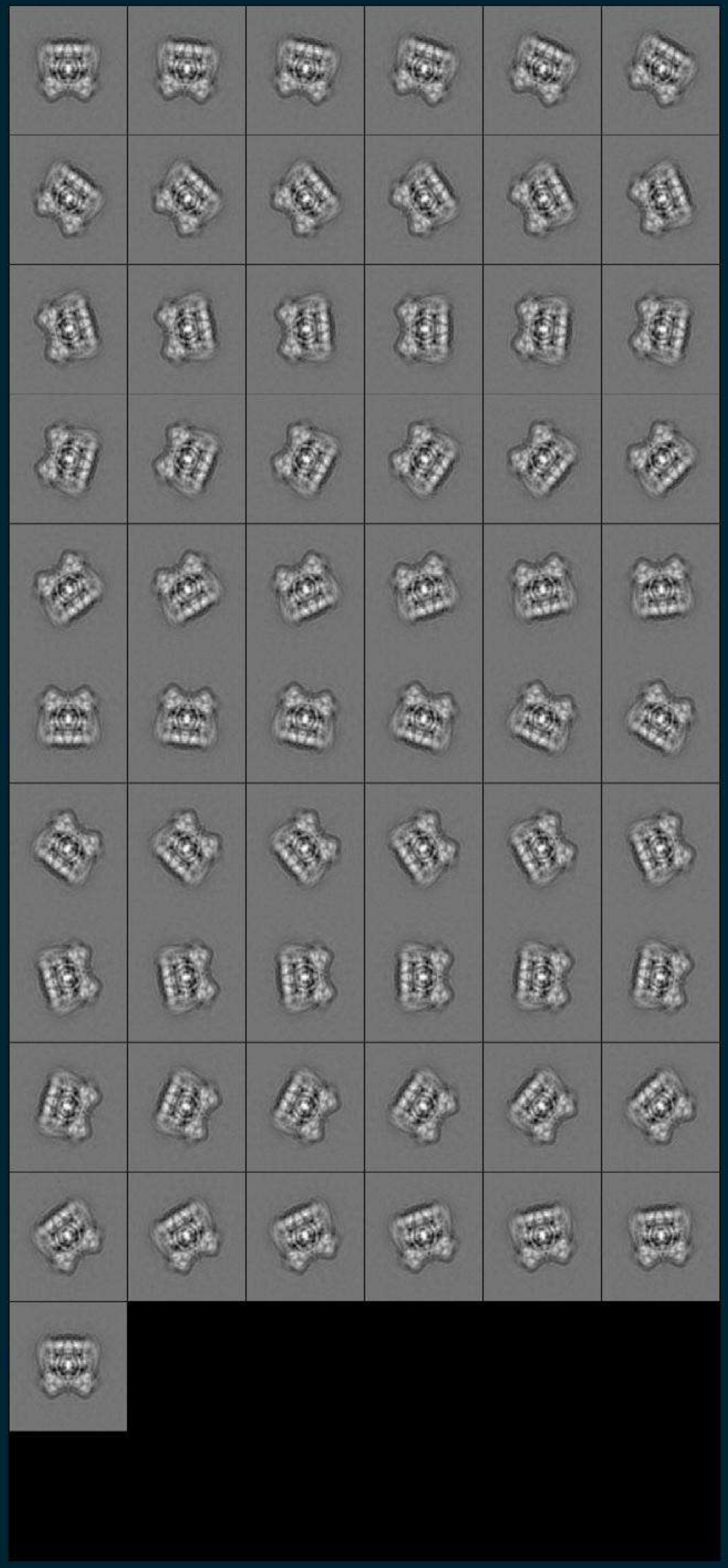


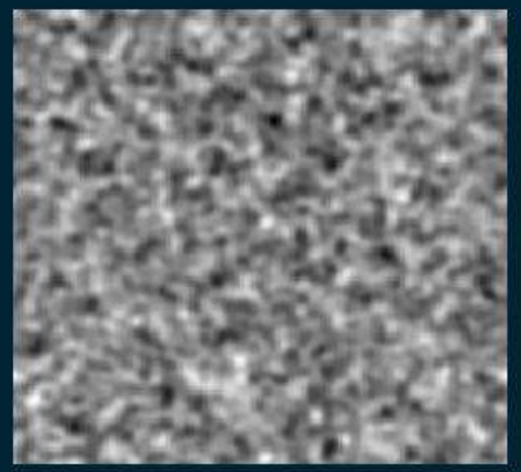


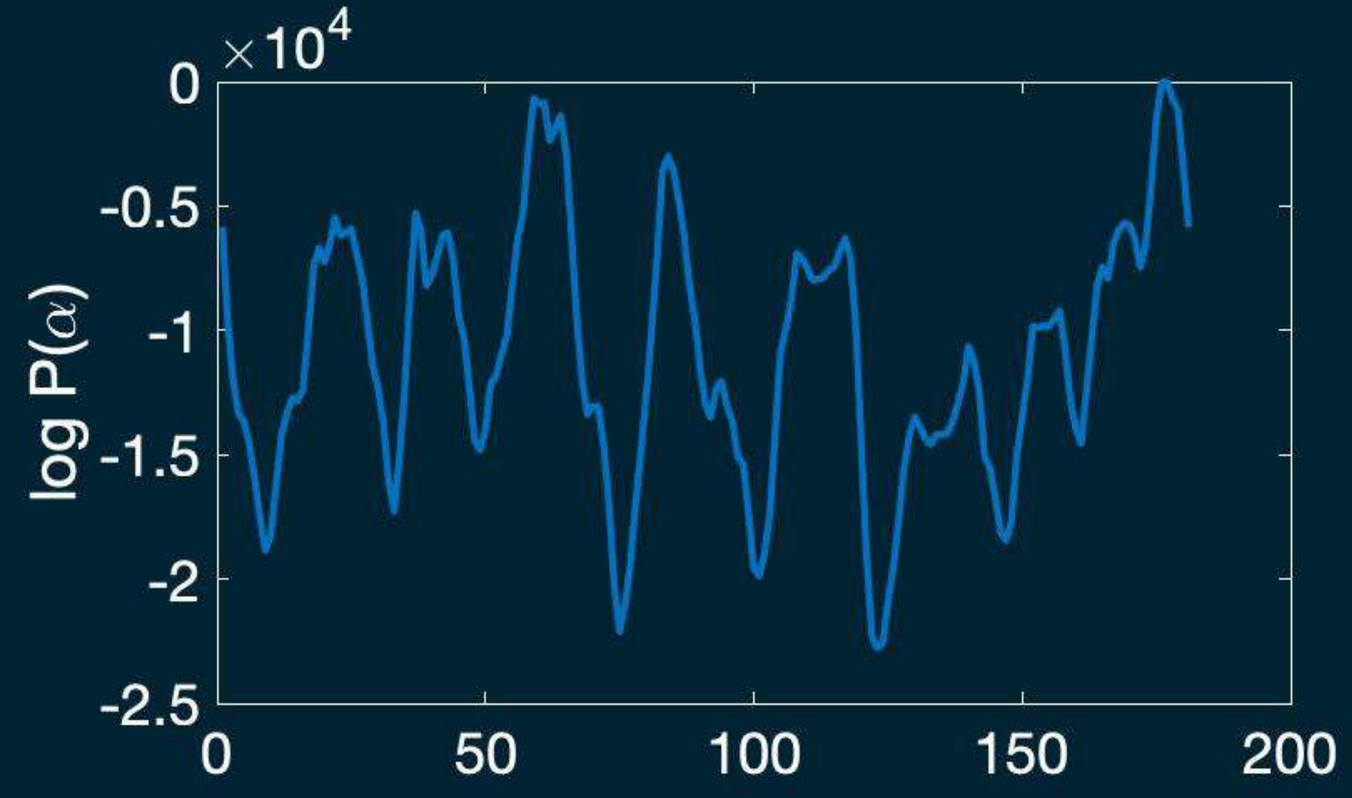


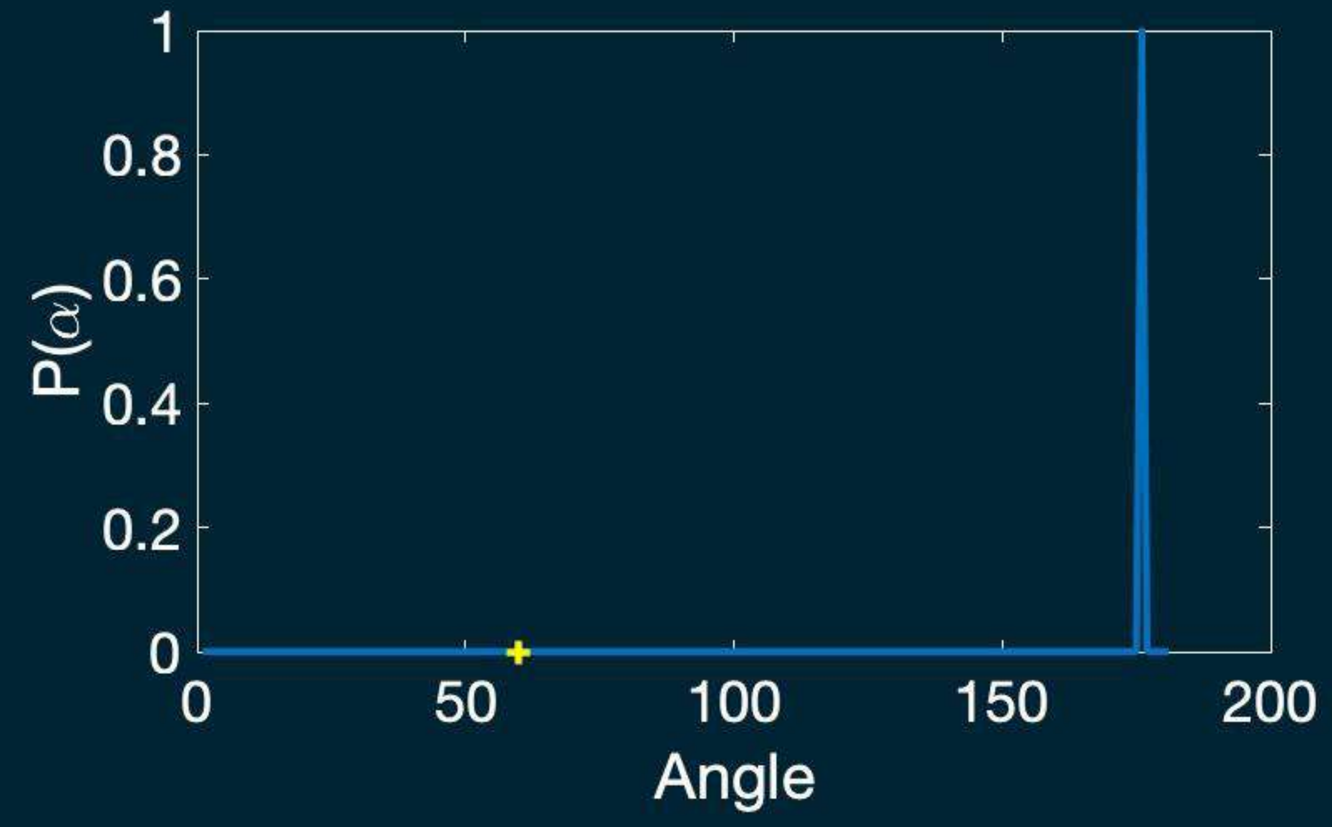


Evaluating Γ_ϕ is expensive: one of 5 parameters

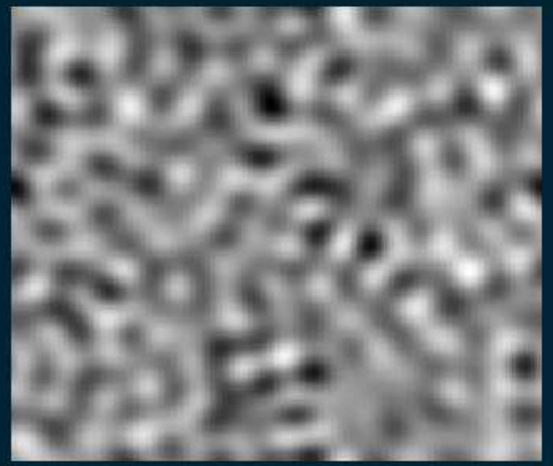


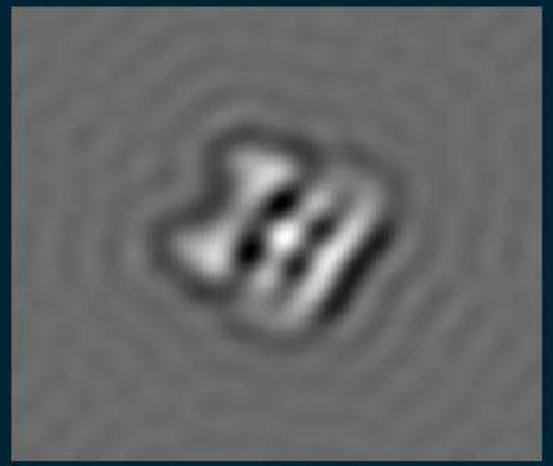




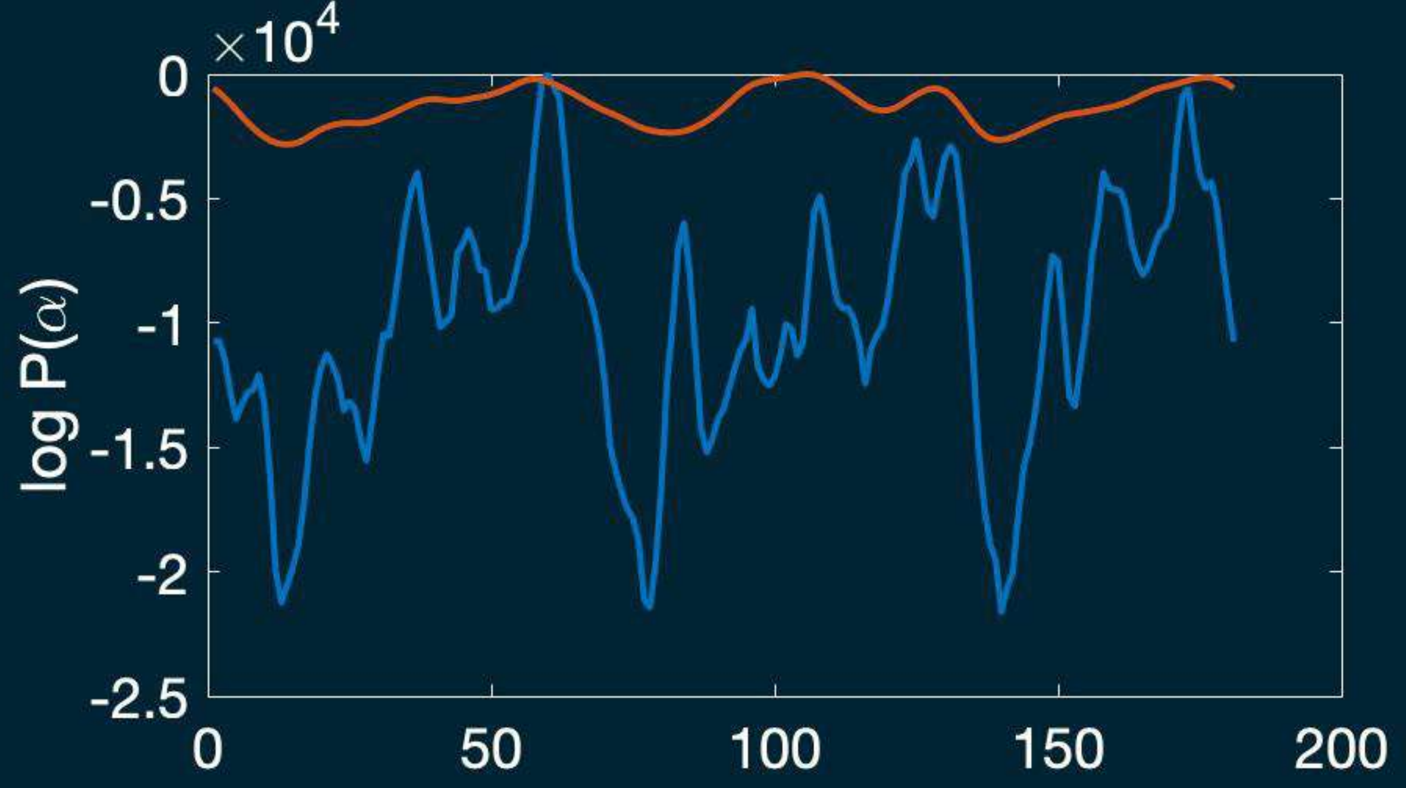


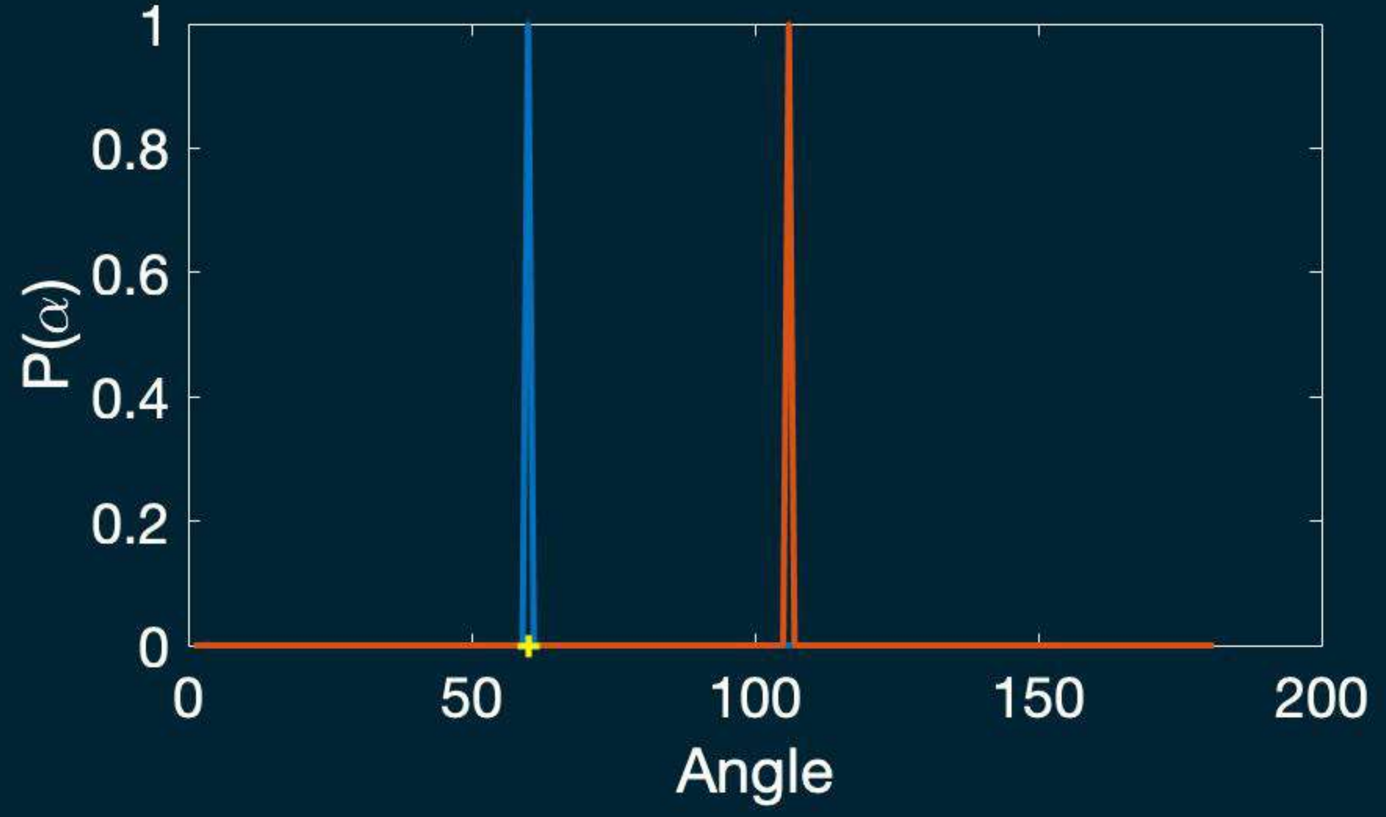
Domain reduction: branch and bound, illustrated for 1D





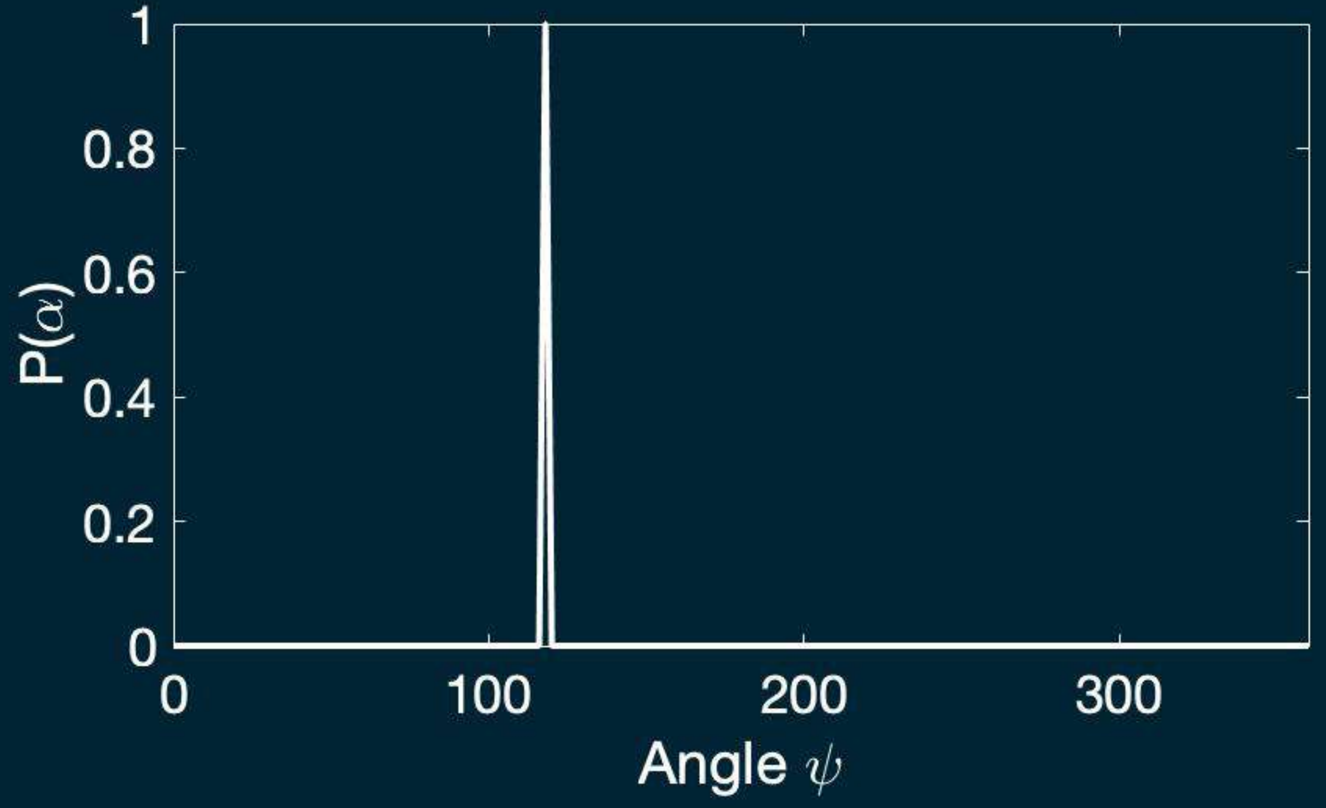
1. To save time, we compute probabilities of orientations at low resolution.

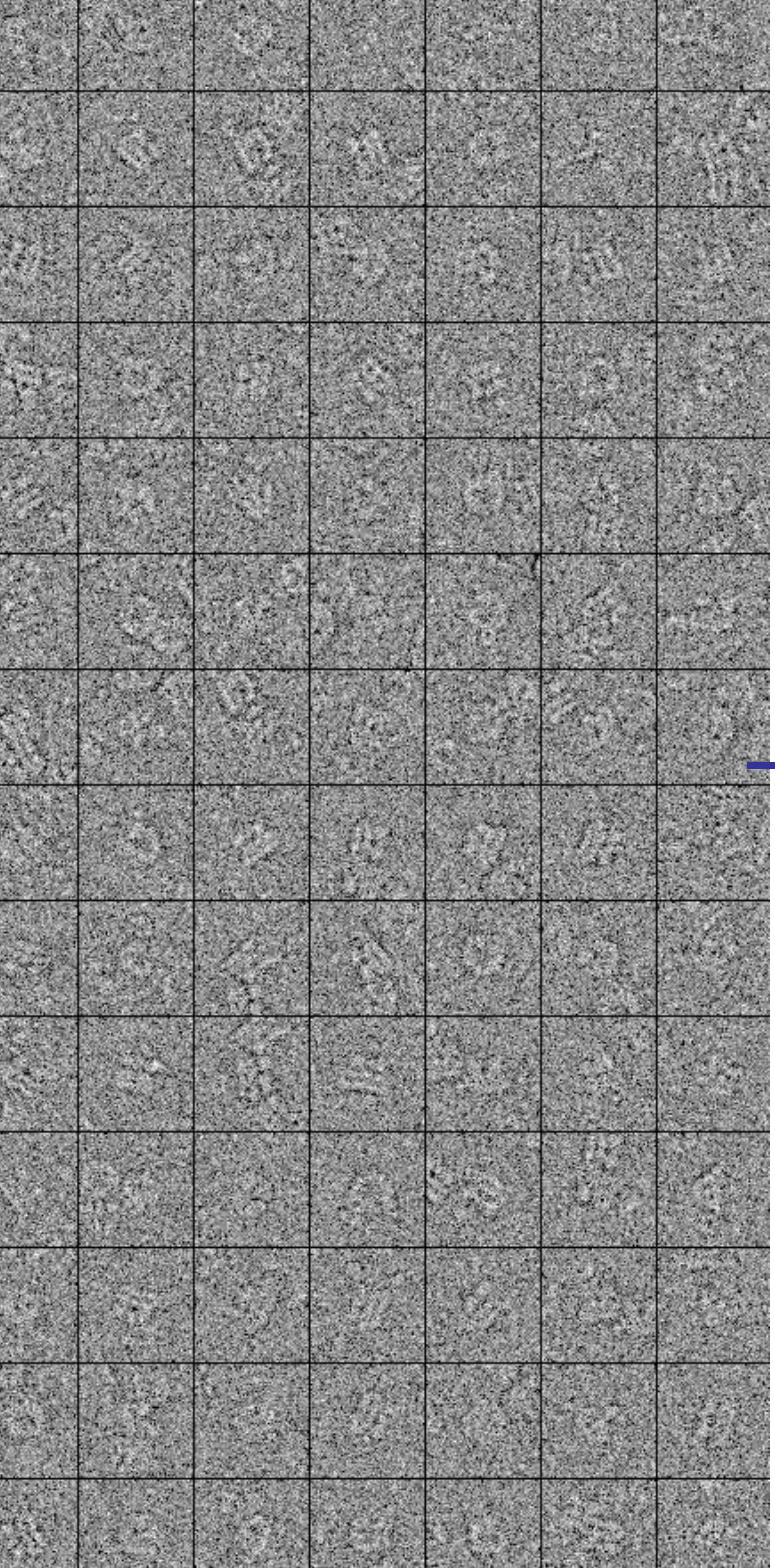


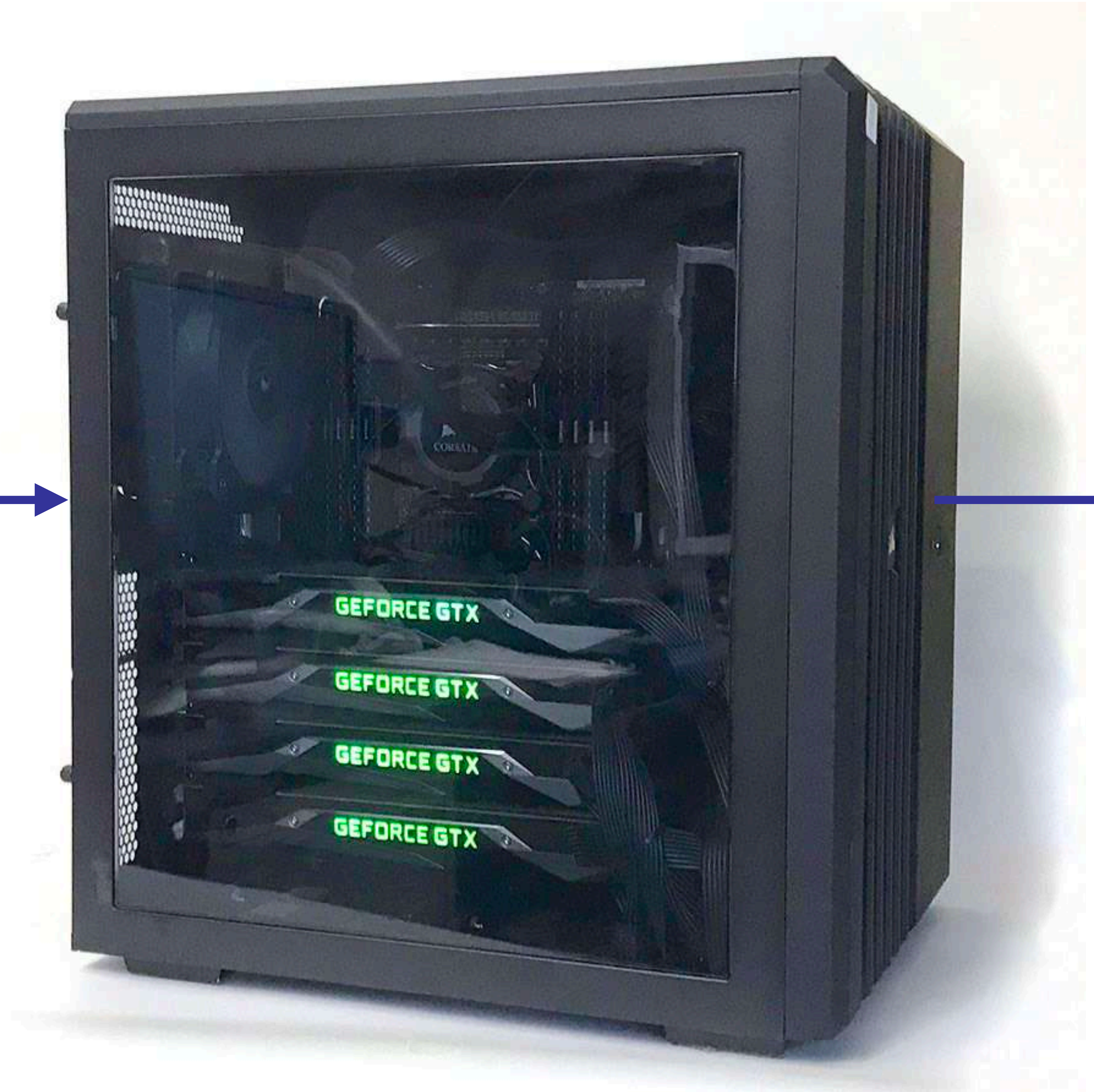


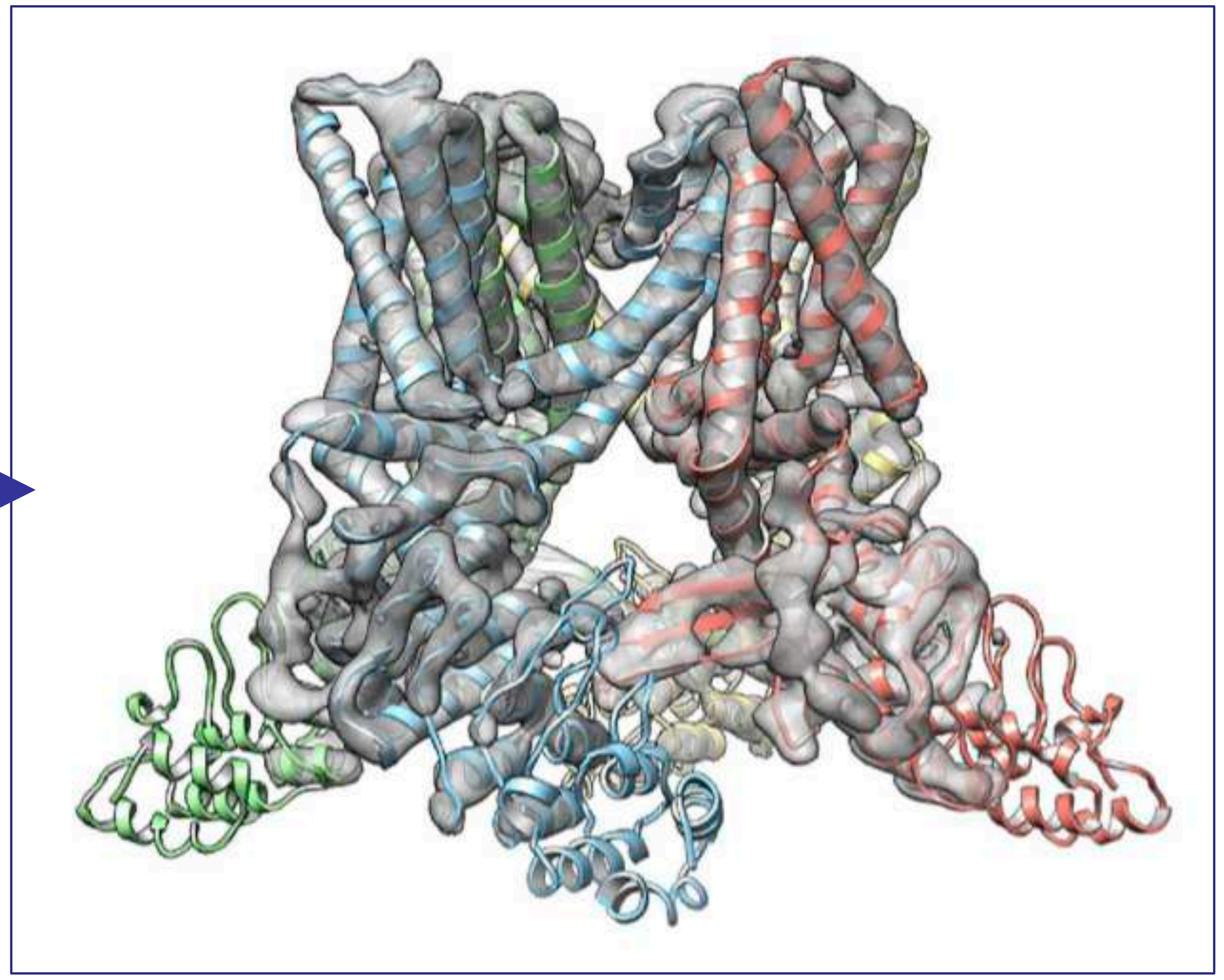
2. We place bounds on how much higher the probabilities could be at full resolution.

Given a cutoff value, we evaluate over a fraction of the domain.









Any sufficiently advanced technology is indistinguishable from magic.
-Arthur C. Clarke