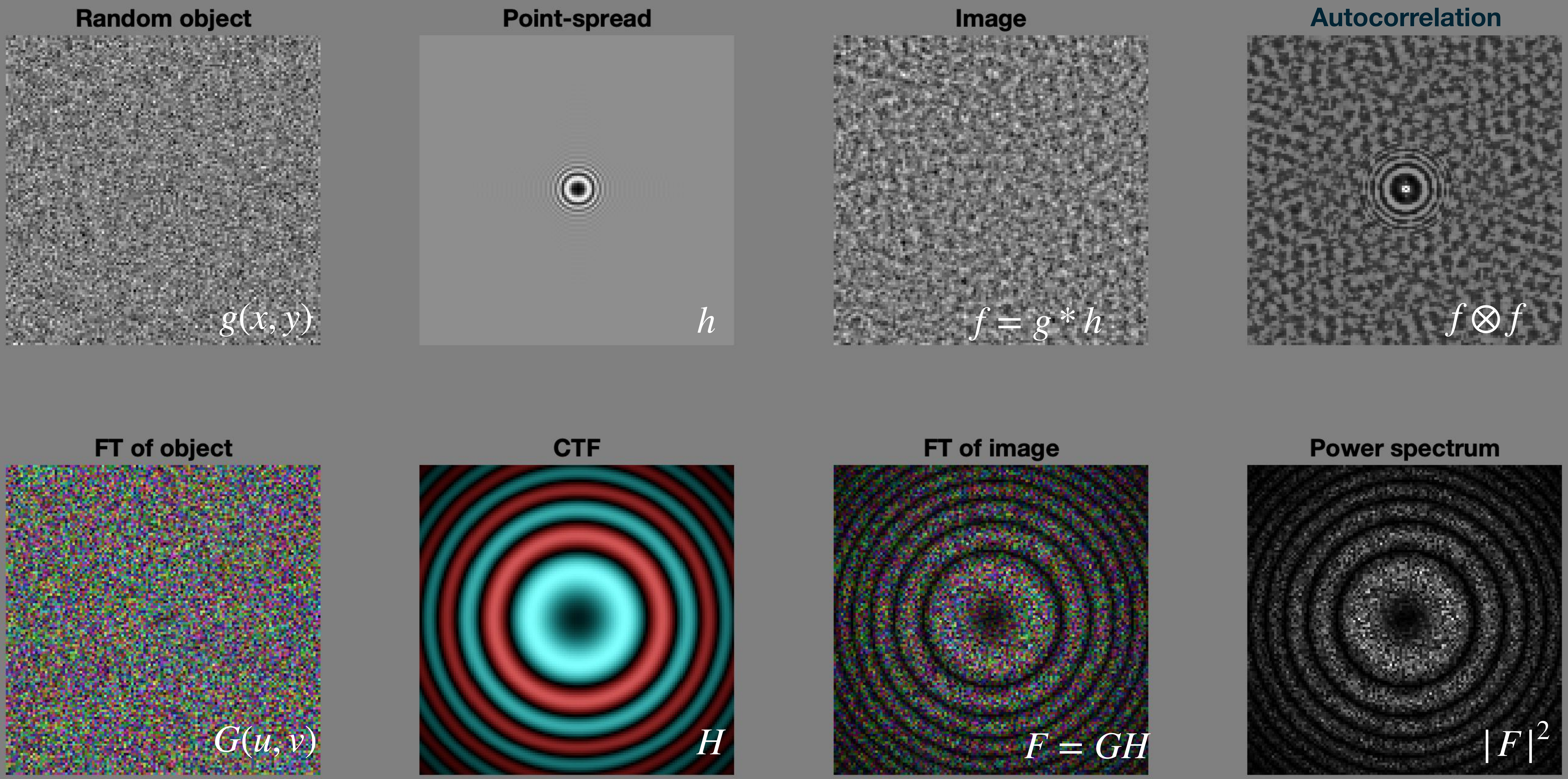
Lecture 4b
Algorithms and Foundational Math, Part 2

Single-particle reconstruction and classification

1. Undoing the CTF
2. Projection matching: Frealign
3. Maximum Likelihood: Relion and cryoSparc

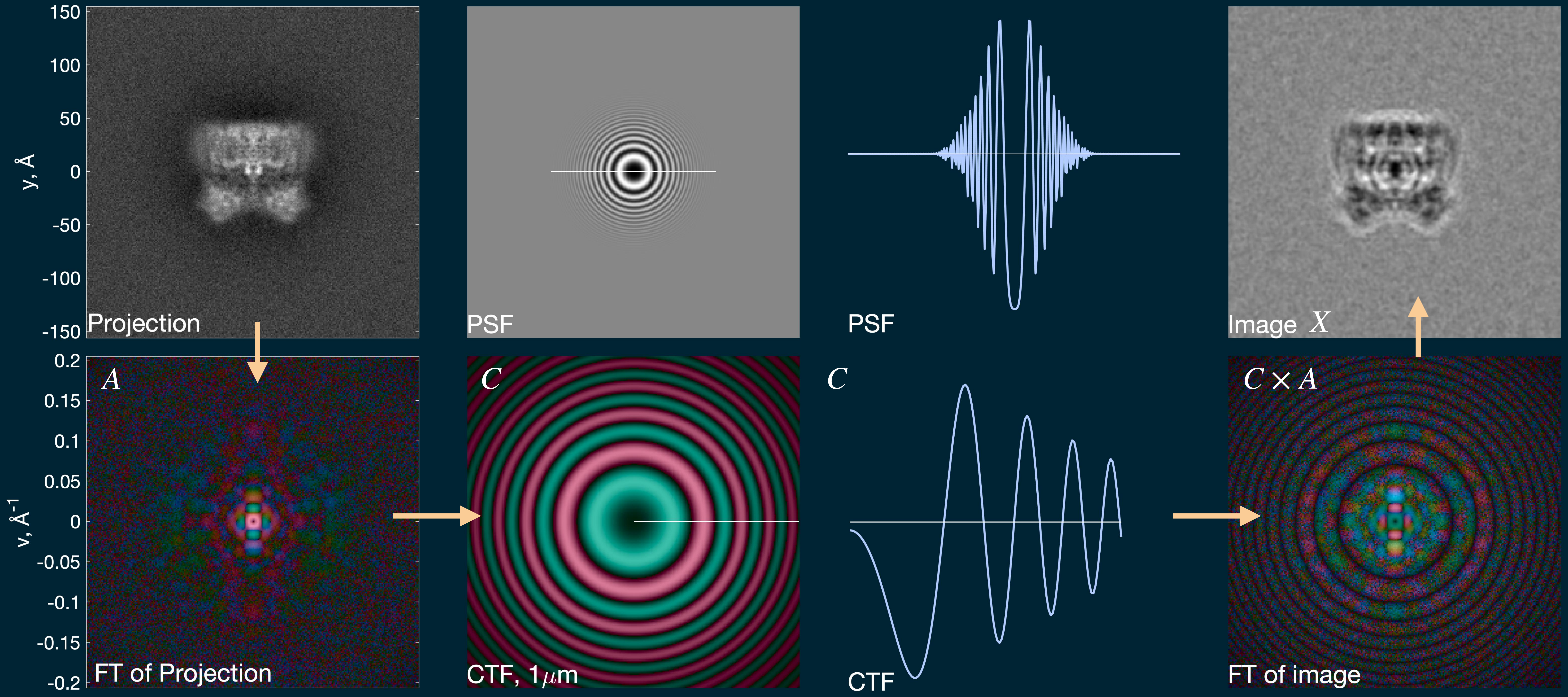
Visualizing the contrast transfer function



Modeling the CTF effect on an image

$$X = CA + N$$

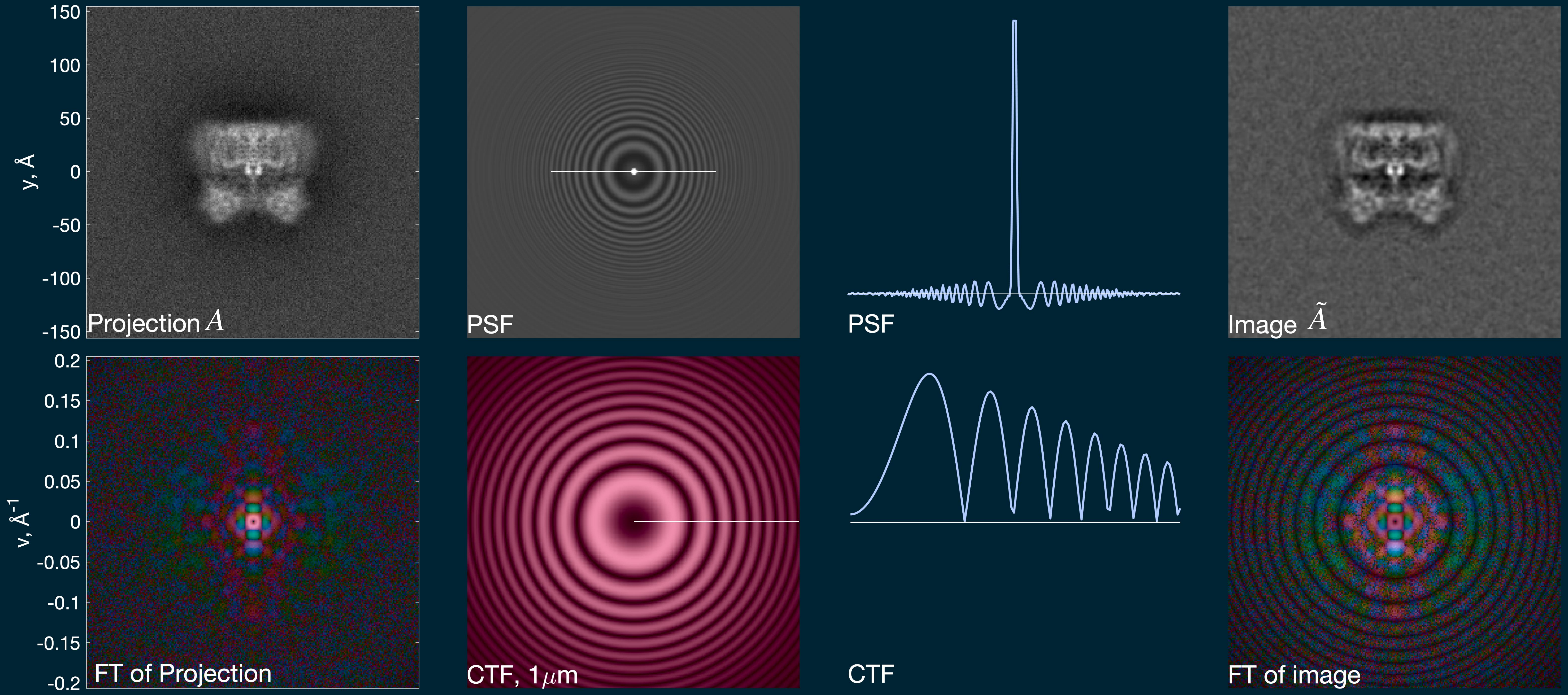
Can we do the deconvolution:
$\tilde{A} = X/C$??



How to undo the CTF effects?

1. **Phase flipping**

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



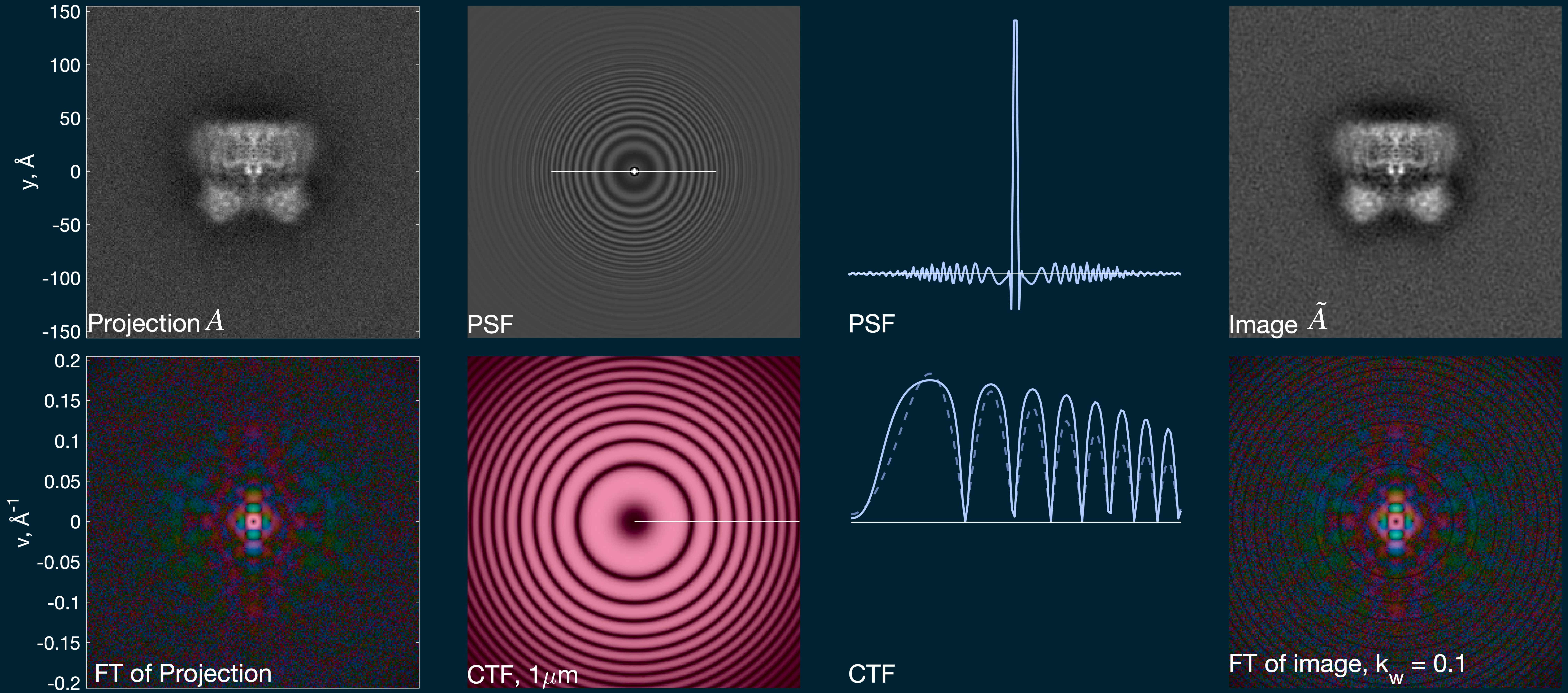
How to undo the CTF effects?

1. **Phase flipping**

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

2. **Wiener filter**

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



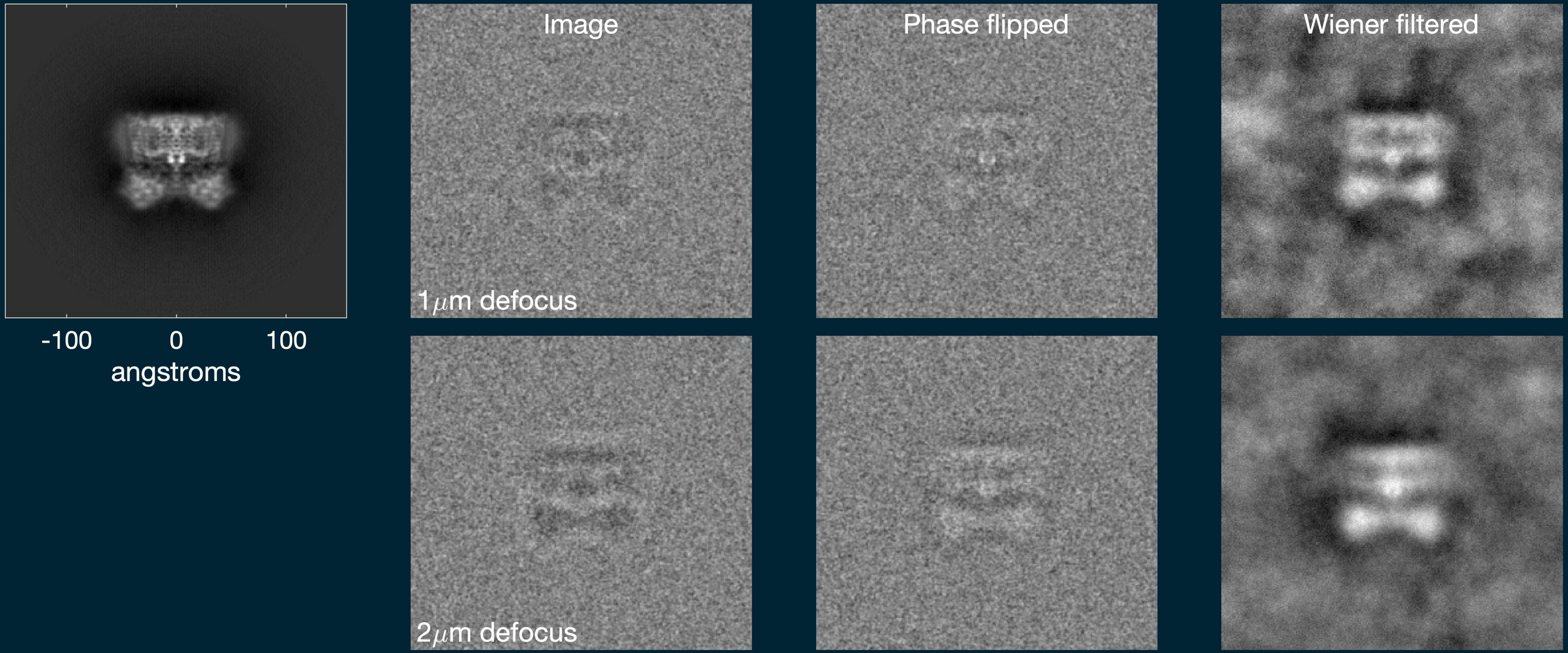
How to undo the CTF effects in noisy images?

1. **Phase flipping**

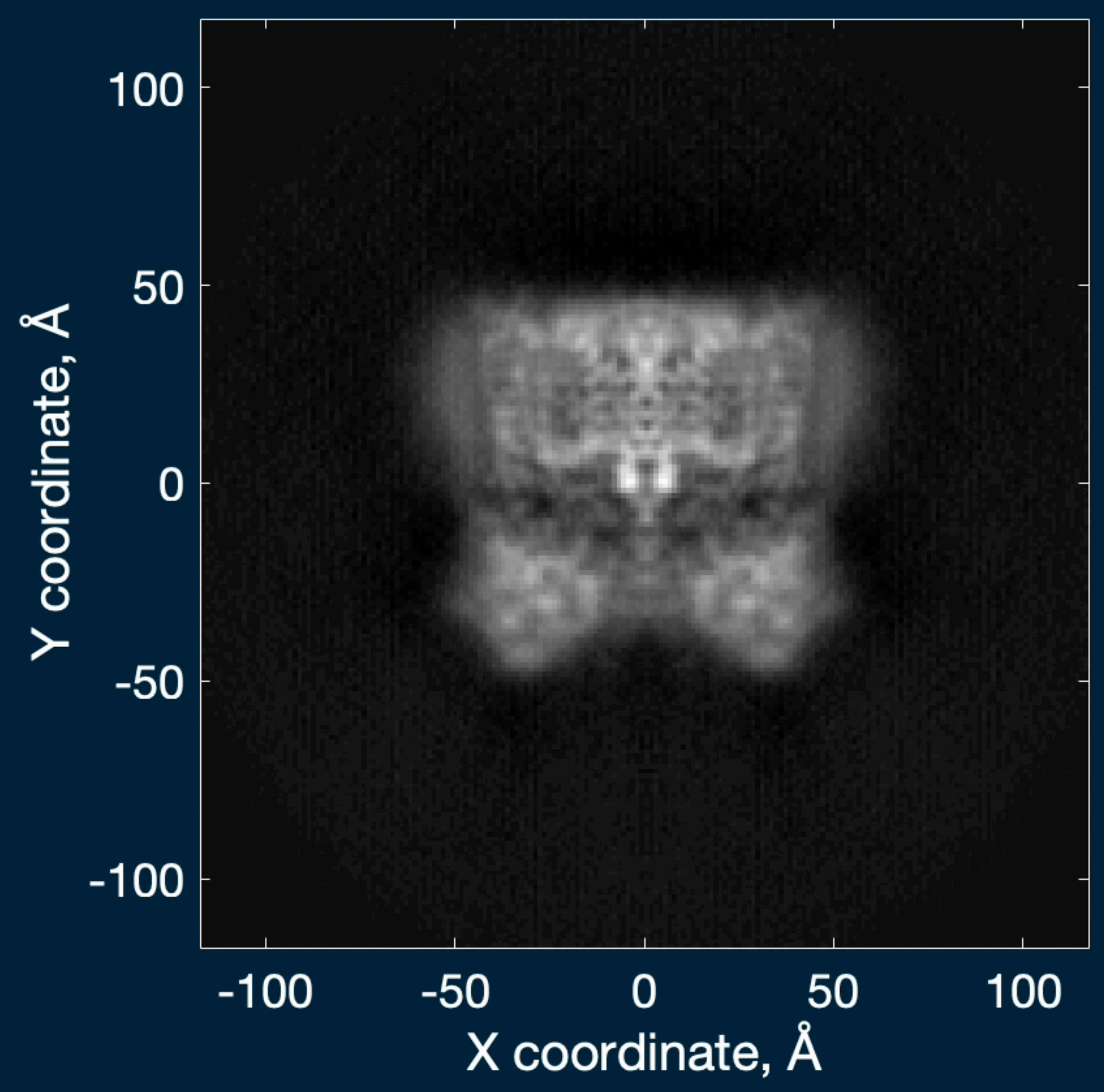
$$\tilde{A} = \mathrm{sgn}(C)X$$

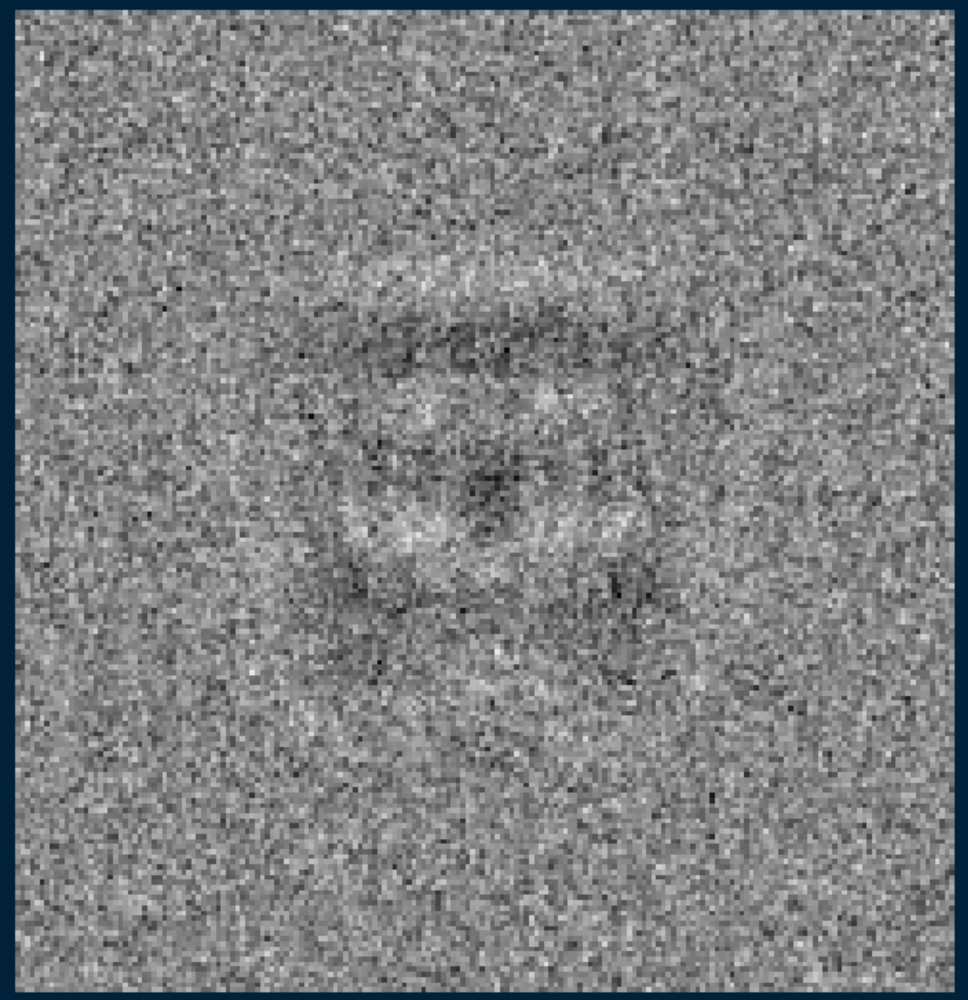
2. **Wiener filter**

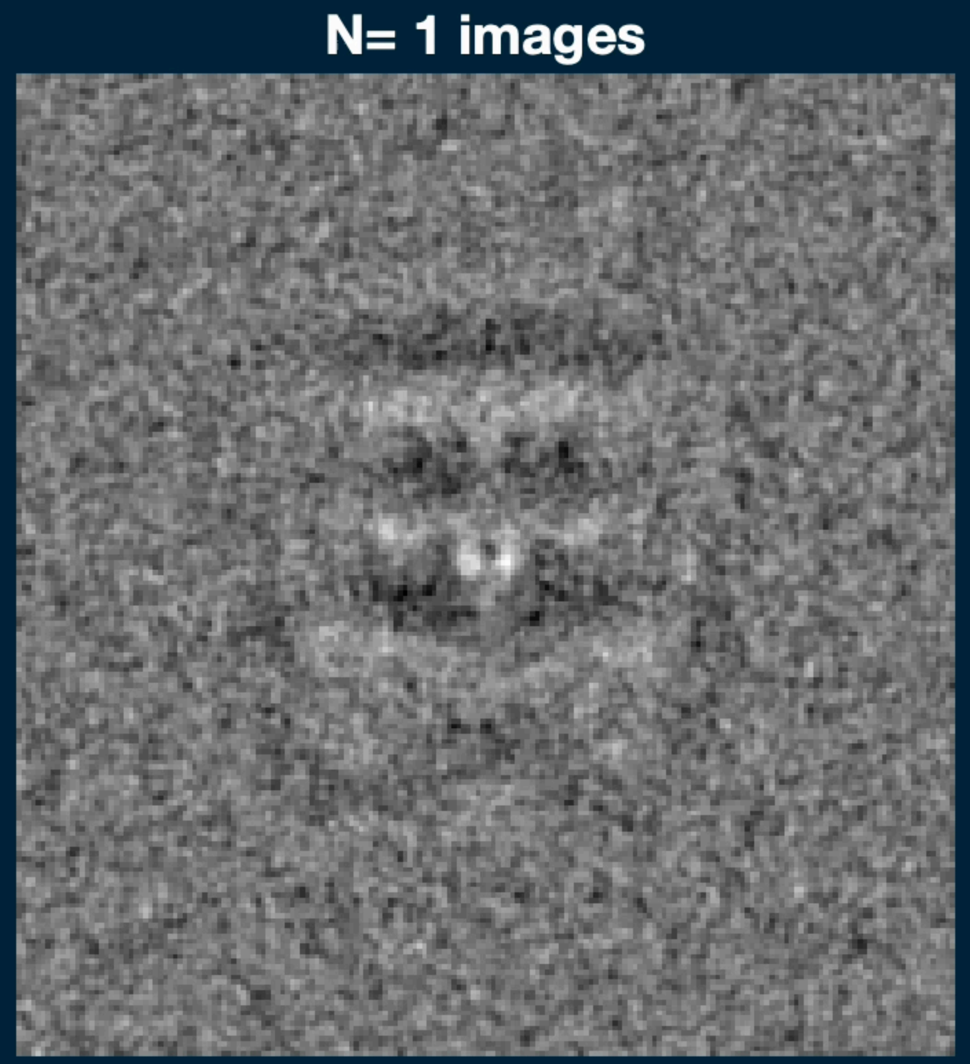
$$\tilde{A} = \frac{CX}{k + C^2}$$



How to undo the CTF effects in noisy images?



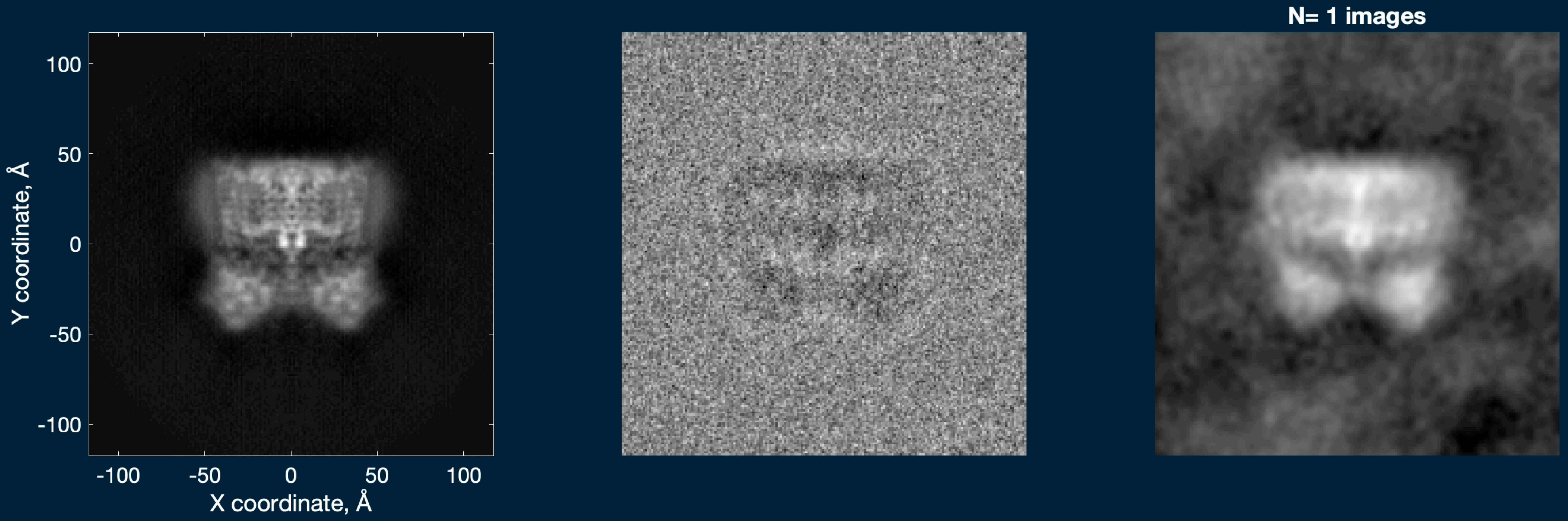




3. Wiener from multiple images

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

How to undo the CTF effects in noisy images?



3. Wiener from multiple images

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

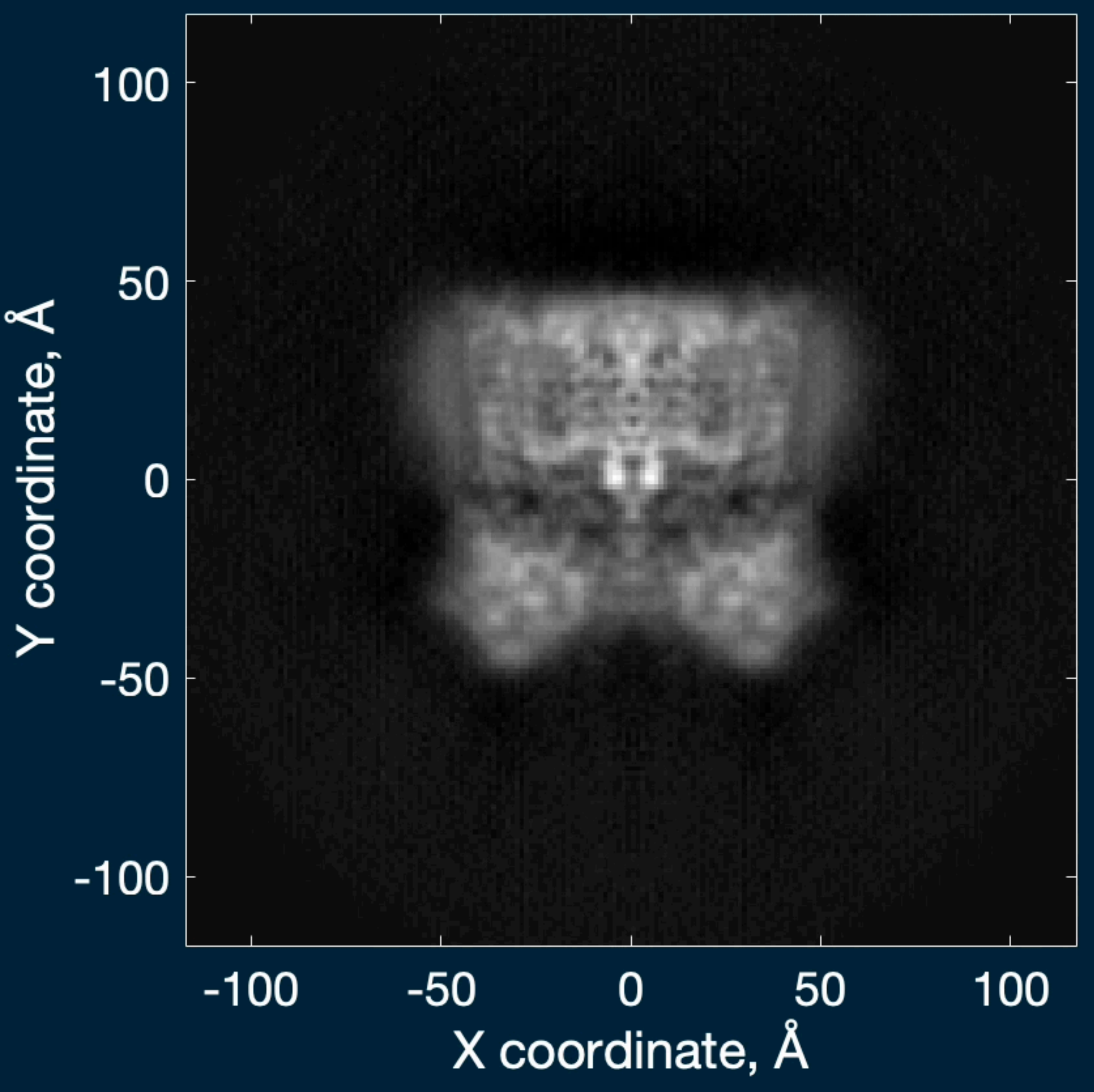
$$k(s) = 1/\mathrm{SNR} = \frac{|N|^2}{|A|^2}$$

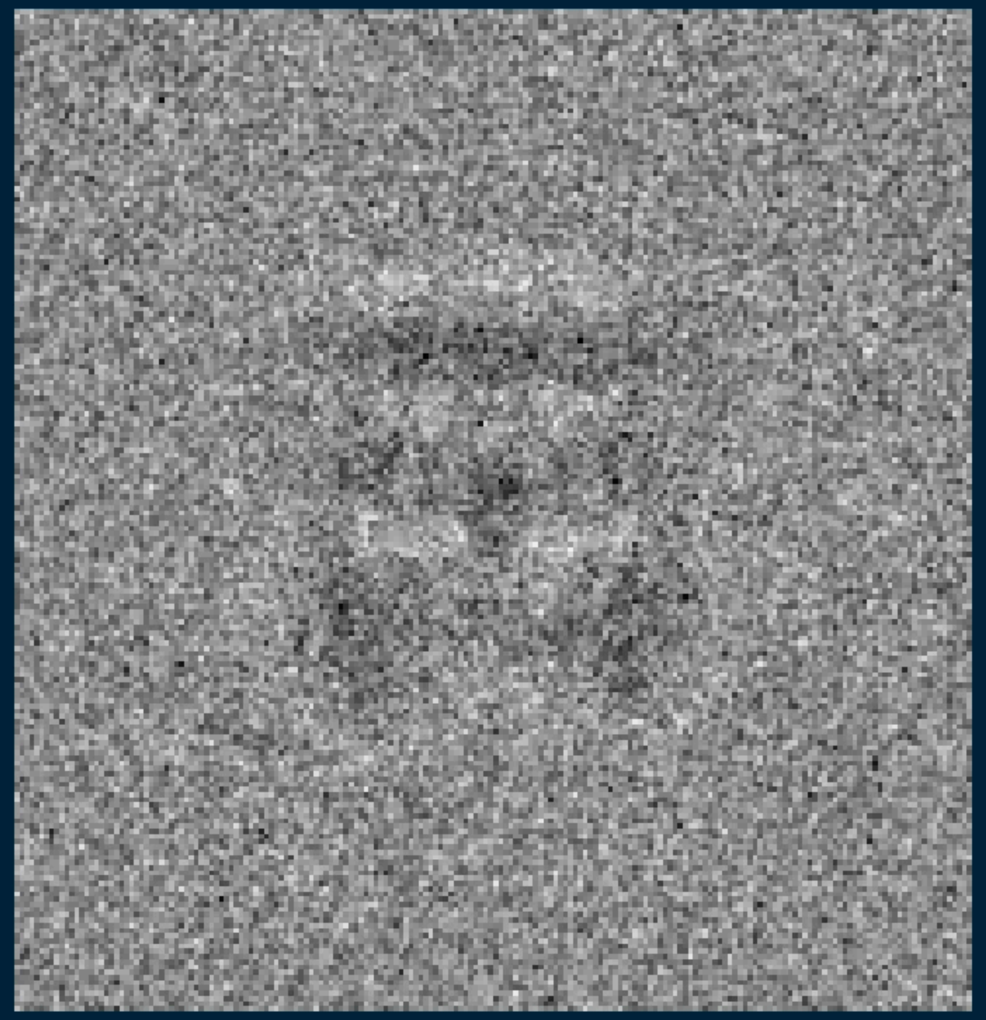
Image restoration when spectral SNR is known

Restoration from multiple images

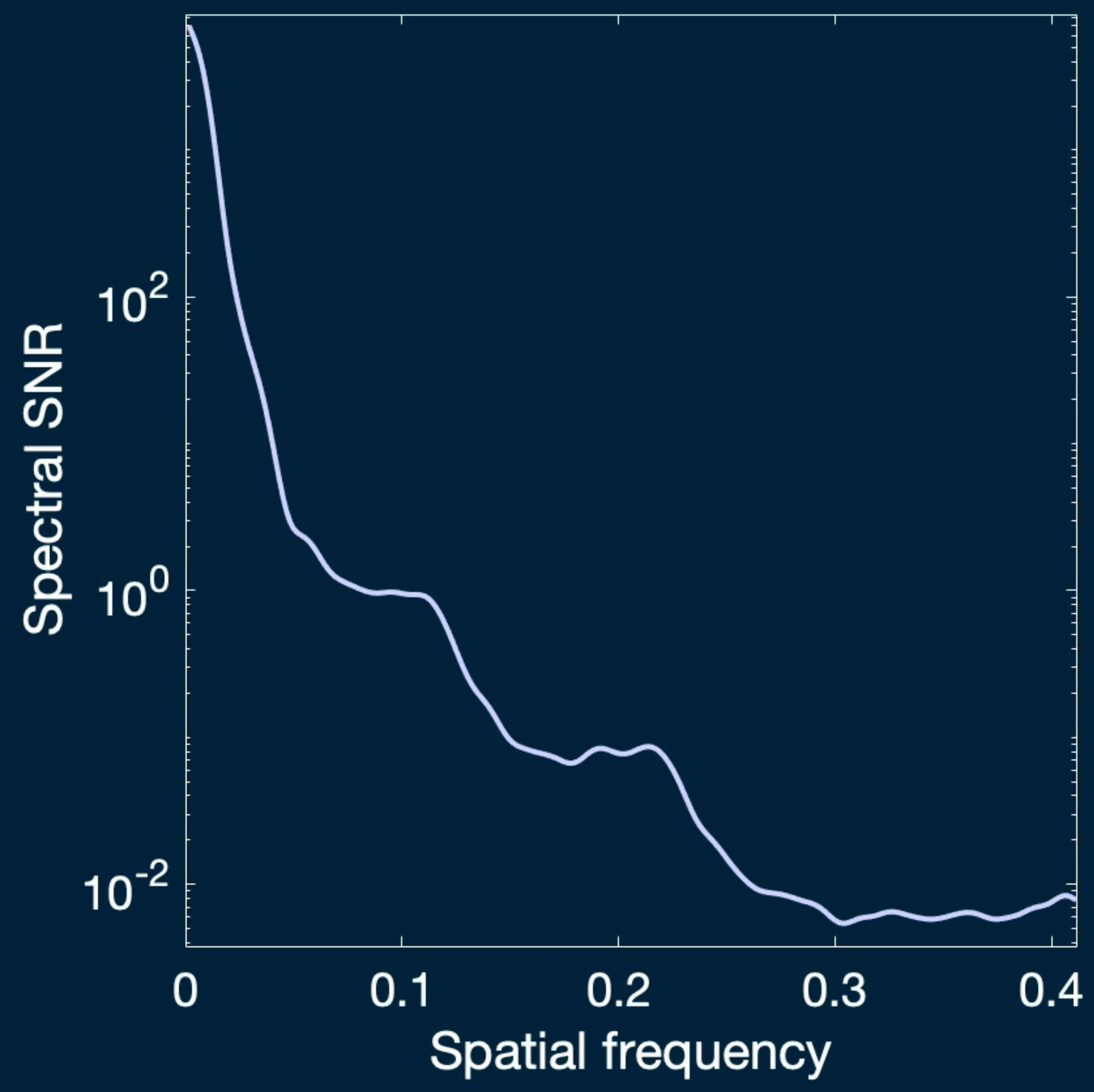
$$\tilde{A} = \frac{\sum_i^N C_i X_i}{\frac{1}{\mathrm{SSNR}} + \sum_i^N C_i^2}$$

The defocus varies to fill in CTF zeros





N= 1 images



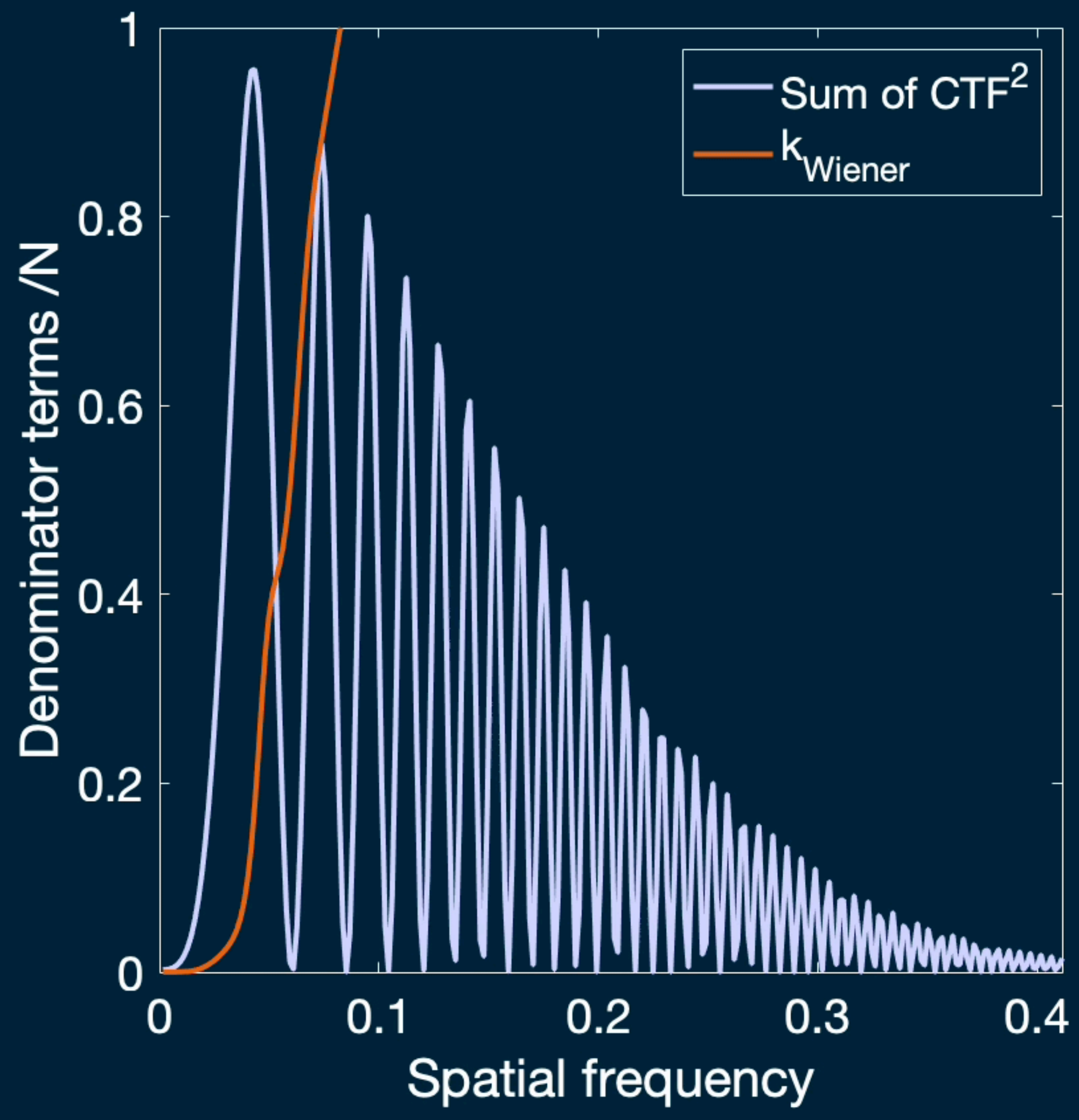
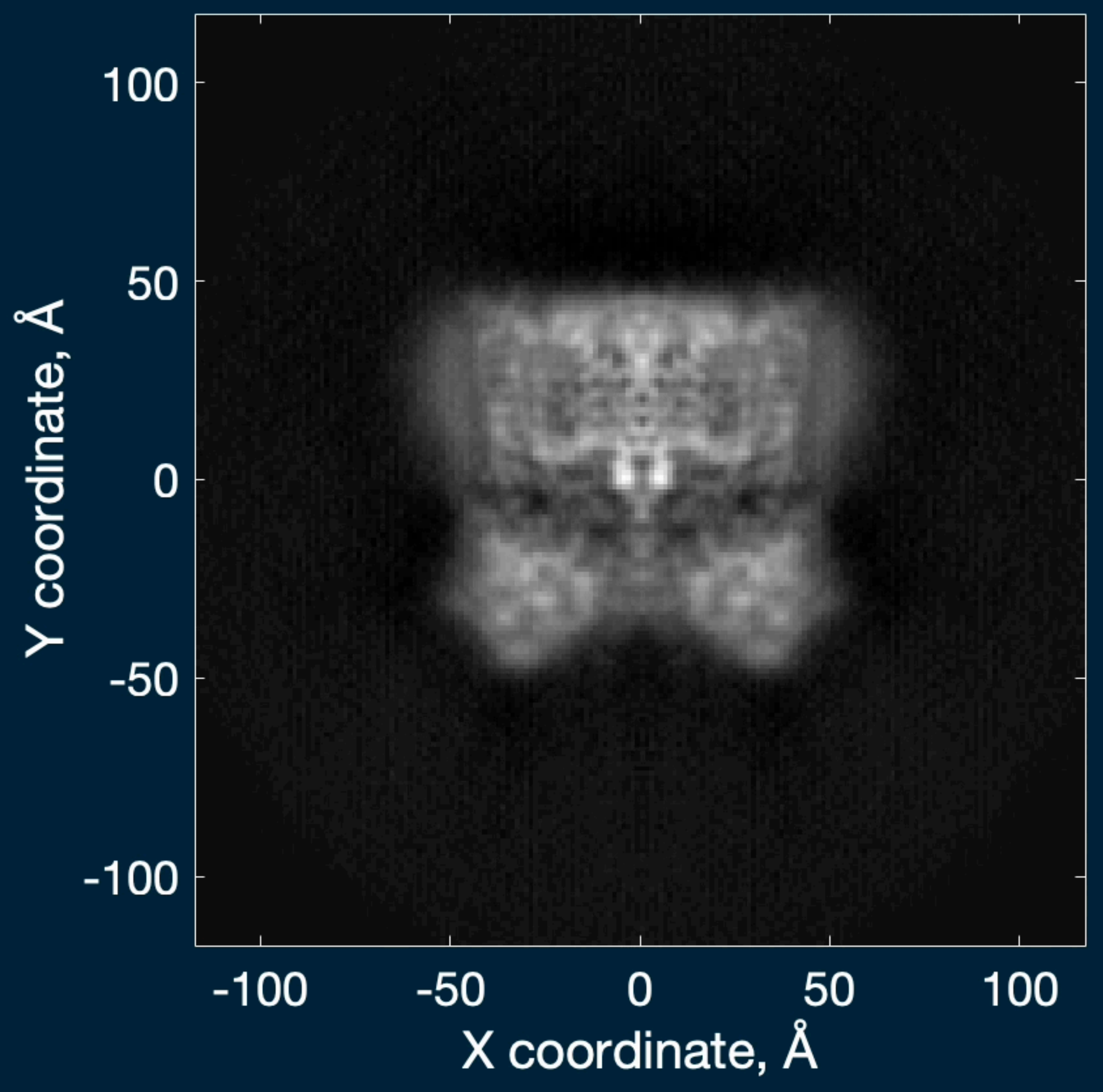


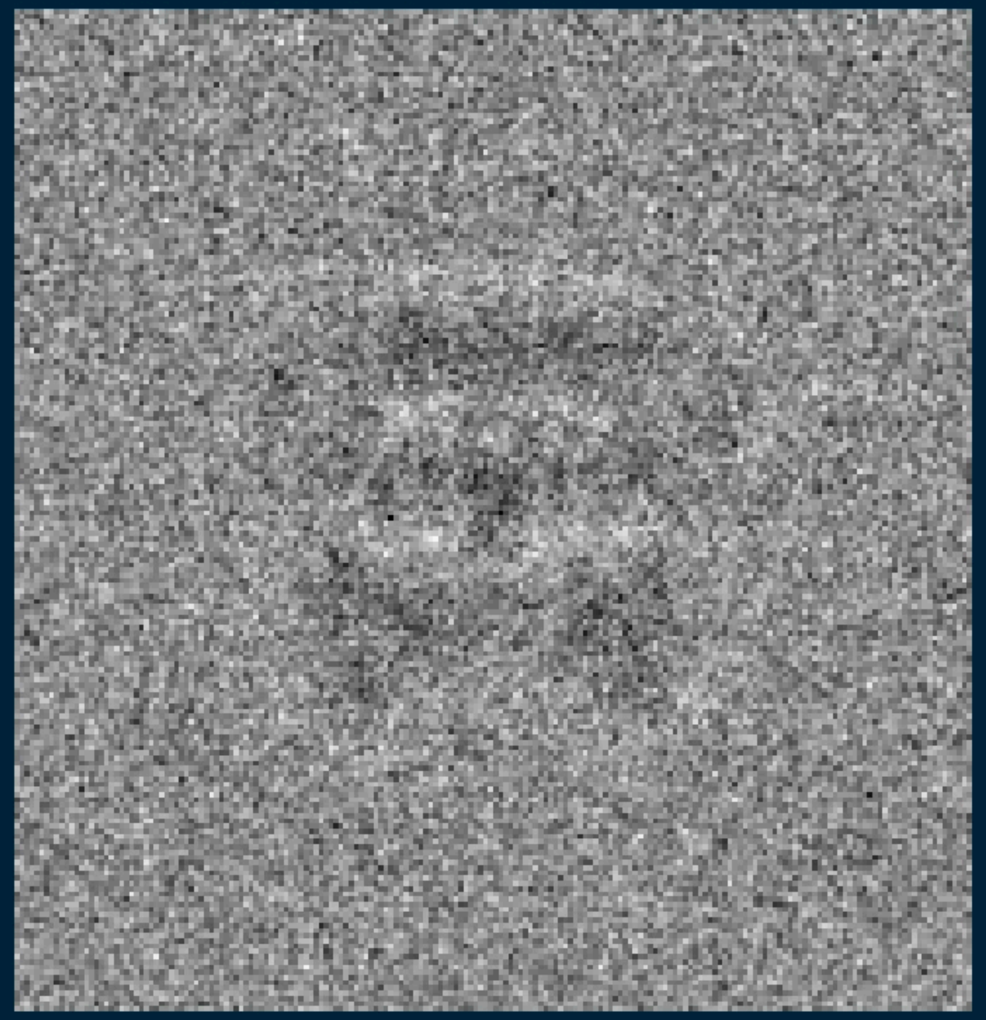
Image restoration when spectral SNR is known

Restoration from multiple images

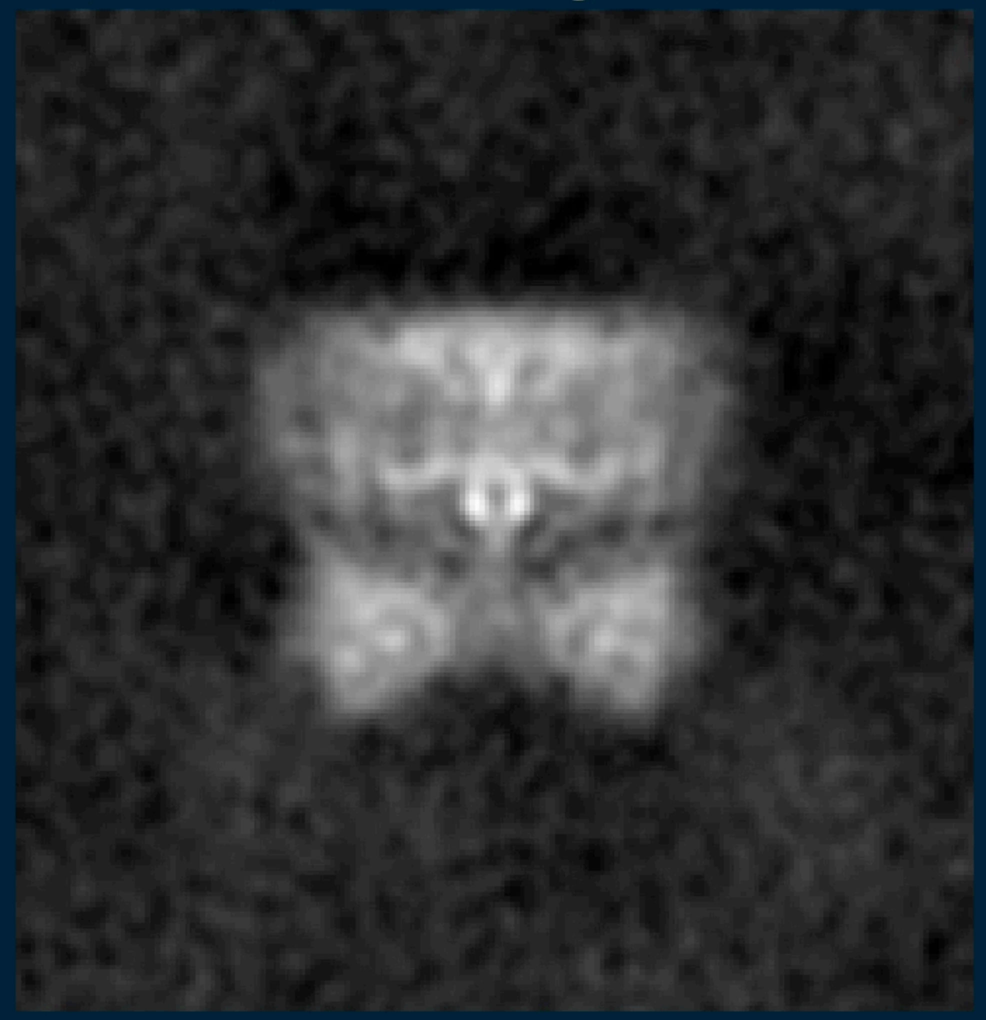
$$\tilde{A} = \frac{\sum_i^N C_i X_i}{\frac{1}{\mathrm{SSNR}} + \sum_i^N C_i^2}$$

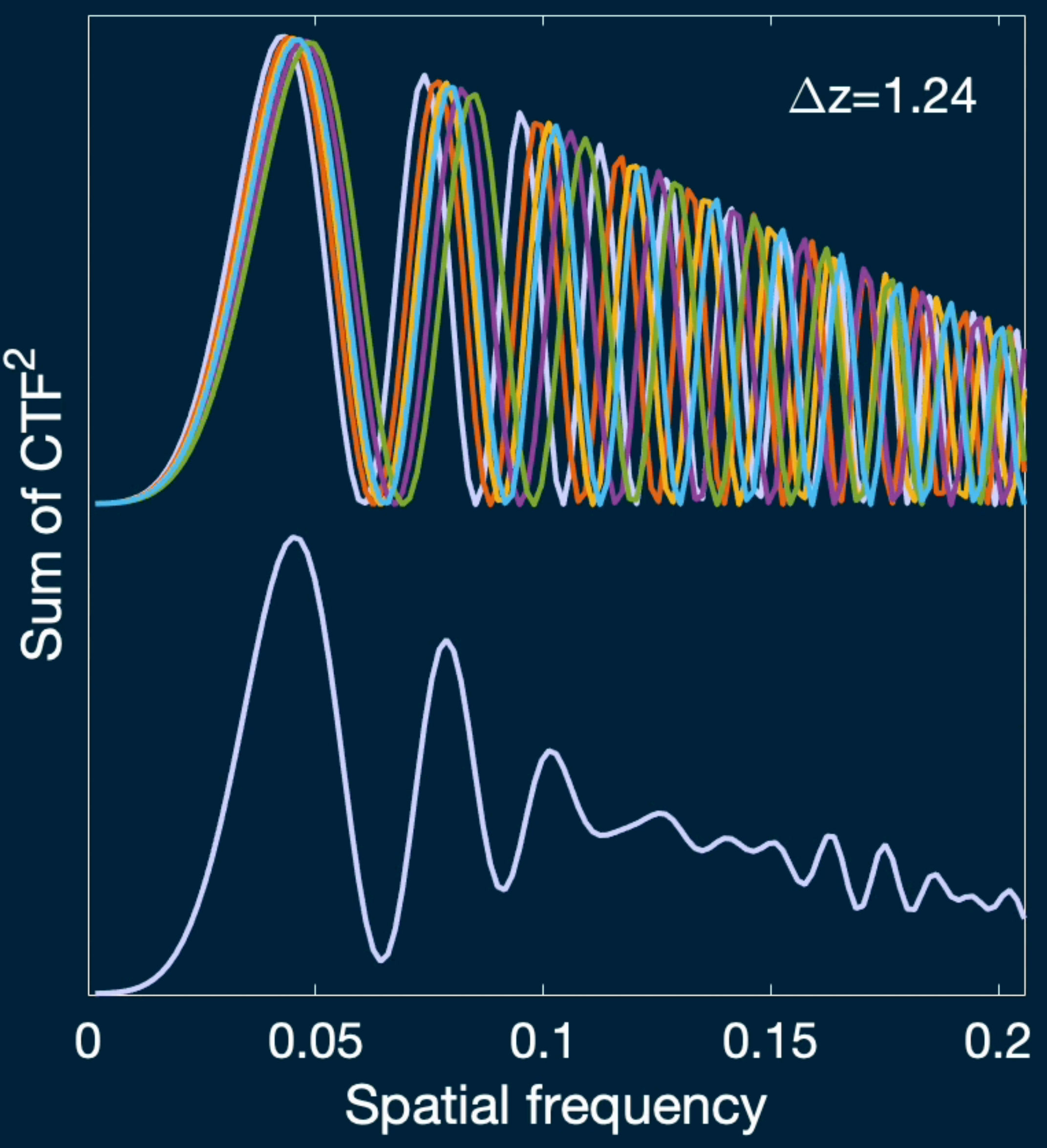
The defocus varies to fill in CTF zeros

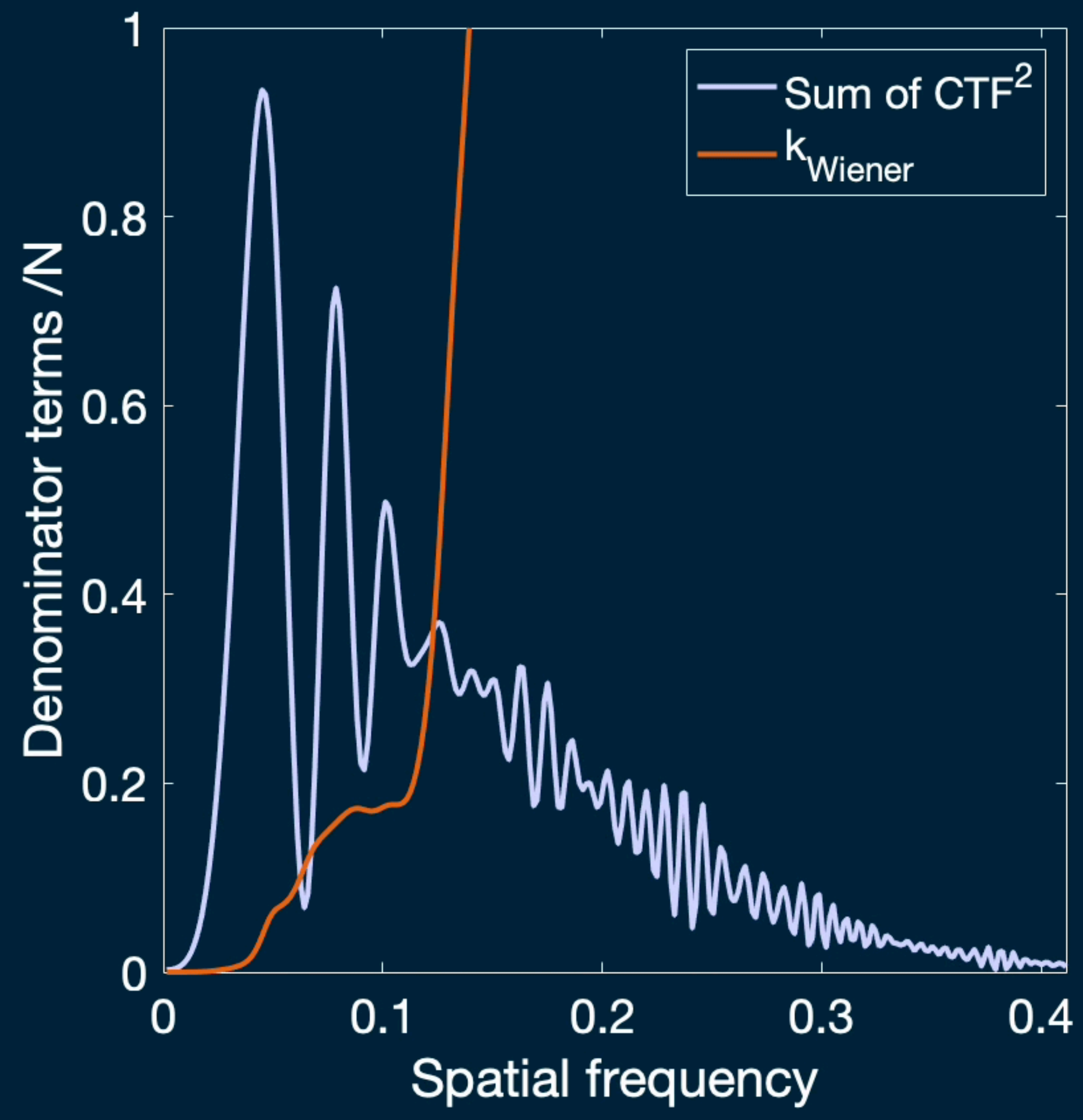




N= 6 images





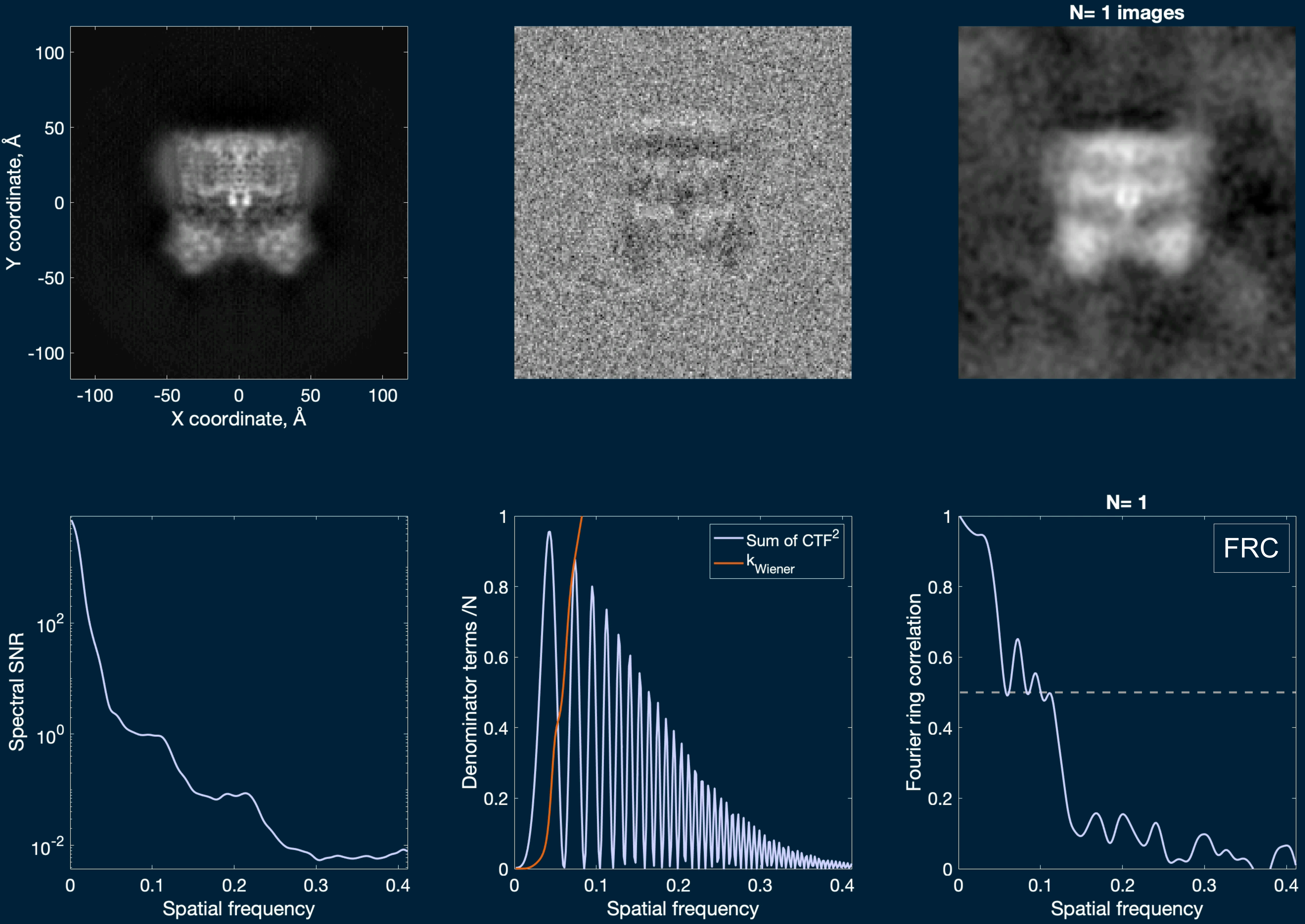


Estimate the resolution with Fourier ring correlation

Restoration from multiple images

$$\tilde{A} = \frac{\sum_i^N C_i X_i}{\frac{1}{\text{SSNR}} + \sum_i^N C_i^2}$$

The defocus varies to fill in CTF zeros



Single-particle reconstruction and classification

1. Undoing the CTF
2. Projection matching: Frealign
3. Maximum Likelihood: Relion and cryoSparc

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:

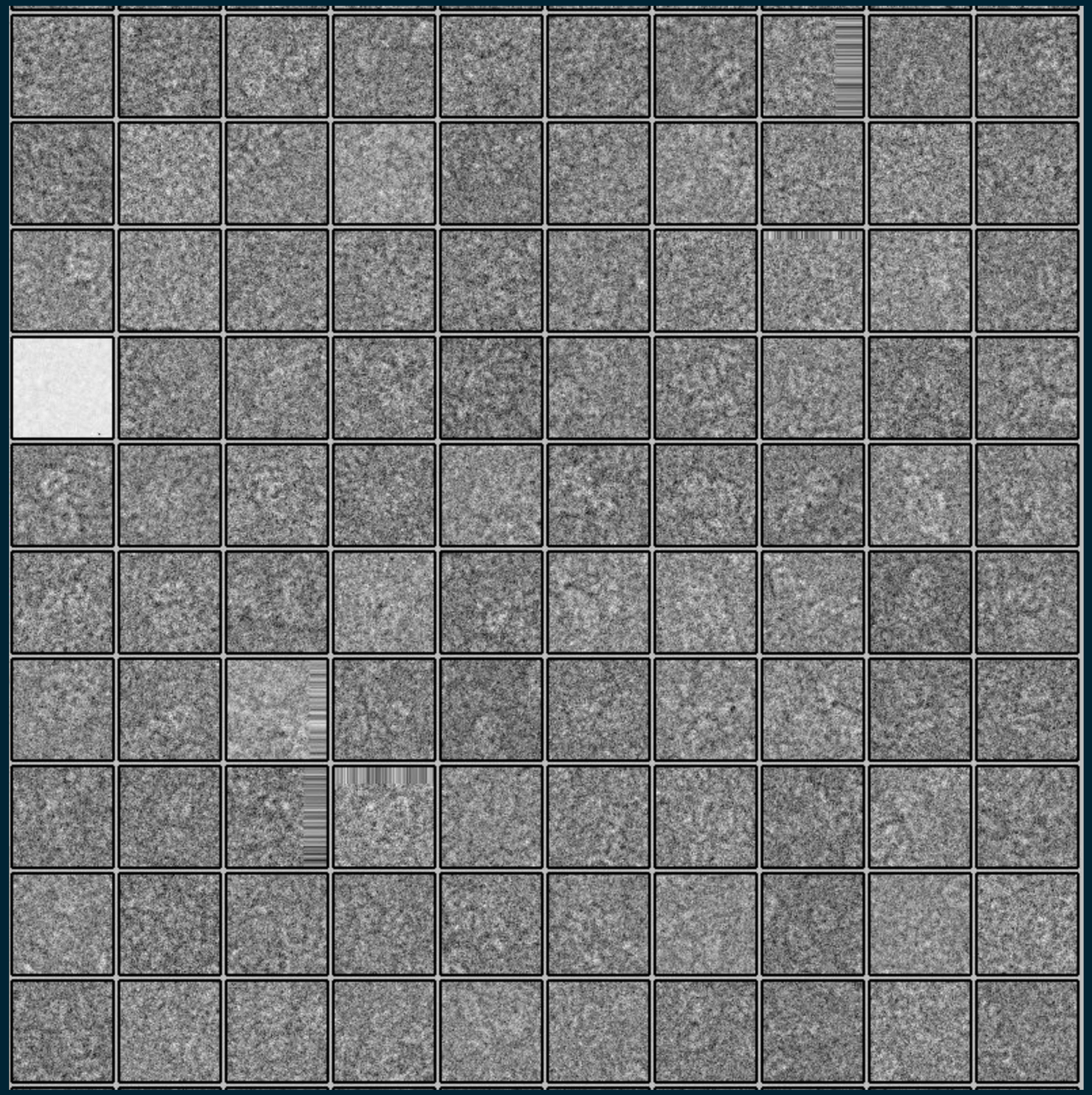
X_j —the j^{th} pixel (out of J pixels total)

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

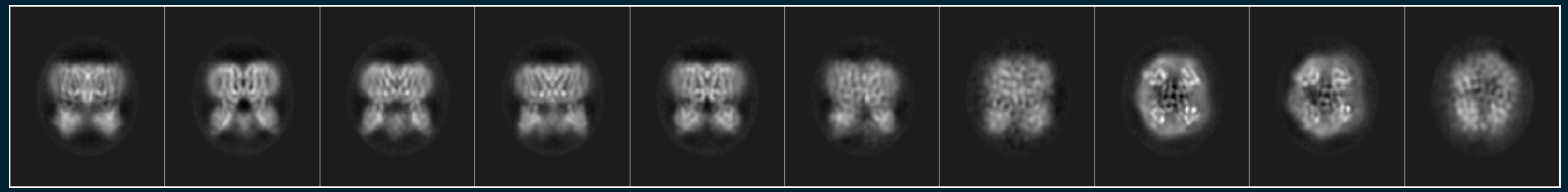
X_i

2D Classification

$X_i,\ i = 1..N$



$A_k,\ k = 1..10$



We assume that image X_i comes from a rotated and translated copy of A_k, one of K "class images":

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

Here,

- C_i is the CTF of the image
- $\mathbf{P}_{\phi_i}$ is the rotation/translation operator for the i^{th} image, where $\phi_i = \{\psi, t_x, t_y\}_i$.
- N_i is the 2D noise field

2D classification by template matching

Given estimates of $A_1 \ldots A_K$ from the n^{th} iteration,

1. Vary the rotations and translations ϕ_i and the class index k_i to find the projection image $R_i = C_i \mathbf{P}_{\phi_i} A_{k_i}$ 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 rotation and translation ϕ_i and the index k_i for each image X_i, update the k^{th}estimate according to

$$A_k^{(n+1)} = \frac{\sum_{\{i'\}} \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i X_i}{\frac{1}{\mathrm{SSNR}} + \sum_{\{i'\}} \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i^2}$$

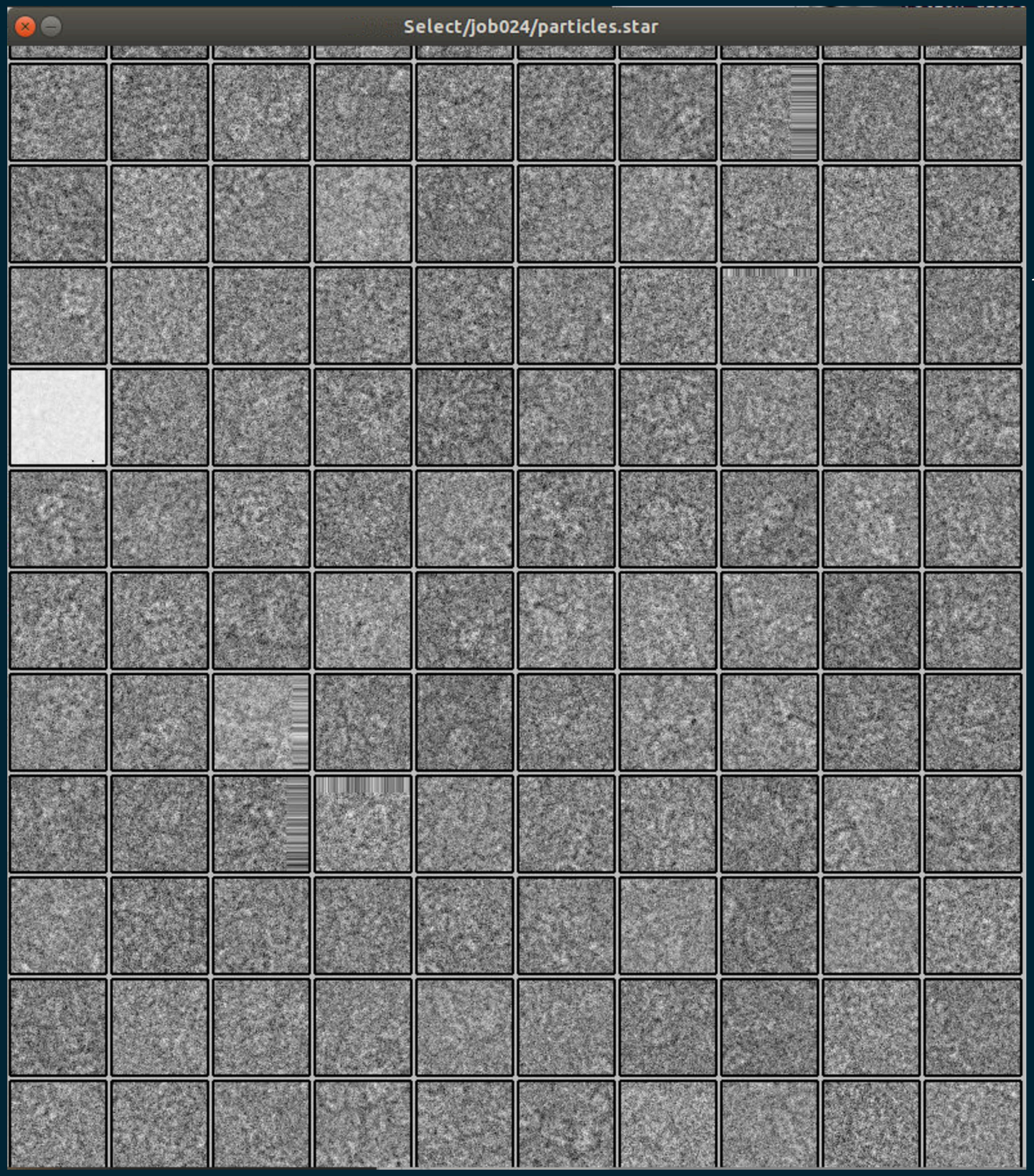
where $\{i'\}$ is the set of i values for which $k_i = k$, and $\mathbf{P}_{\phi_i}^{\mathbf{T}}$ is the inverse of the transformation $\mathbf{P}_{\phi_i}$.

Single-particle reconstruction

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

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

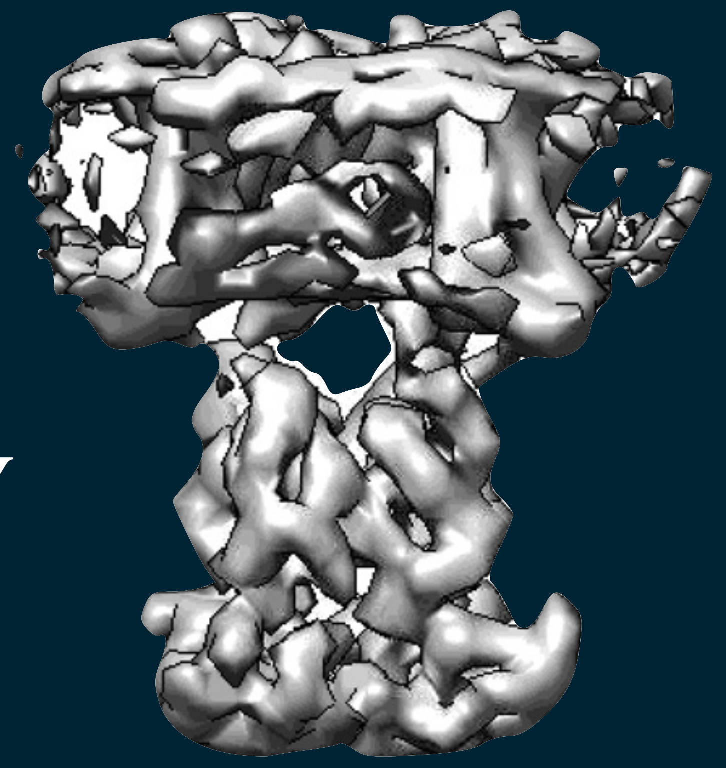
The goal is to discover the volume V



X_i

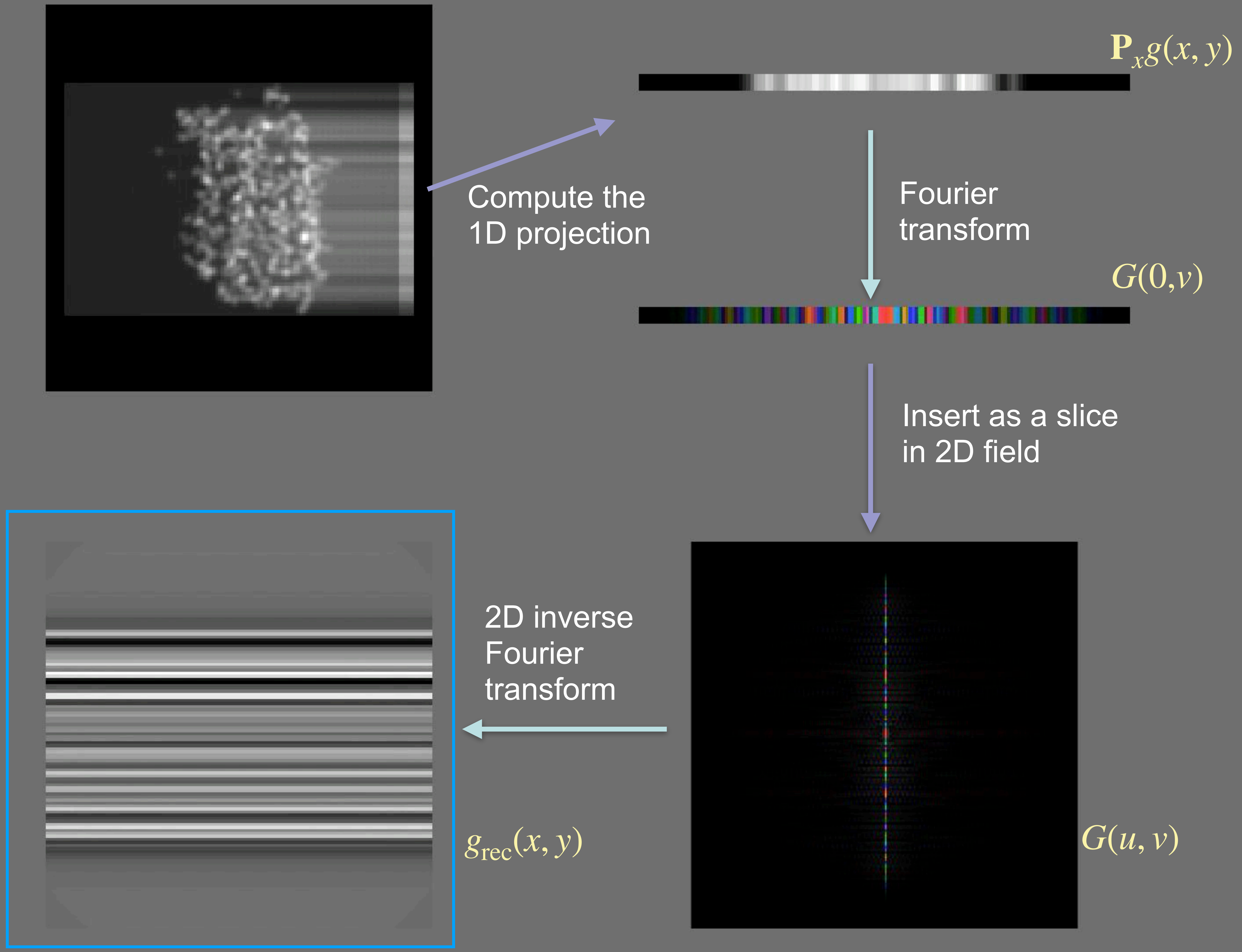
Project along ϕ_i

V



Knowing the orientation angles, we can do 3D reconstruction using the Fourier Slice Theorem

Reconstruction using the Fourier Slice Theorem

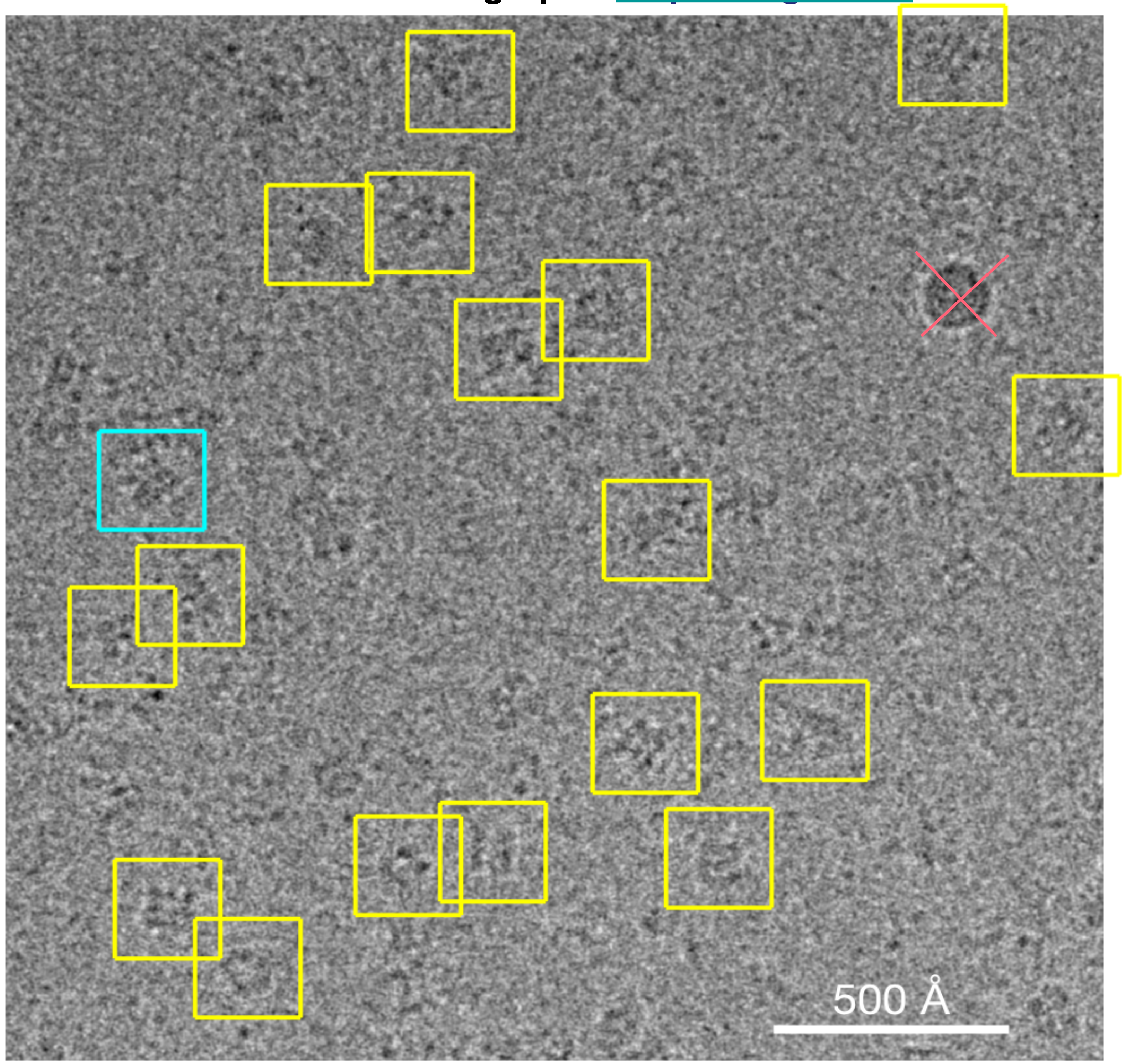


Determining the orientation angles: example from the TRPV1 dataset

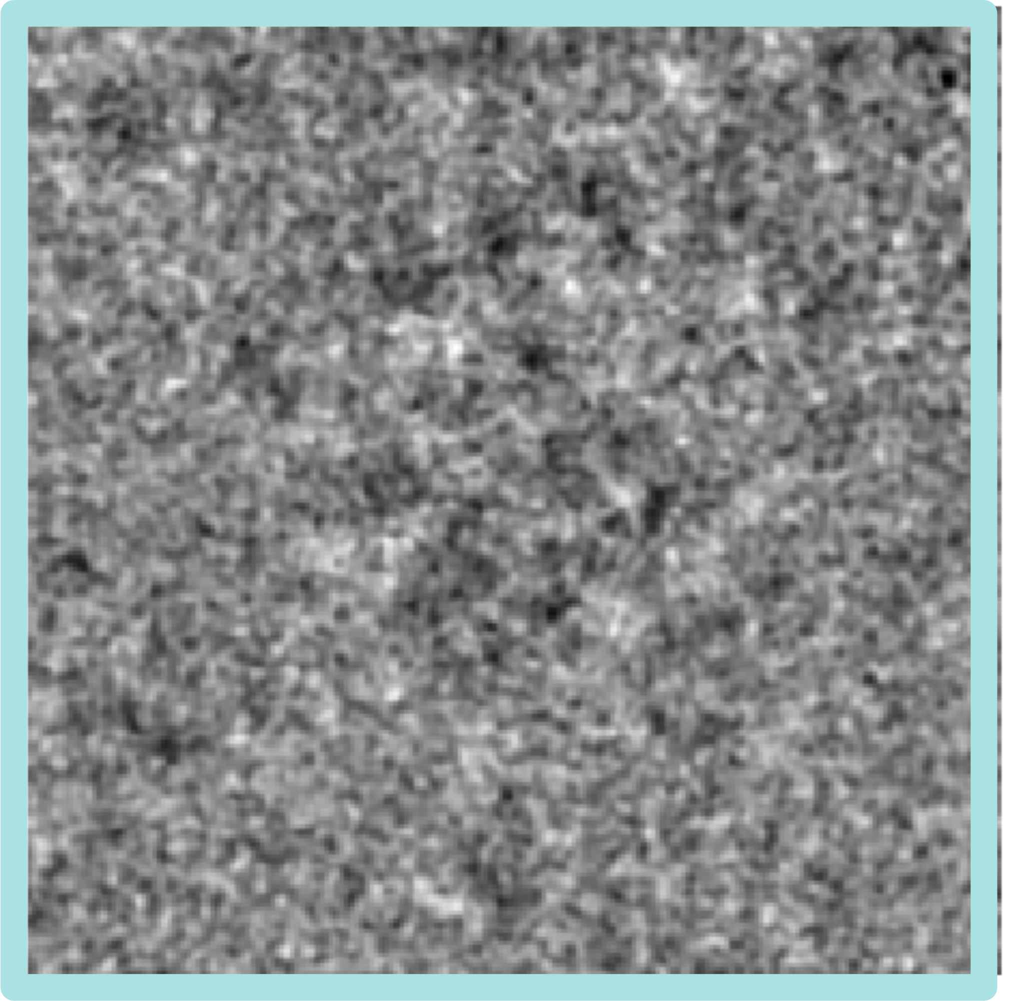
Structure of the TRPV1 ion channel determined by electron cryo-microscopy

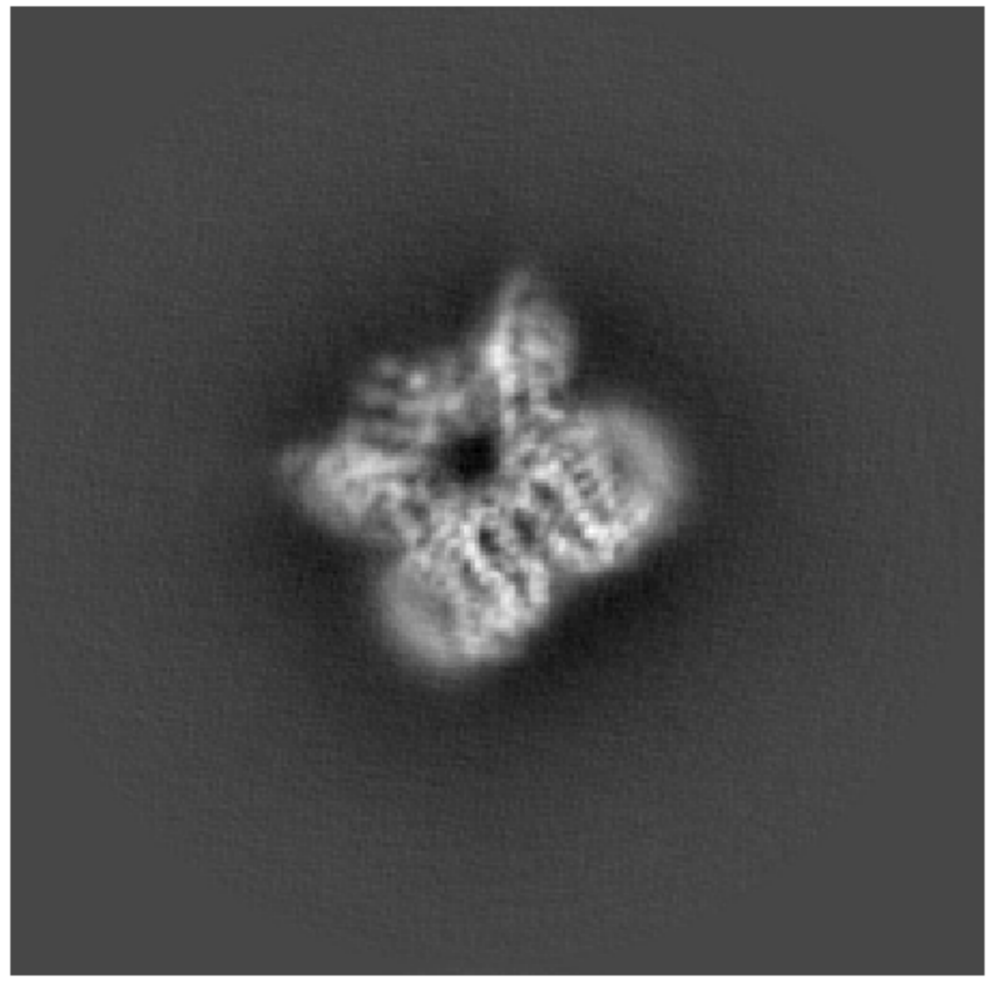
Maofu Liao[1]*, Erhu Cao[2]*, David Julius[2] & Yifan Cheng[1]

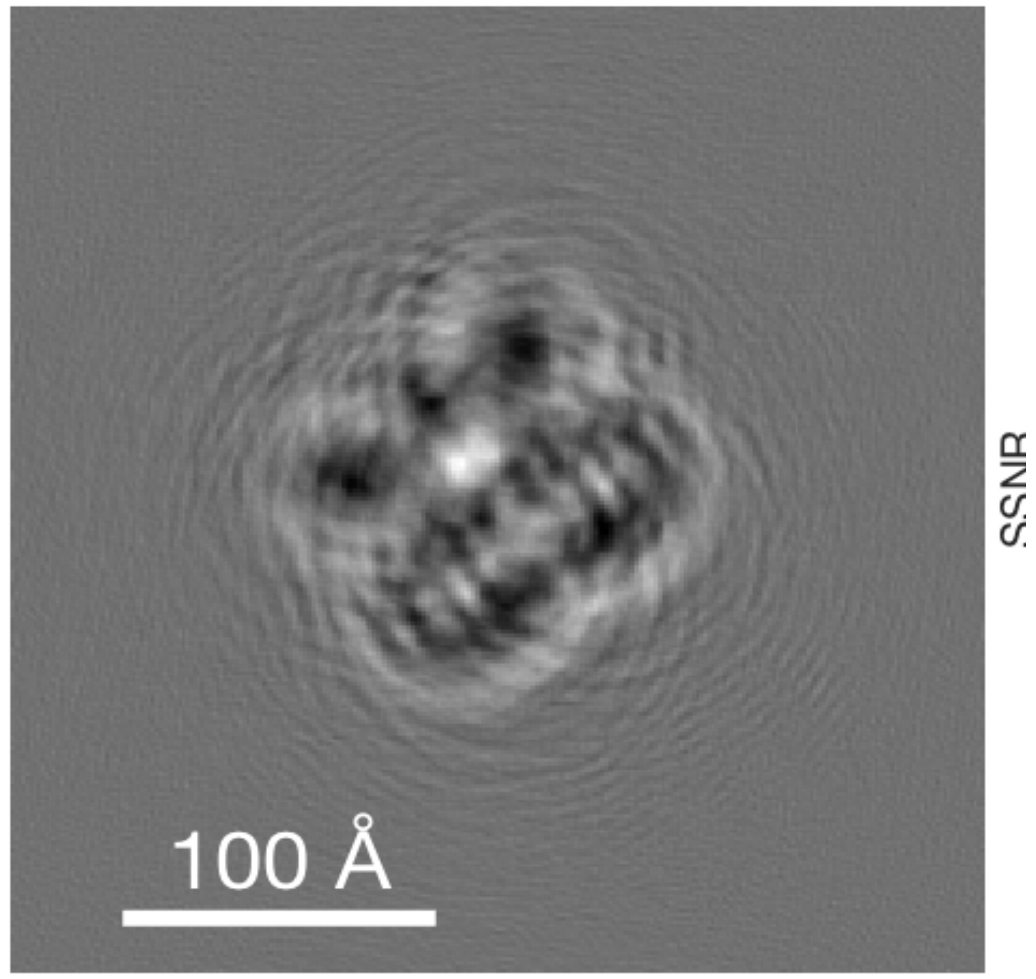
1/4 of a micrograph - empiar.org/10005

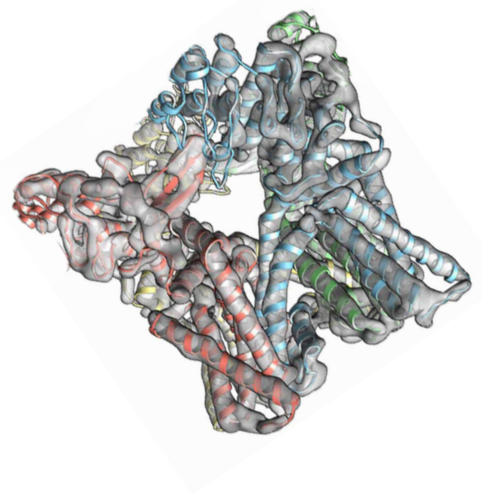


One particle image

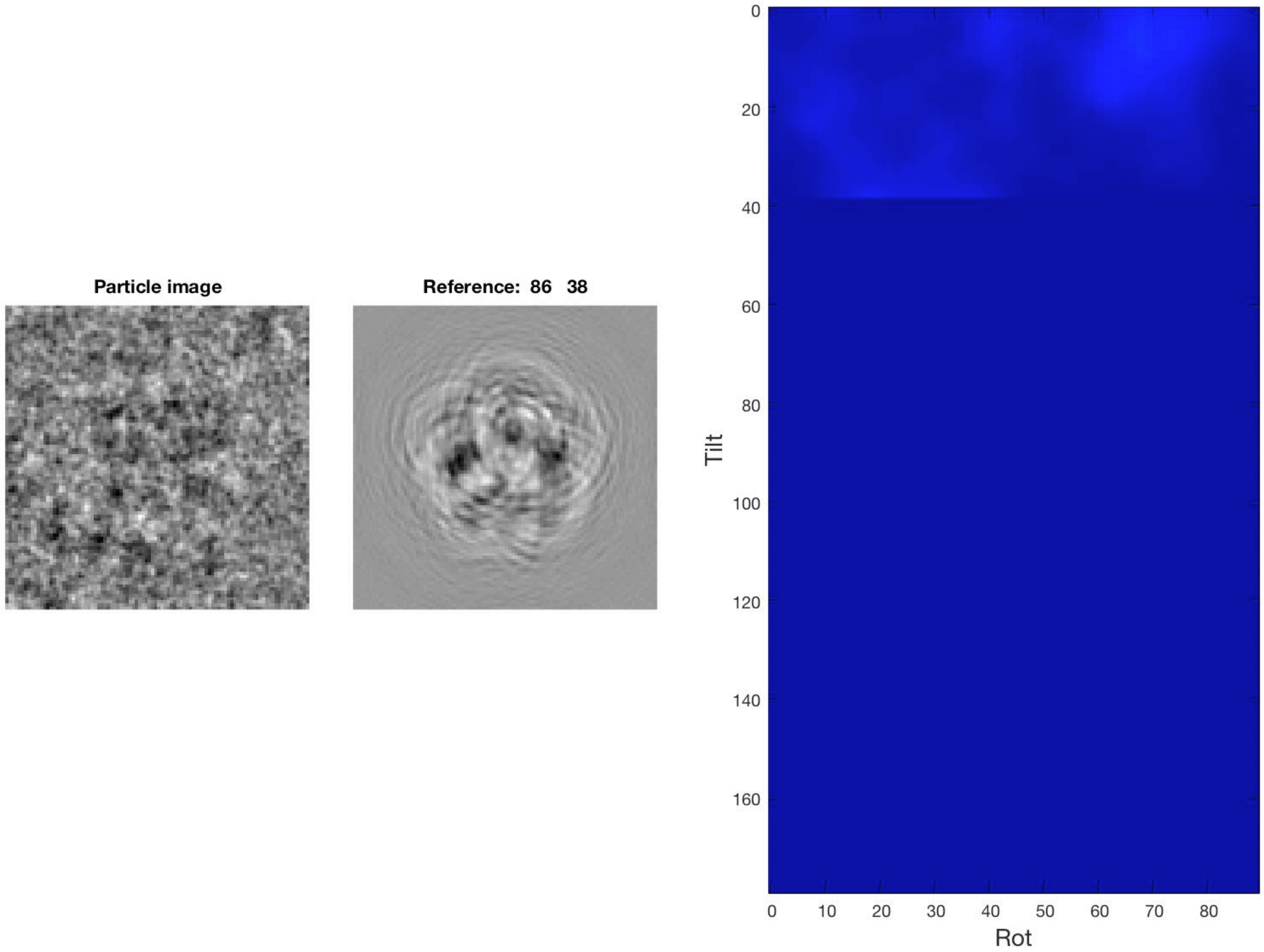




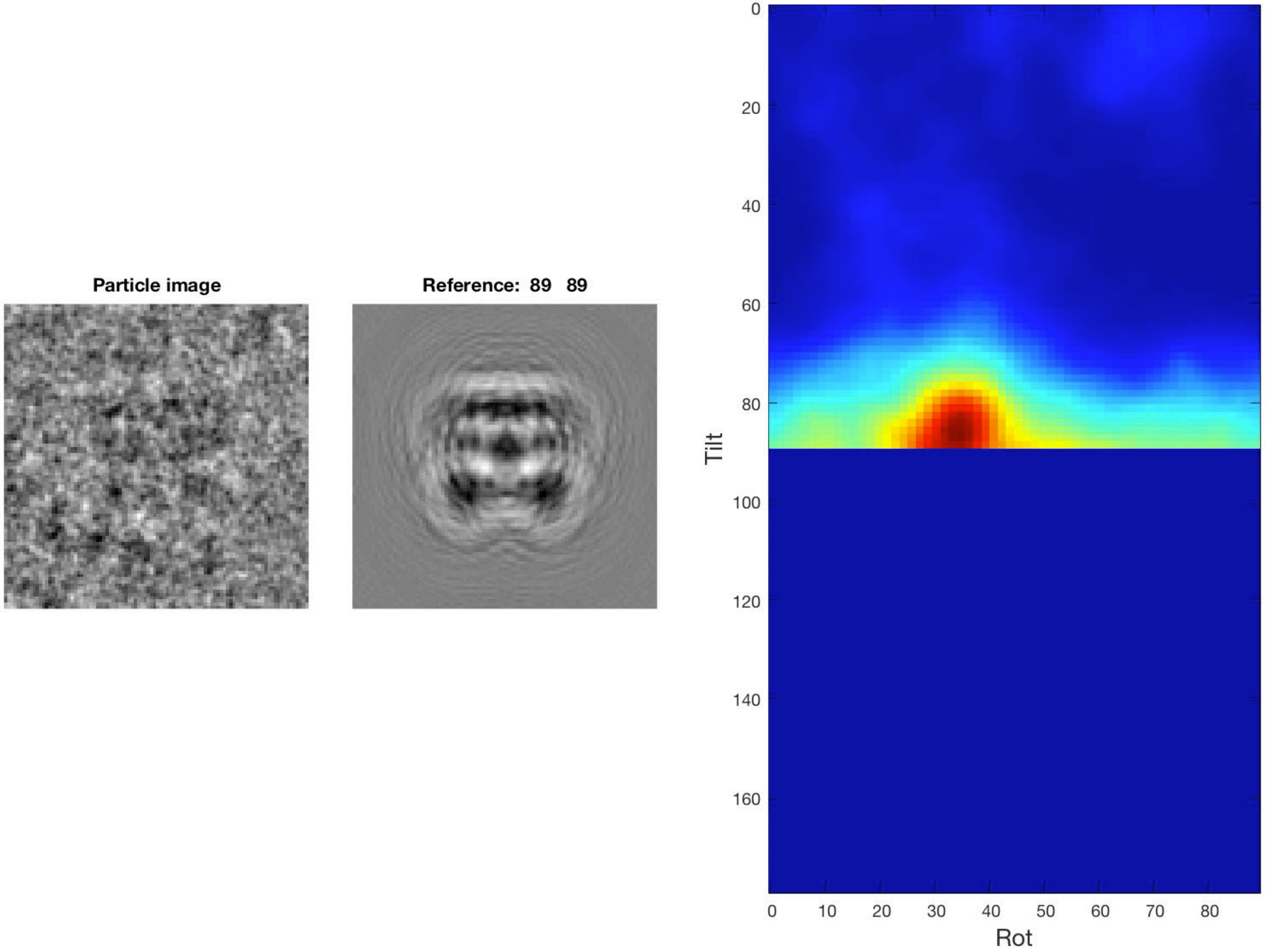




The probability of orientations $P(\phi \mid X, V)$ is remarkably sharp

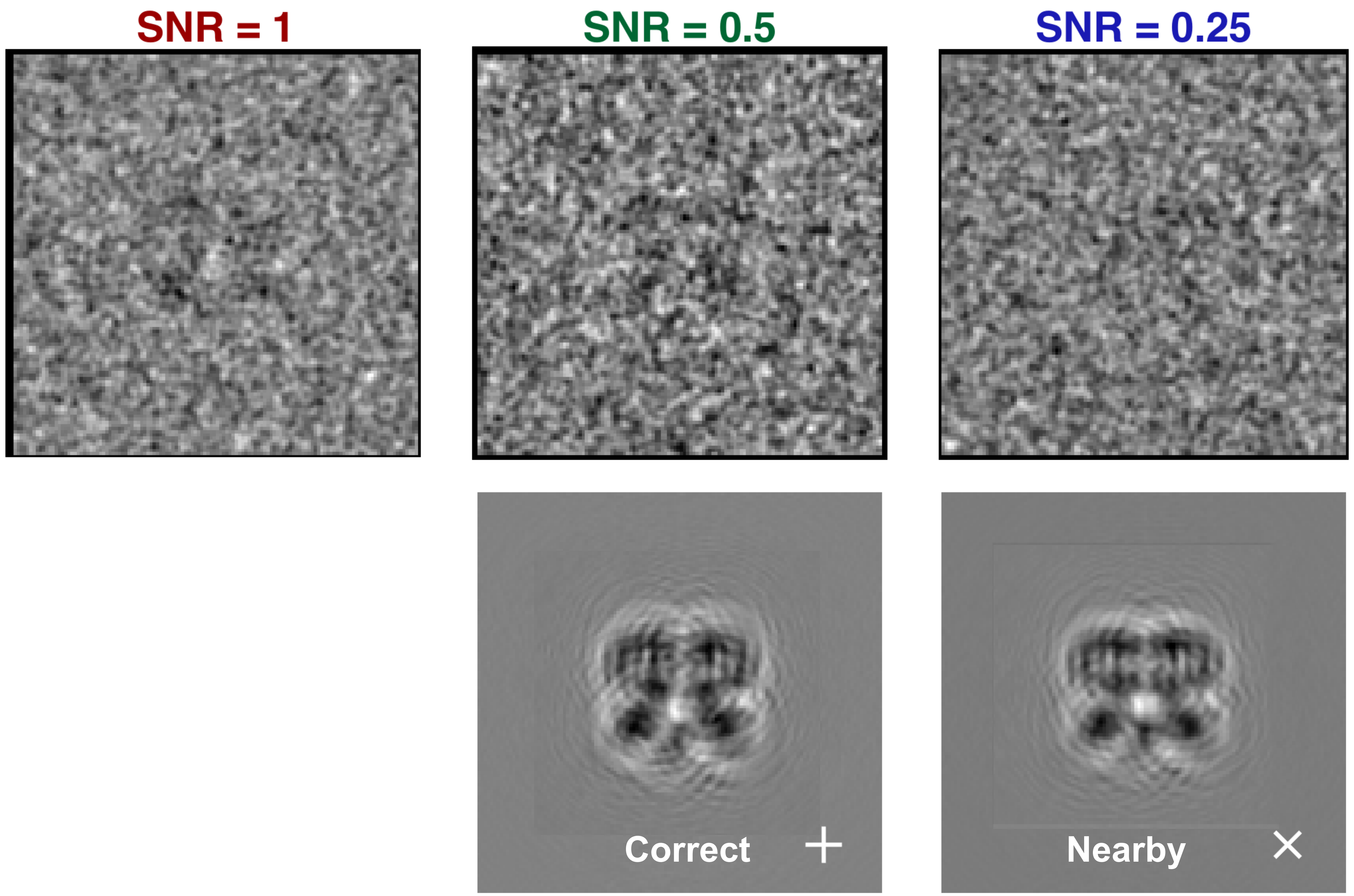


The probability of orientations $P(\phi \,|\, X, V)$ is remarkably sharp



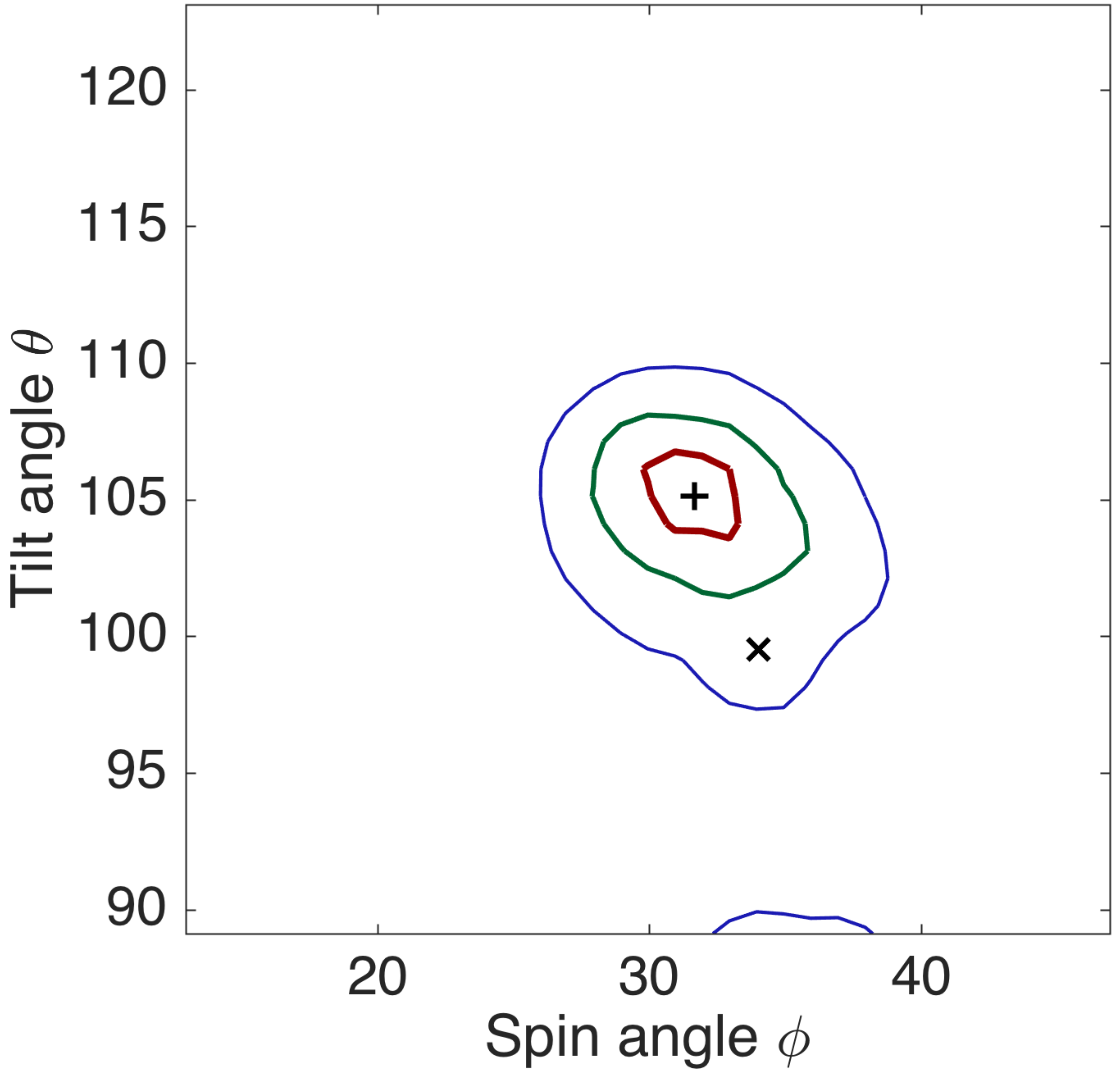
Precise angle determination by image matching

Single-particle images (simulated)



Noiseless projections

Angle accuracy (50th percentile)



3D reconstruction in FREALIGN: correlation and Wiener filtering

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.
4. The sum
$$\sum_i^N \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i X_i$$
is therefore the insertion of *N* slices.

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 Classification in FREALIGN

Suppose our model is that an image X can come from any of K different particle types $V_1, V_2, .. V_K$ and our images are selected randomly from these volumes, projected with noise added.

1. The references are

$$R_{ik} = C_i \mathbf{P}_{\phi_i} V_k \, .$$

We assign k_i such that V_{k_i} yields the projection (with direction ϕ_i) that gives the highest correlation coefficient with X_i.

2. Update the volume according to

$$V_k^{(n+1)} = \frac{\sum_{k_i=k} \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i X_i}{k_w + \sum_{k_i=k} \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i^2}$$

Single-particle reconstruction and classification

1. Undoing the CTF
2. Projection matching: Frealign
3. Maximum Likelihood: Relion and cryoSparc

Probabilities, another way to compare images

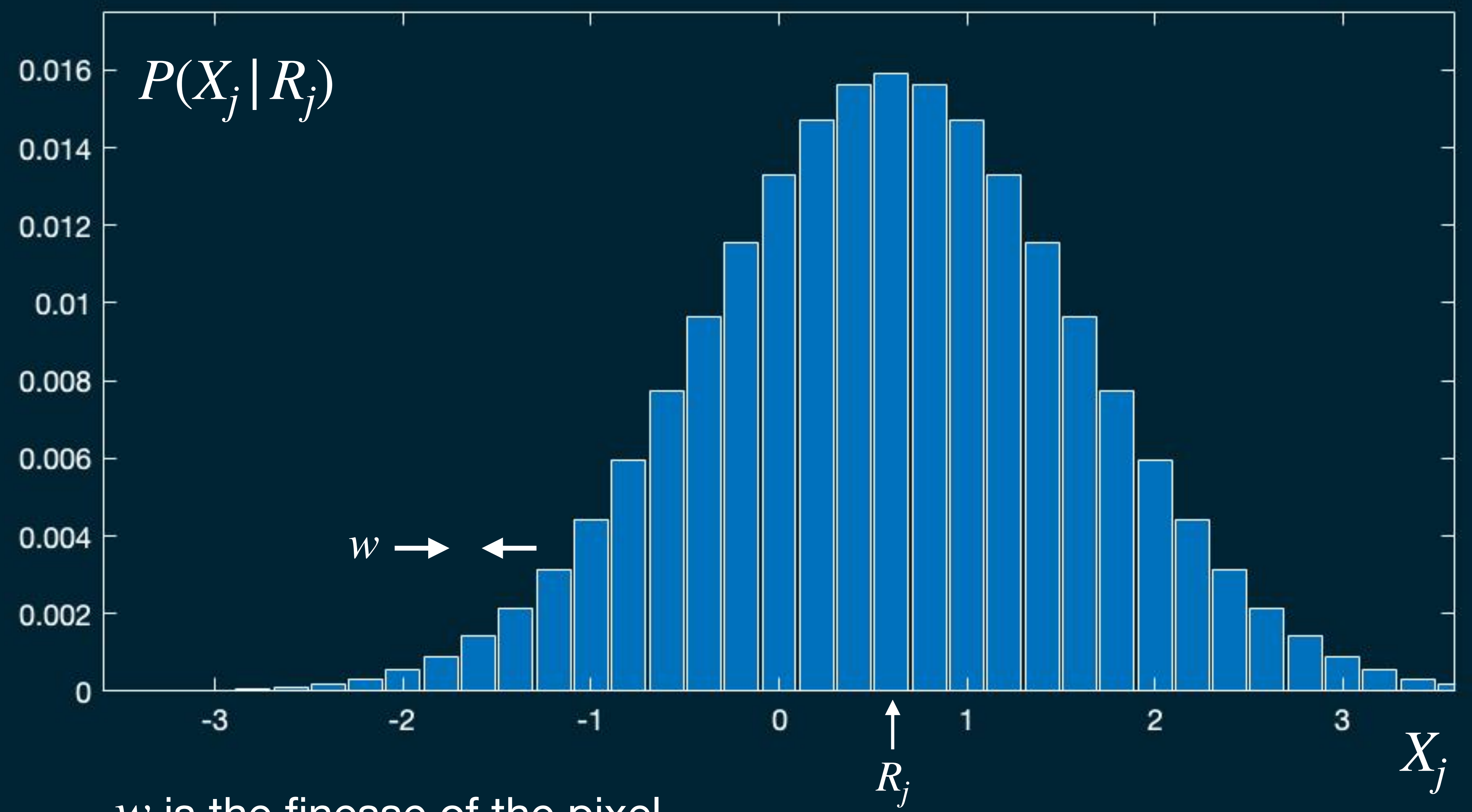
Image model: $X = R + N$

Probability of the j^{th} pixel value:

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

Probability of observing an entire image that comes from R:

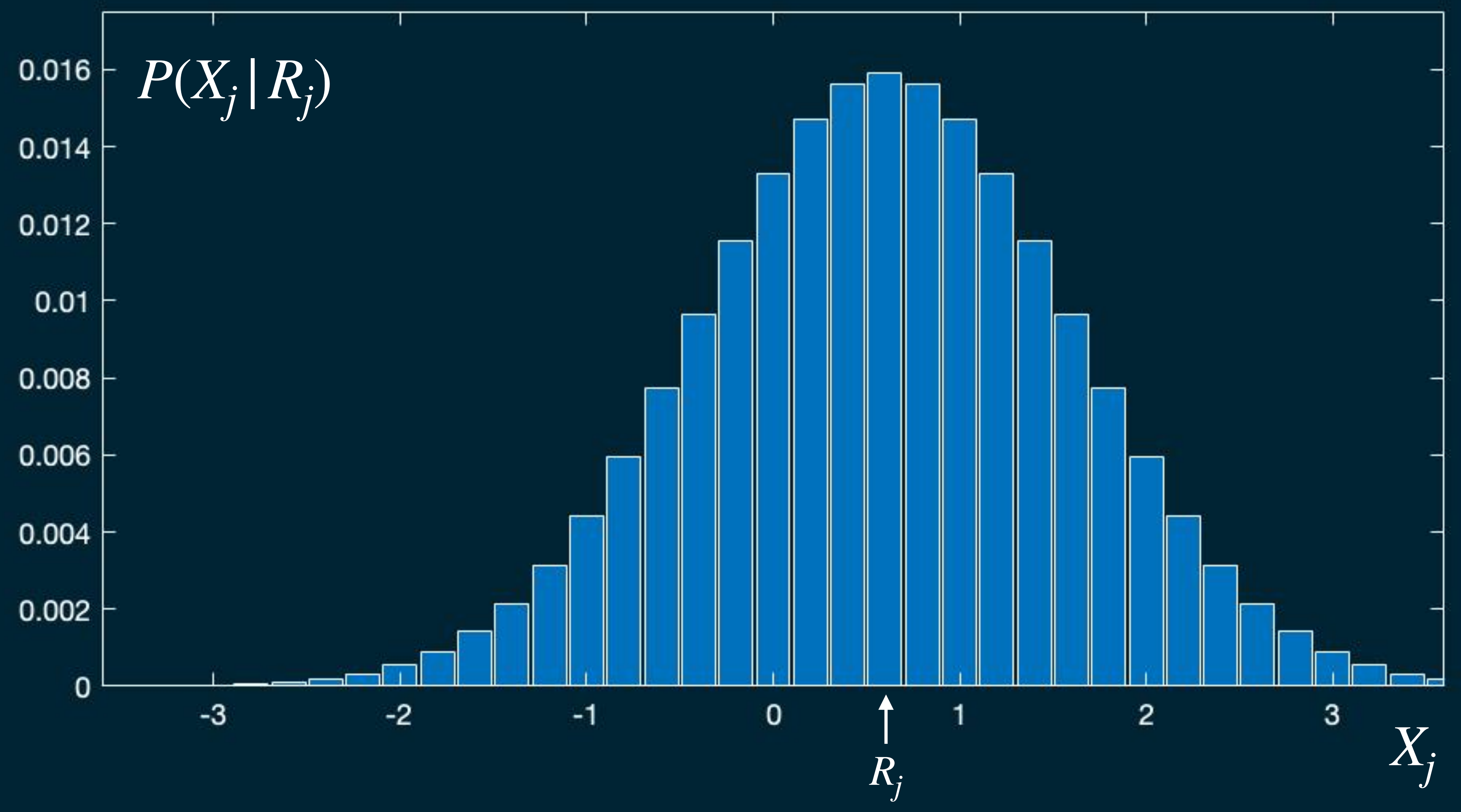
$$P(X | R) = \frac{\cancel{w^J}\ 1}{(2\pi\sigma^2)^{J/2}} e^{-||X-R||^2 / 2\sigma^2}$$



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

Simplified image probability

$$X = R + N$$



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 for simplicity.)

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 V consistent with these data, say maximizing the posterior probability

$$P(V \,|\, \mathbf{X}).$$

According to Bayes' theorem,

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

prior → **Experiment** → ***posterior***

likelihood

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

$$\boxed{\mathrm{Lik}(V) = P(\mathbf{X} \,|\, V)}$$

We know how to compute the likelihood

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

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

A common assumption is that $P(\phi)$ is uniform.

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,

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

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 does not)
Maximum Likelihood converges to the right answer.

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 if we know ϕ:

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

Probability of a projection direction for X_i:

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

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:

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

…Relion's compute-intensive "Expectation" step is basically the evaluation of $\Gamma_i(\phi)$ for each image X_i, based on the current estimate $V^{(n)}$ of the volume.

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

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

3D Classification

We can use Expectation-Maximization to optimize K different volumes $V_1, V_2, .. V_K$ simultaneously. The formula is essential the same except that the function Γ depends also on k, so

$\Gamma = \Gamma_i^{(n)}(\phi, k)$

The iteration, guaranteed to increase the likelihood, is

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

For comparison, here is Frealign's iteration:

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

$$V_k^{(n+1)} = \frac{\sum_{k_i=k} \mathbf{P}_{\phi_i}^\mathbf{T} C_i X_i}{k_w + \sum_{k_i=k} \mathbf{P}_{\phi_i}^\mathbf{T} C_i^2}$$

…Relion's compute-intensive "Expectation" step is basically the evaluation of $\Gamma_{i,k}(\phi)$ for each image X_i and volume V_k

In Relion, 2D and 3D classification and refinement use the same program

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

Quantity	Meaning in 3D classification	Meaning in 2D classification
V_k	Class volume	Class average image
ϕ	3 Euler angles of orientation + 2 translations	1 angle of rotation + 2 translations
$\mathbf{P}_\phi$	Projection operator 3D $\longrightarrow$ 2D	Image rotation and shift
$\mathbf{P}_\phi^{\mathbf{T}}$	Back-projection operator 2D $\longrightarrow$ 3D	Reverse shift and rotation

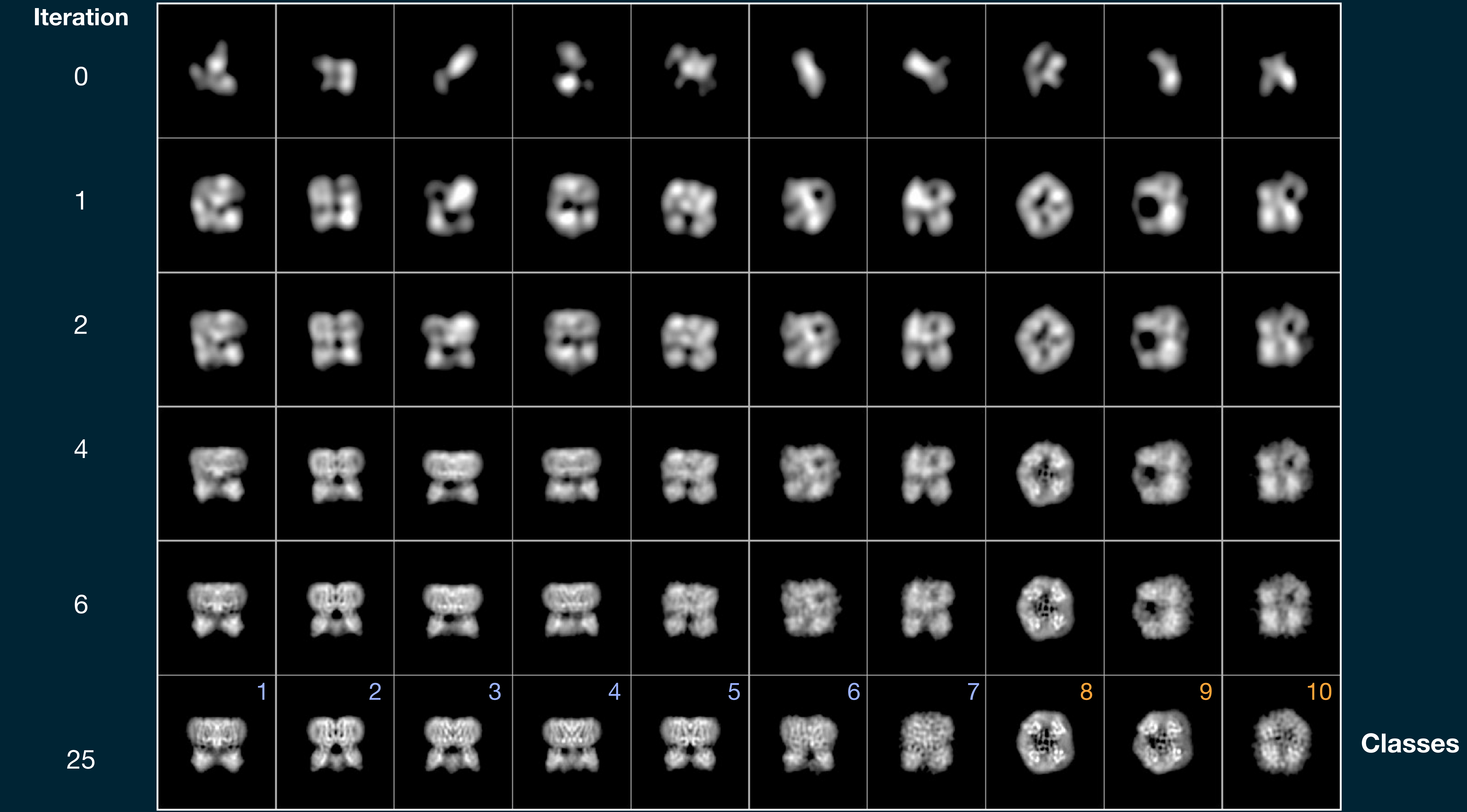
2D classification by maximum likelihood

Given estimates of $A_1 \dots A_K$ from the n^{th} iteration, let $\Gamma_i(\phi, k)$ be the probability of the transformation ϕ and index k for the i^{th} image and k^{th} class. Then the new class estimate in the $(n+1)^{\text{st}}$ iteration will be

$$A_k^{(n+1)} = \frac{\sum_i^N \int \Gamma_i(\phi, k) \mathbf{P}_\phi^{\mathbf{T}} C_i X_i \, d\phi}{\frac{1}{\text{SSNR}} + \sum_i^N \int \Gamma_i(\phi, k) \mathbf{P}_{\phi_i}^{\mathbf{T}} C_i^2 \, d\phi}$$

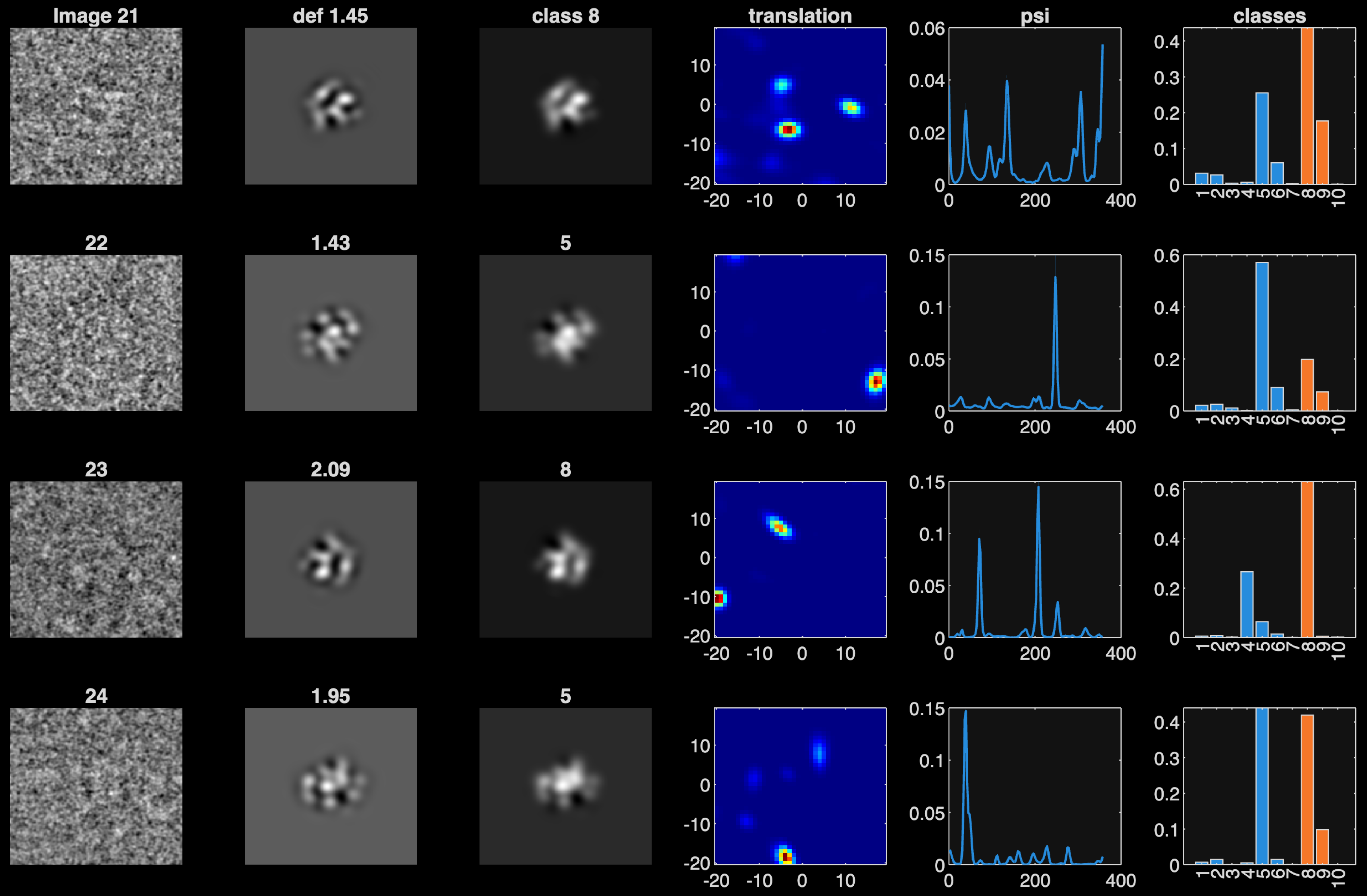
Again, $\mathbf{P}_{\phi_i}^{\mathbf{T}}$ is the inverse of the transformation $\mathbf{P}_{\phi_i}$.

2D classification by maximum likelihood



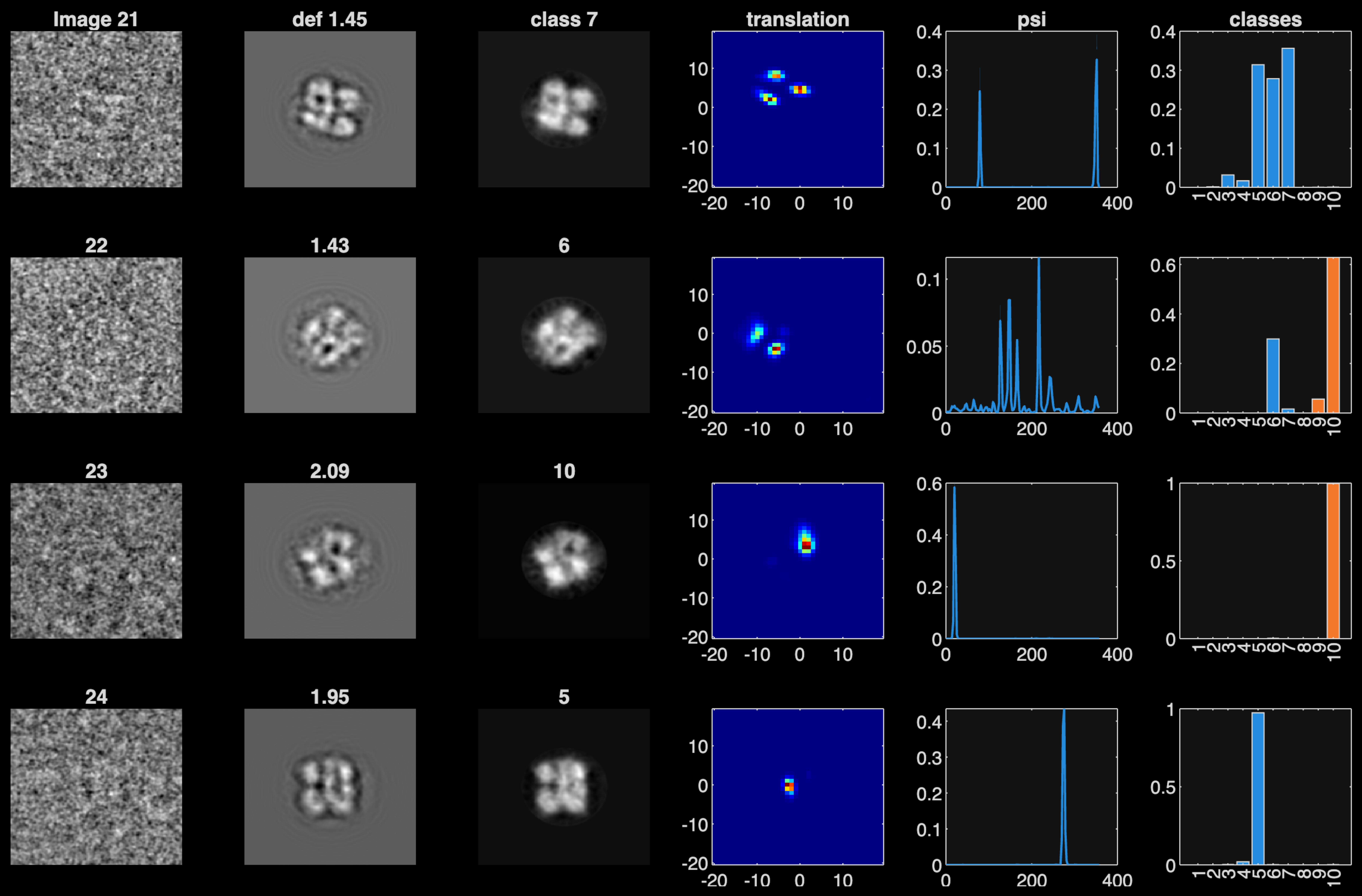
2D classification by maximum likelihood

Iteration 0



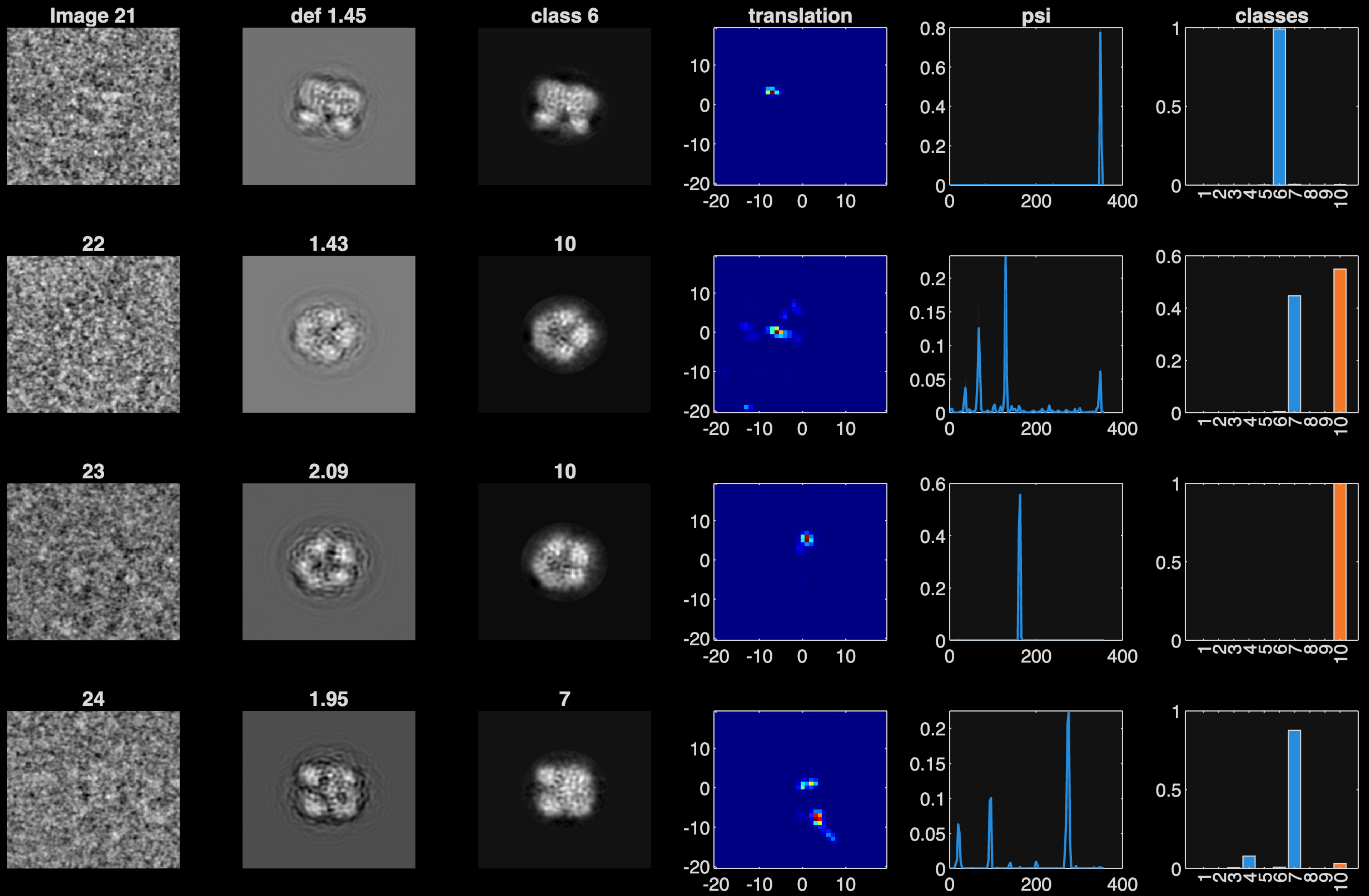
2D classification by maximum likelihood

Iteration 4

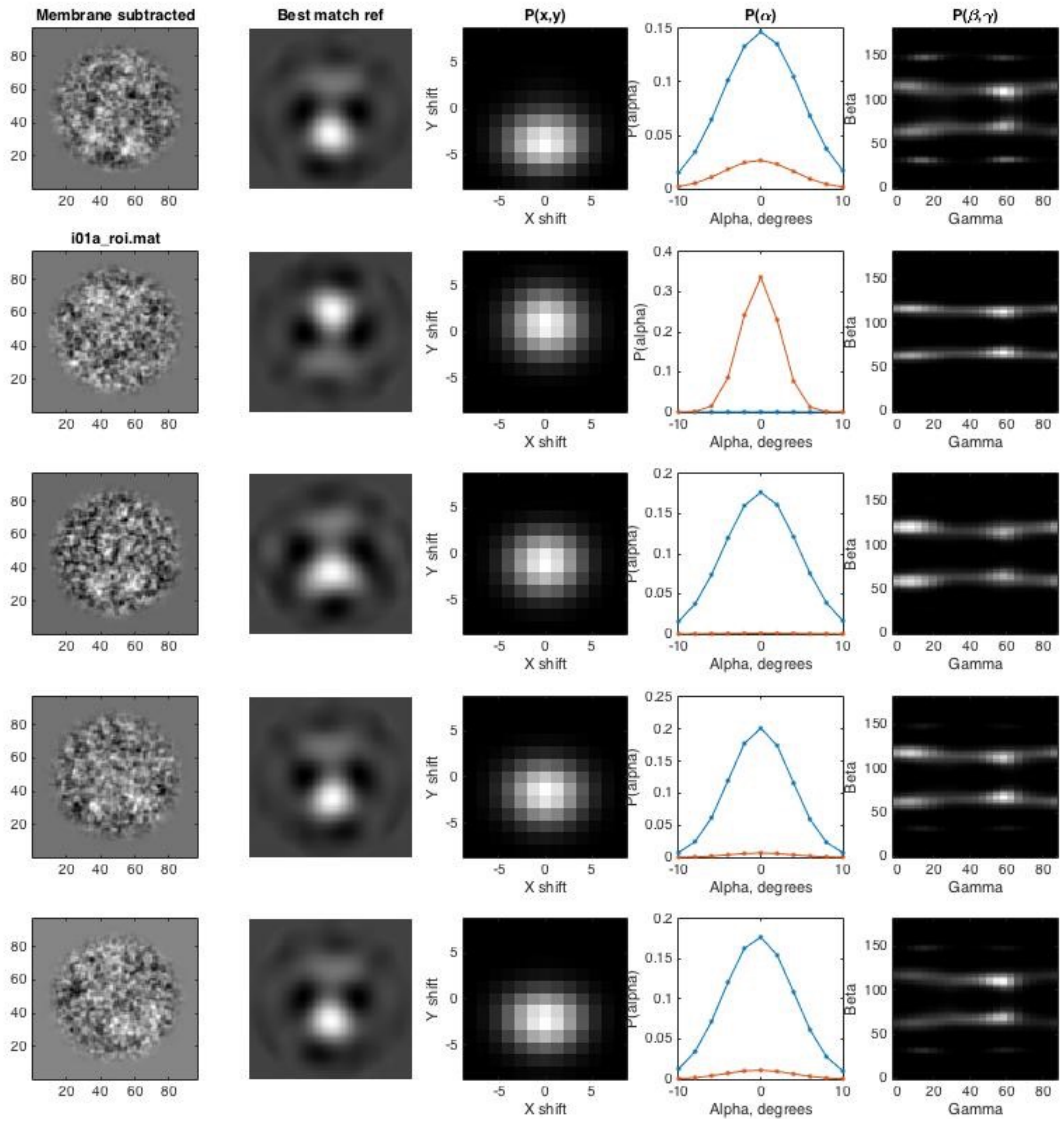


2D classification by maximum likelihood

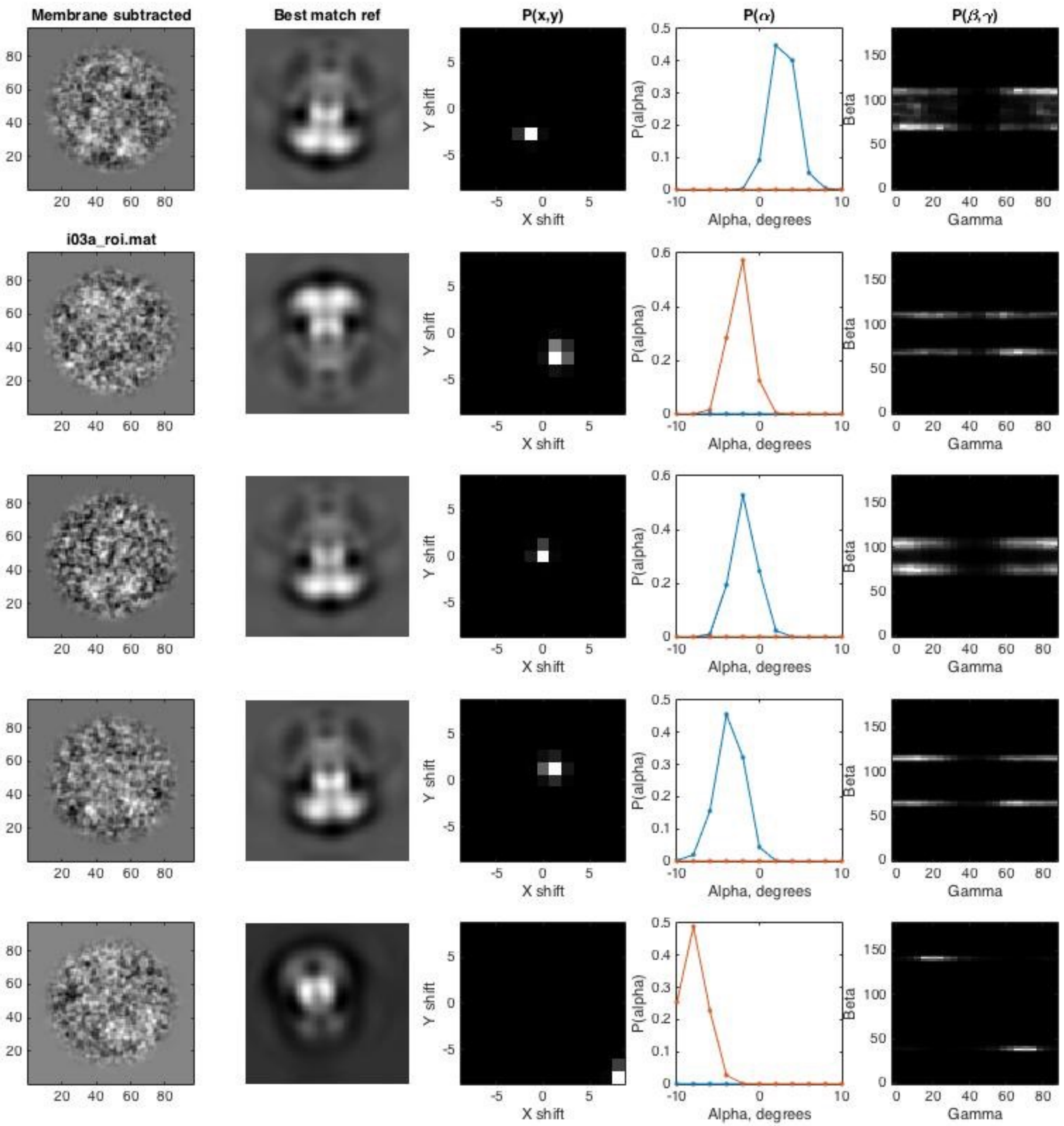
Iteration 25



3D Reconstruction: on the first EM iteration, angle assignments are not sharp

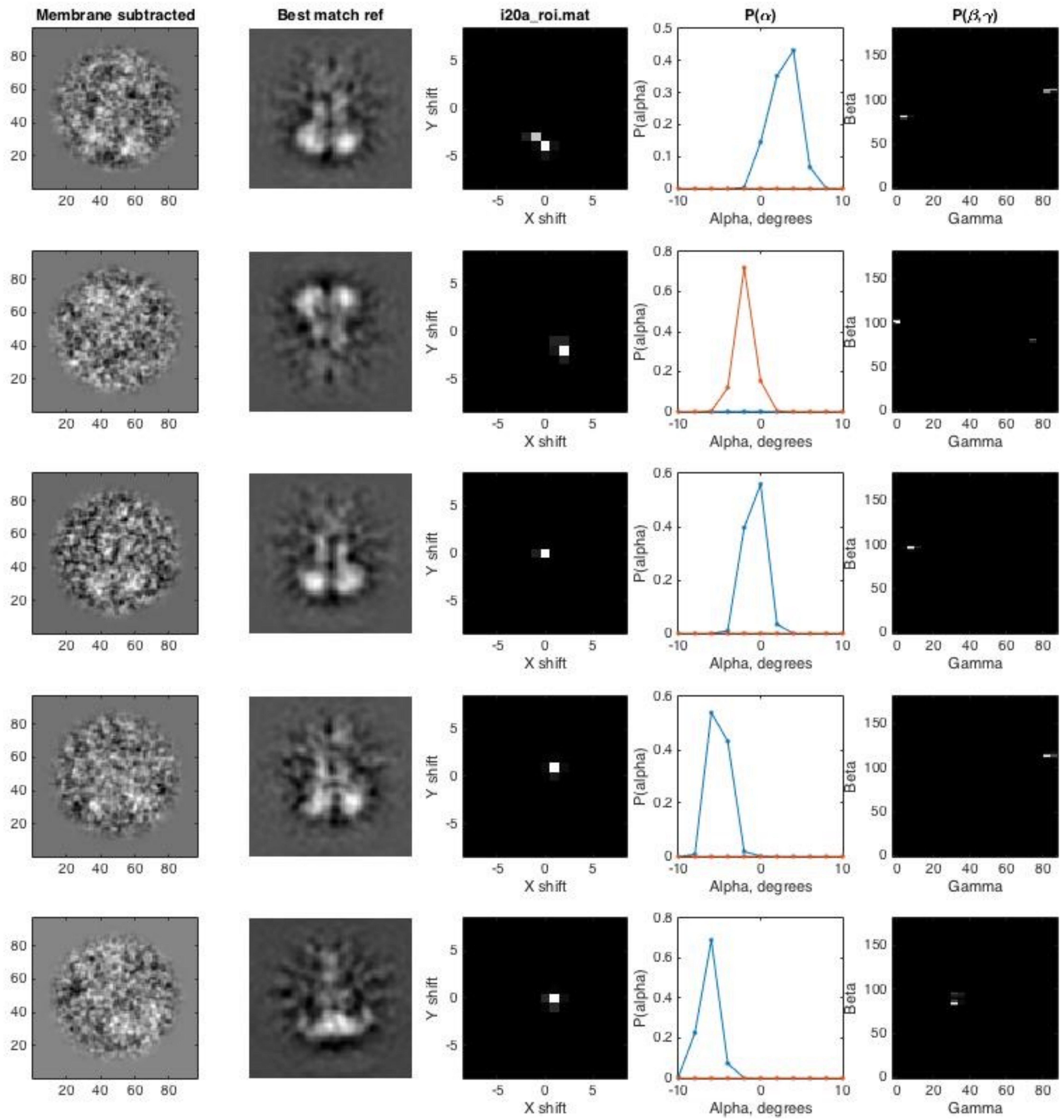


Iteration 3



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 14, near convergence: distributions are becoming sharp



For 3D reconstruction, orientation determination is the most expensive step

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

For 3D reconstruction, orientation determination is the most expensive step

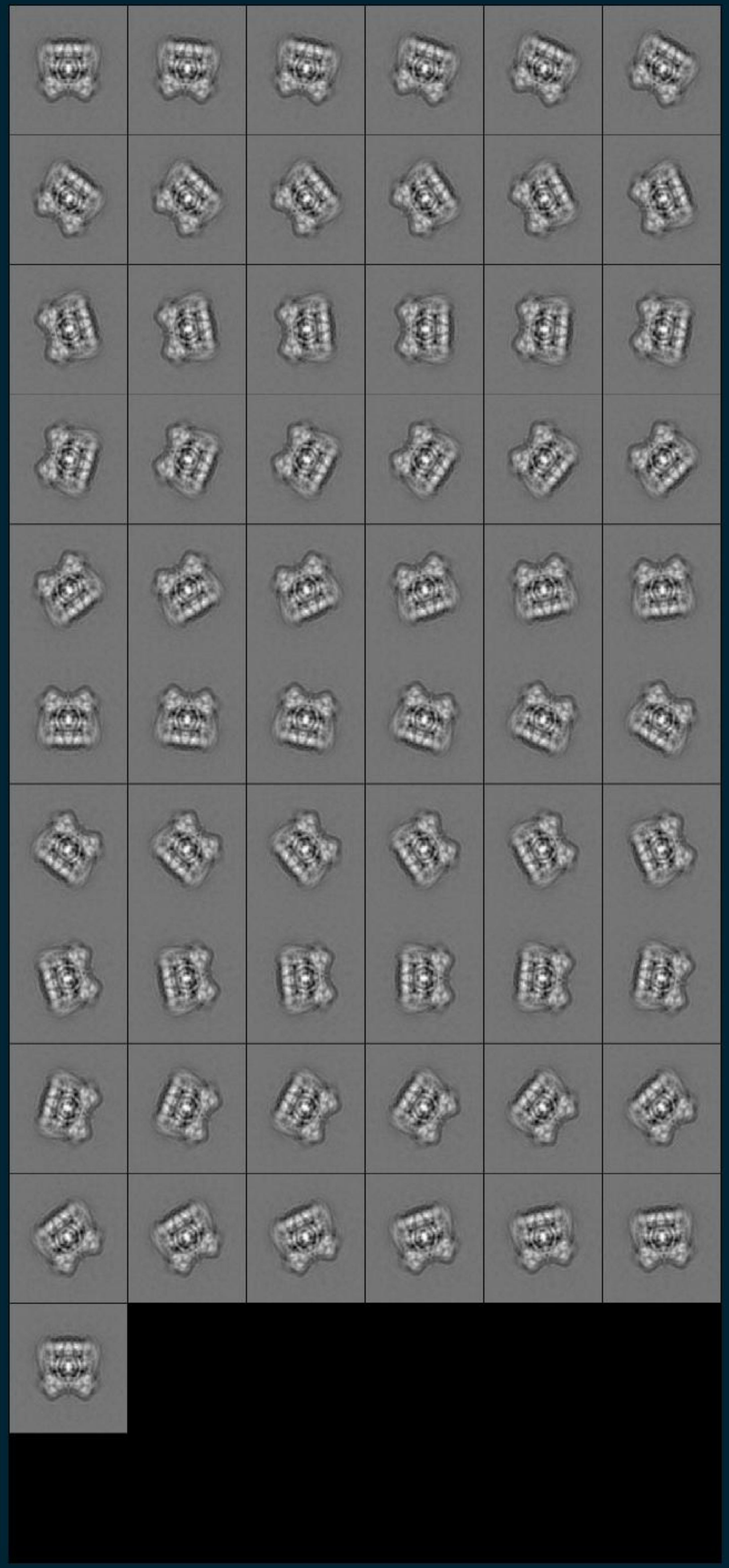
$$\text{No. operations} \approx \underbrace{\frac{\pi^3}{8} t^2 n^5 N}_{\text{finding orientations}} + \underbrace{\pi n^4 + N n^2}_{\text{3D reconstruction}}$$

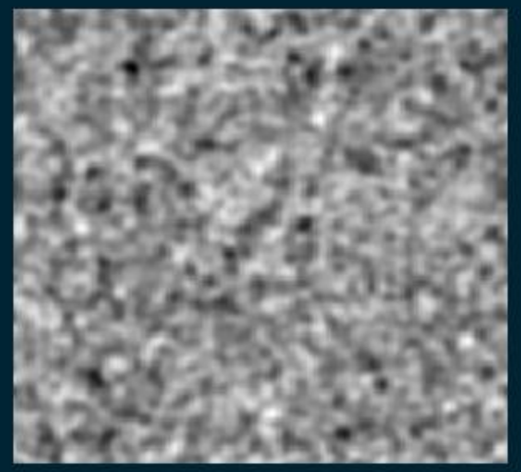
e.g. $N=10^5$, $n=128$, $t=7$

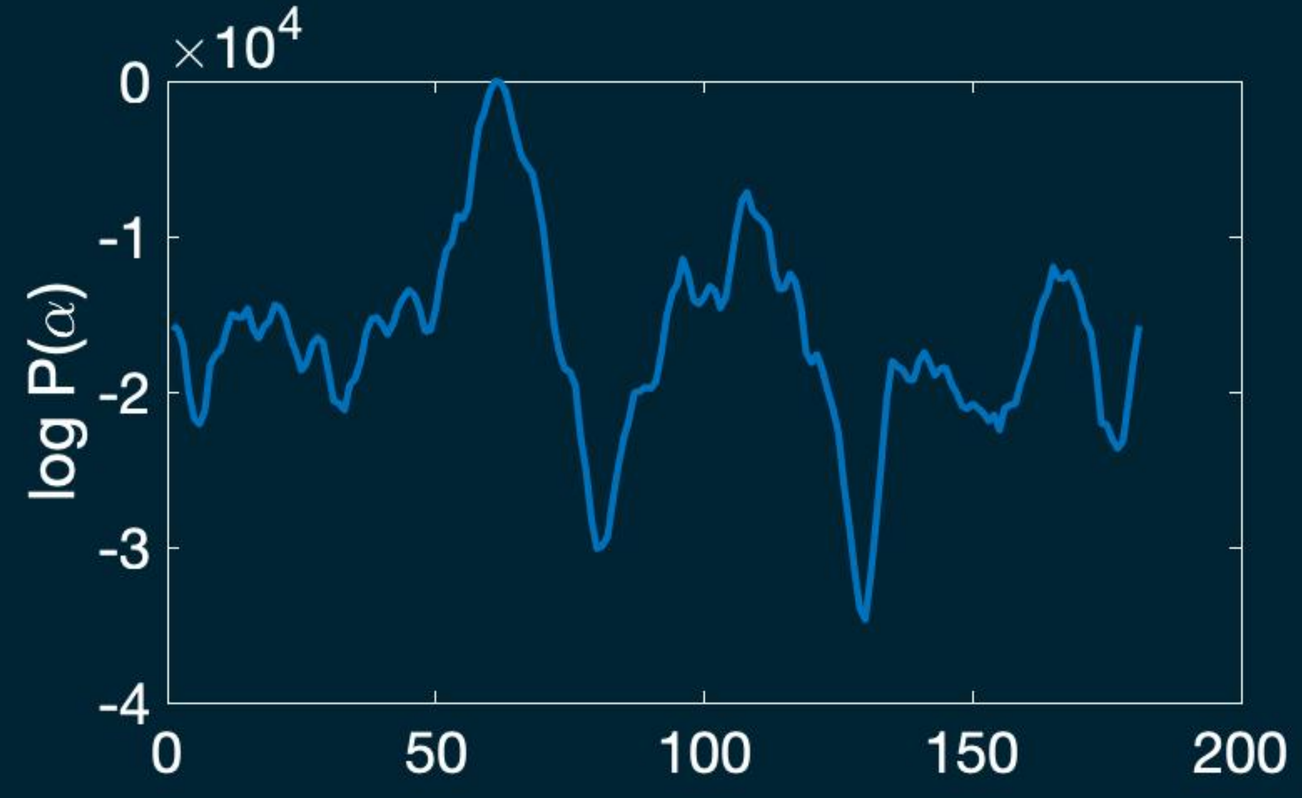
No. operations ≈ 6 x 10^{17} ≈ 19 CPU-years

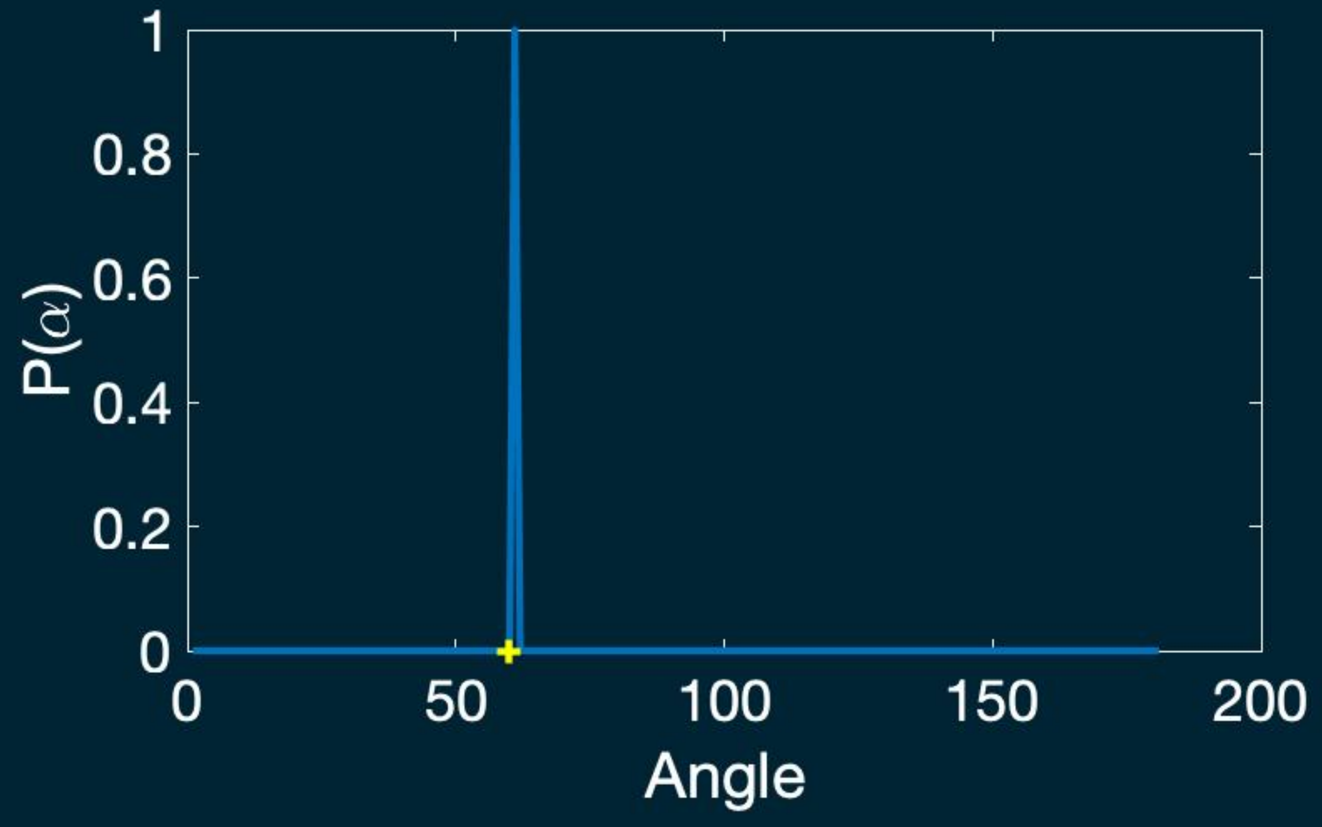
With efficient programs, ~ 1 CPU-day

Evaluating Γ_ϕ is expensive: here we estimate only one of five parameters

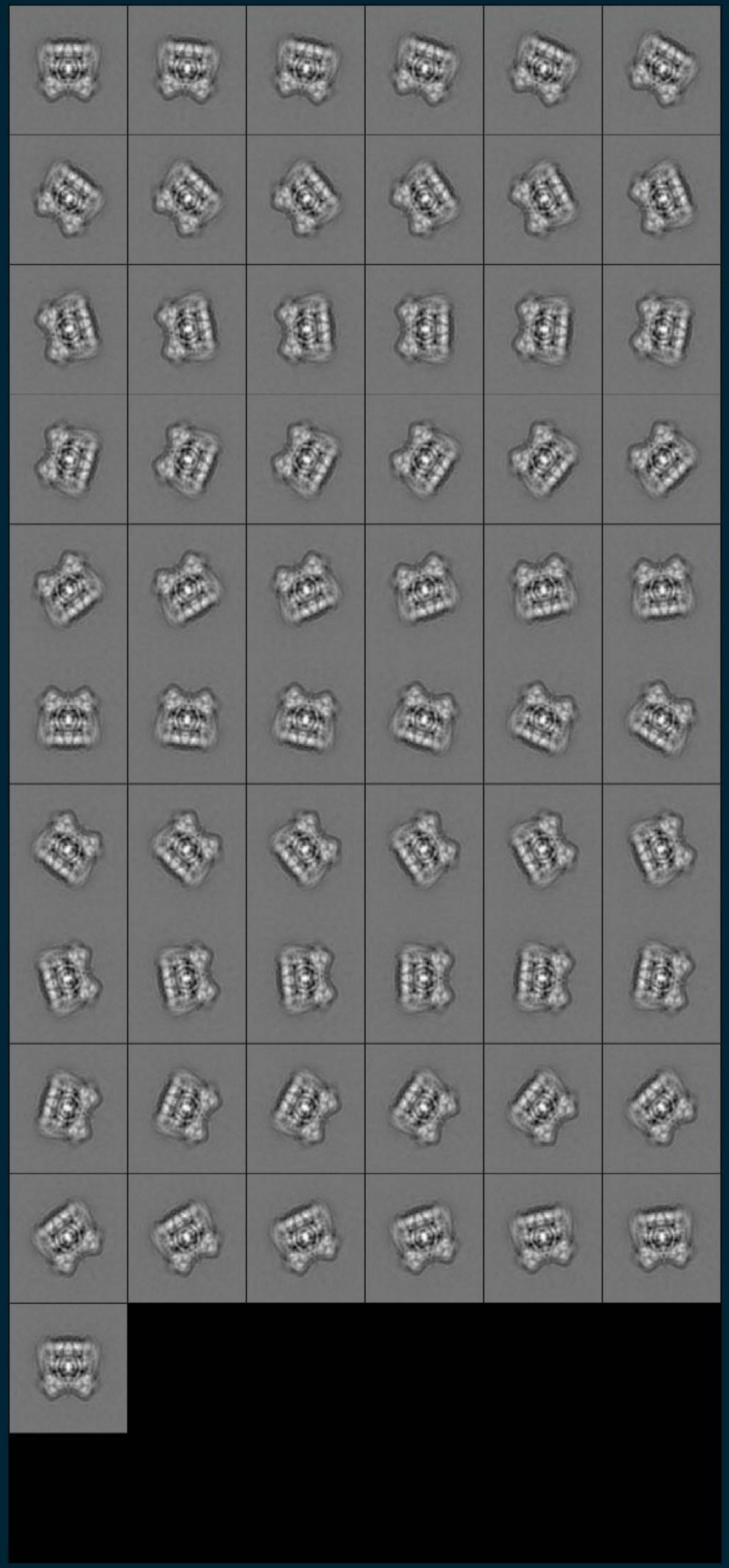


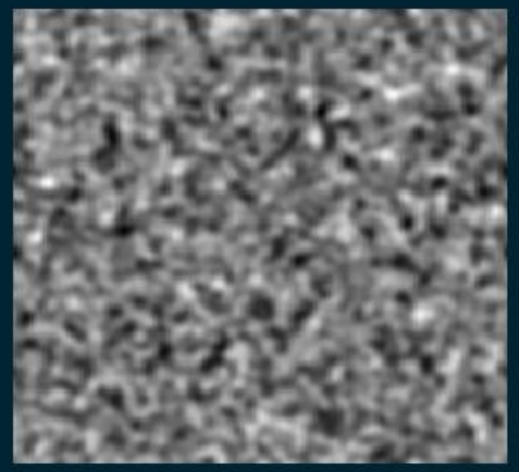


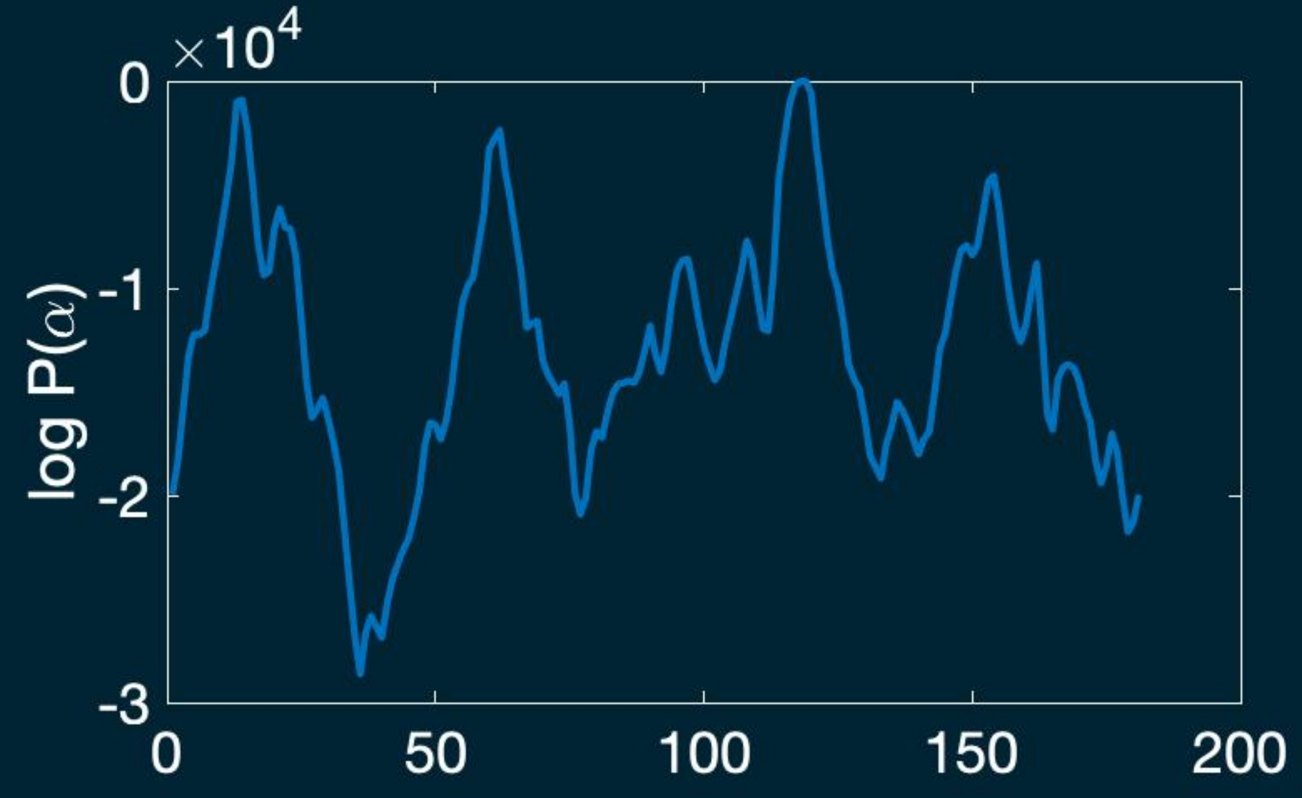


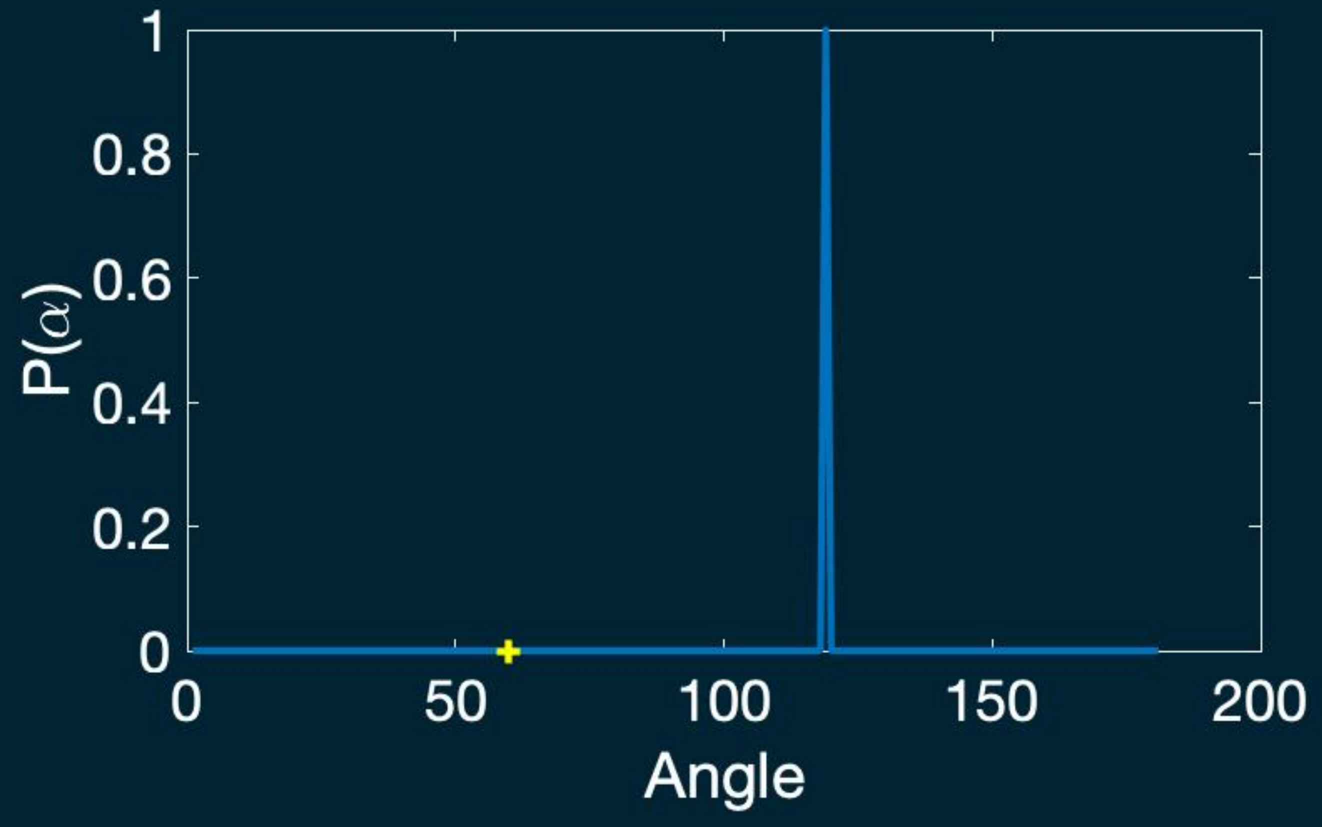


Evaluating Γ_ϕ is expensive: one of 5 parameters

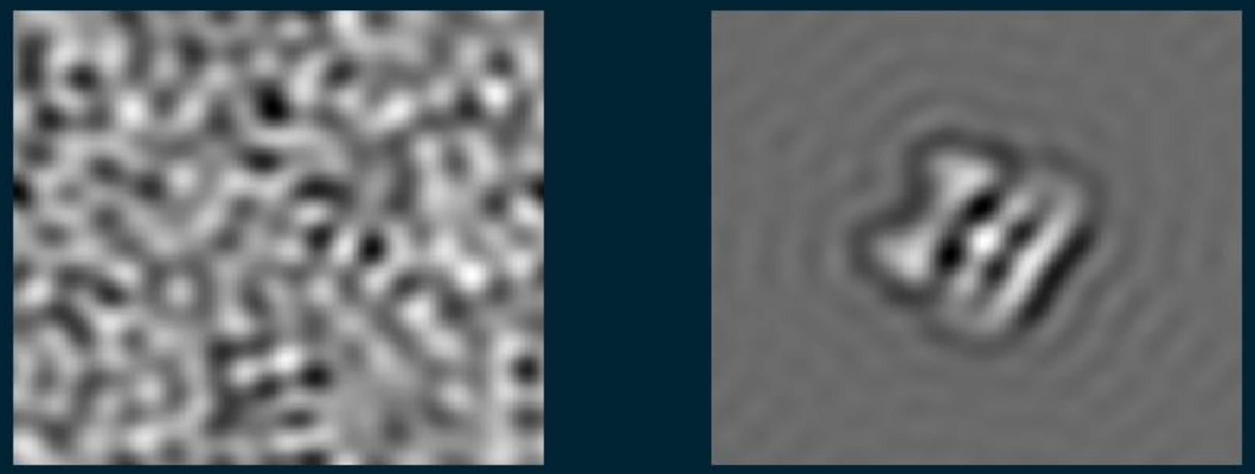




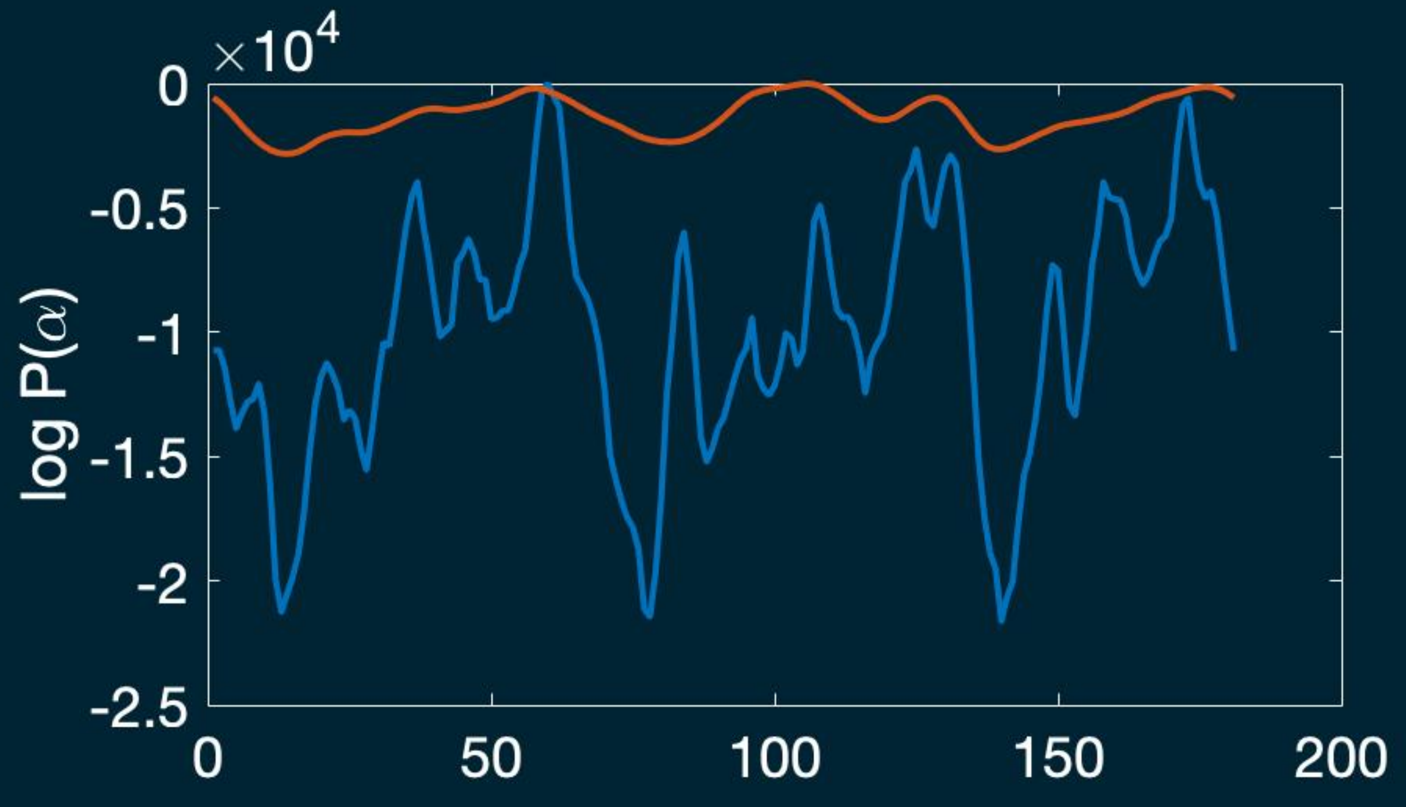


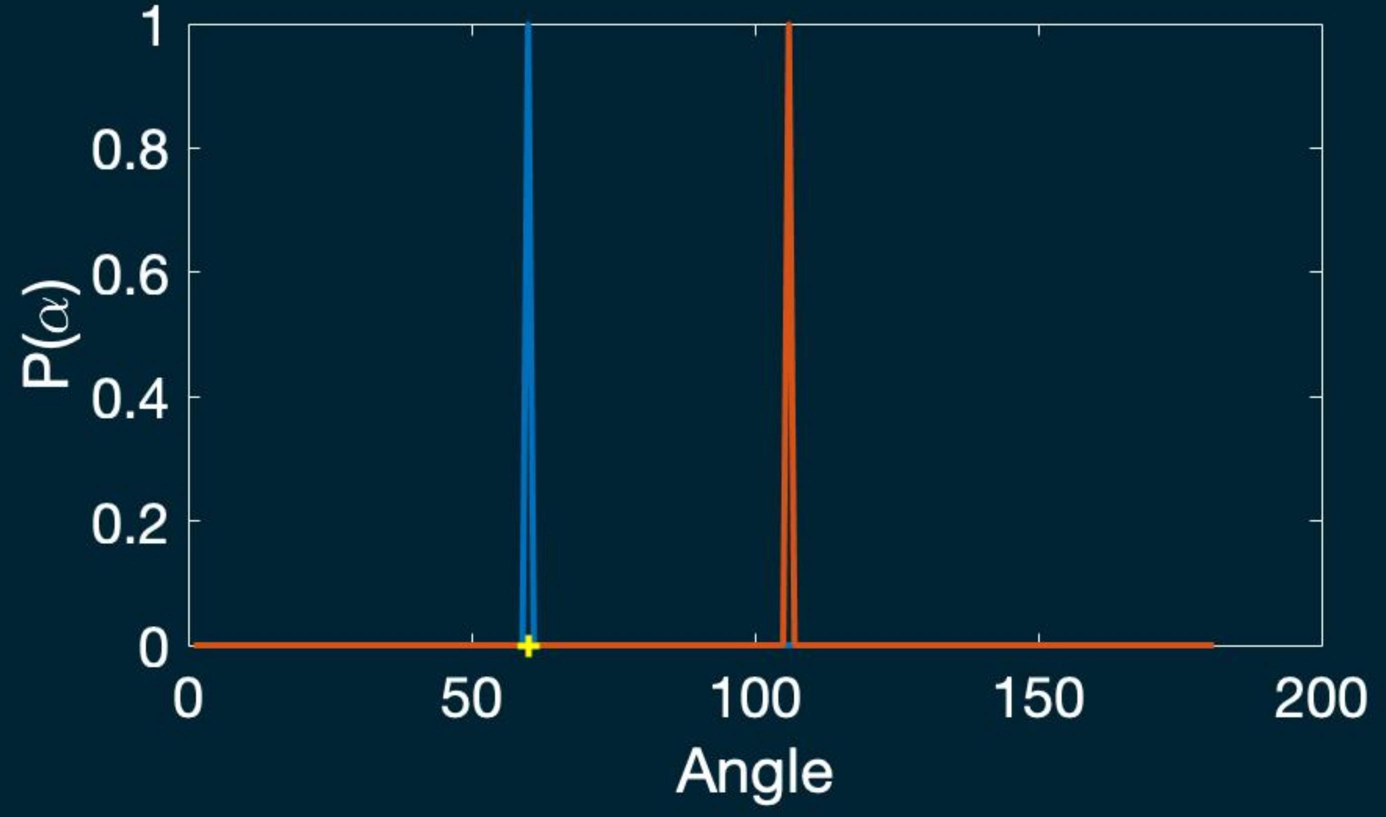


Domain reduction: branch and bound, illustrated for 1D



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

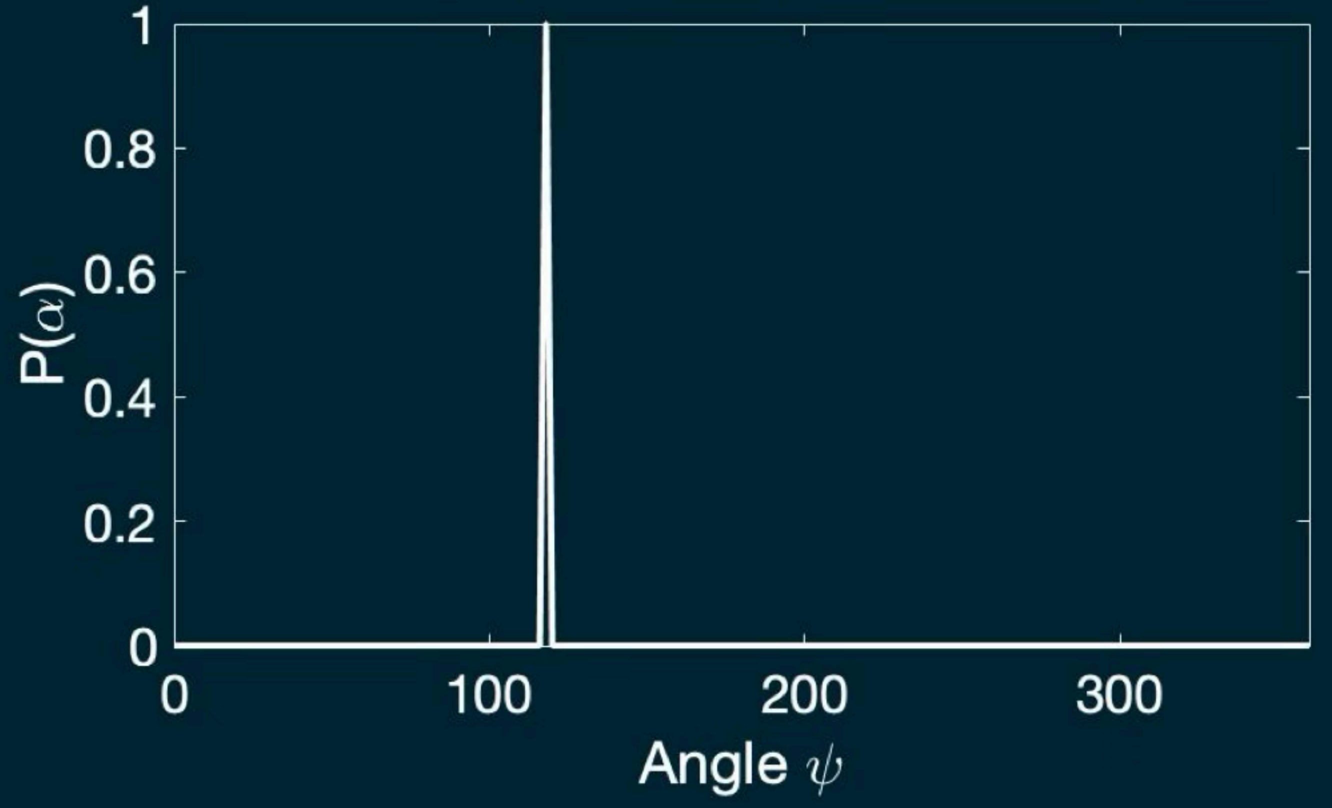




2. We place bounds on how much higher the probabilities could be at full resolution.

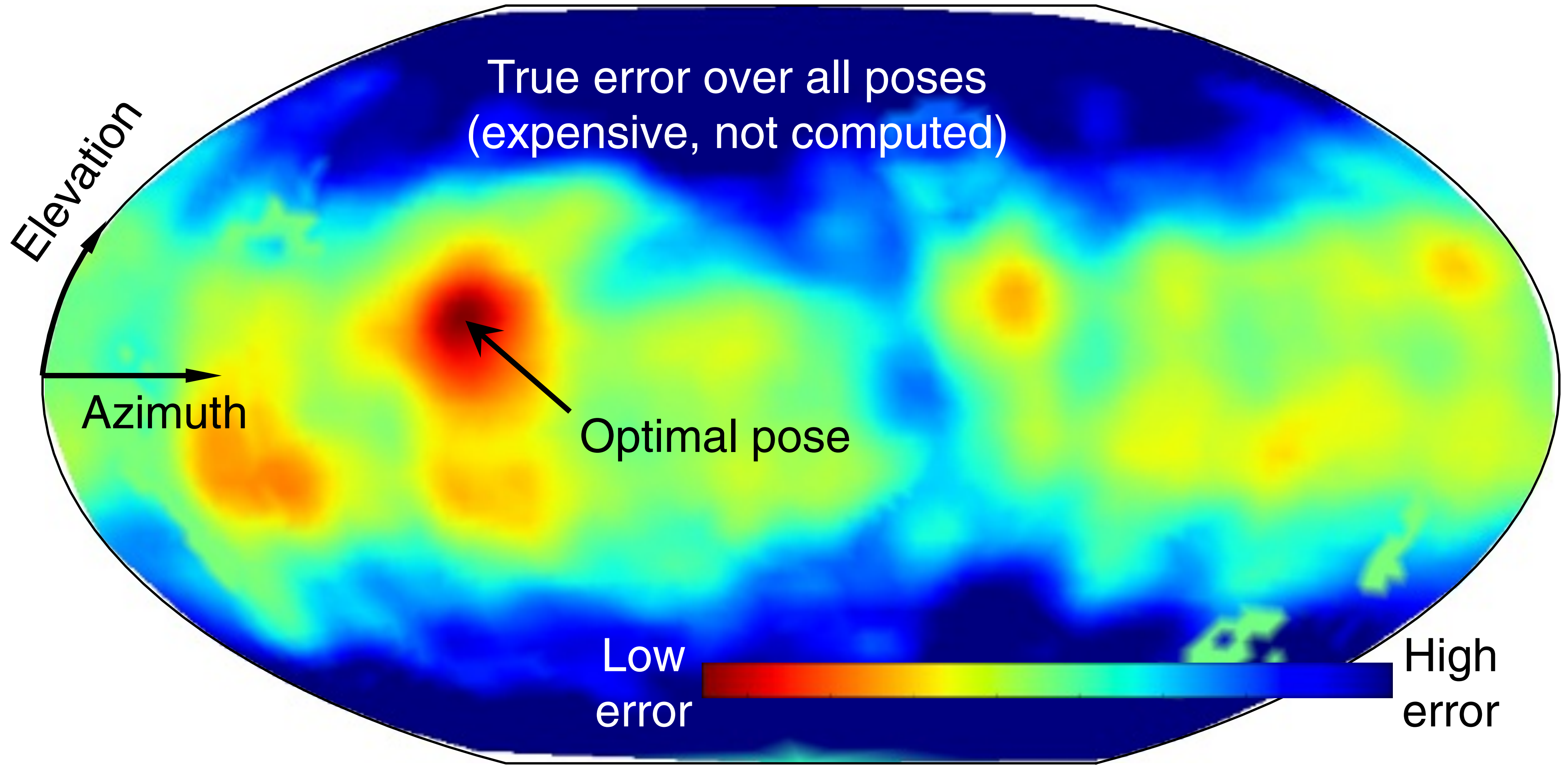
Given a cutoff value, we evaluate over a fraction of the domain.

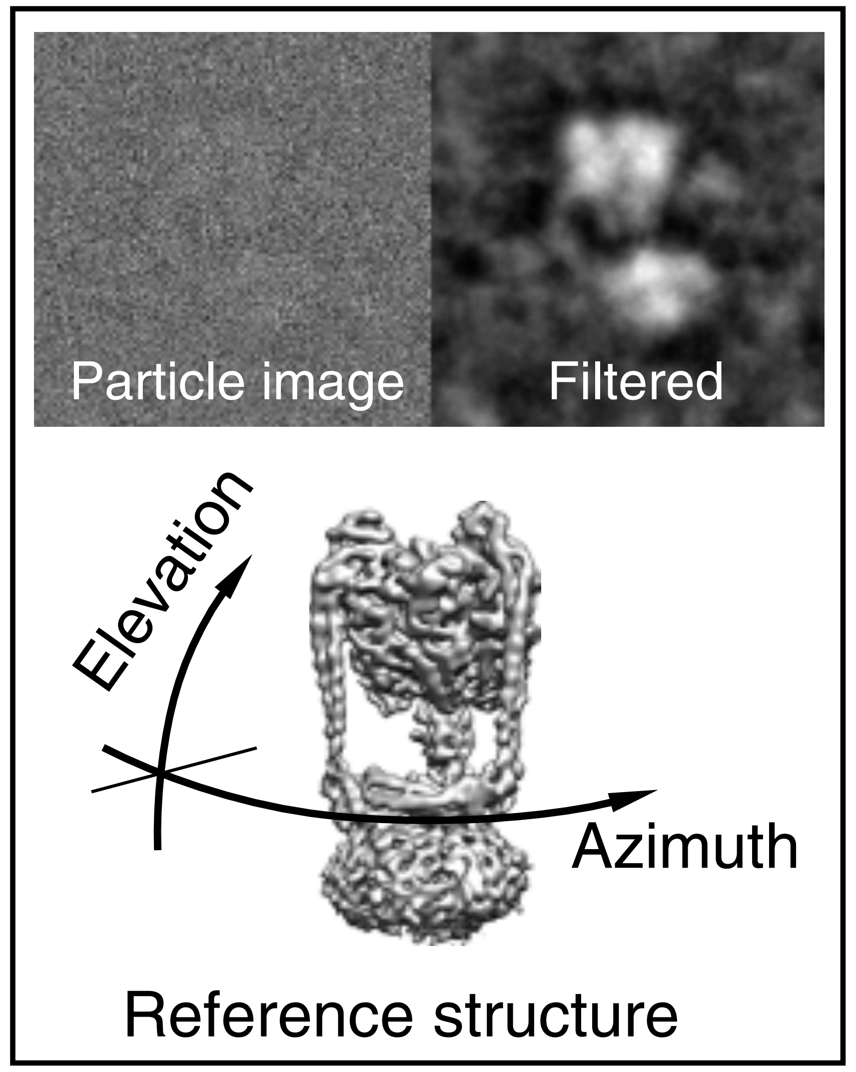
Then we search only the limited domain to find the just the highest probability values.



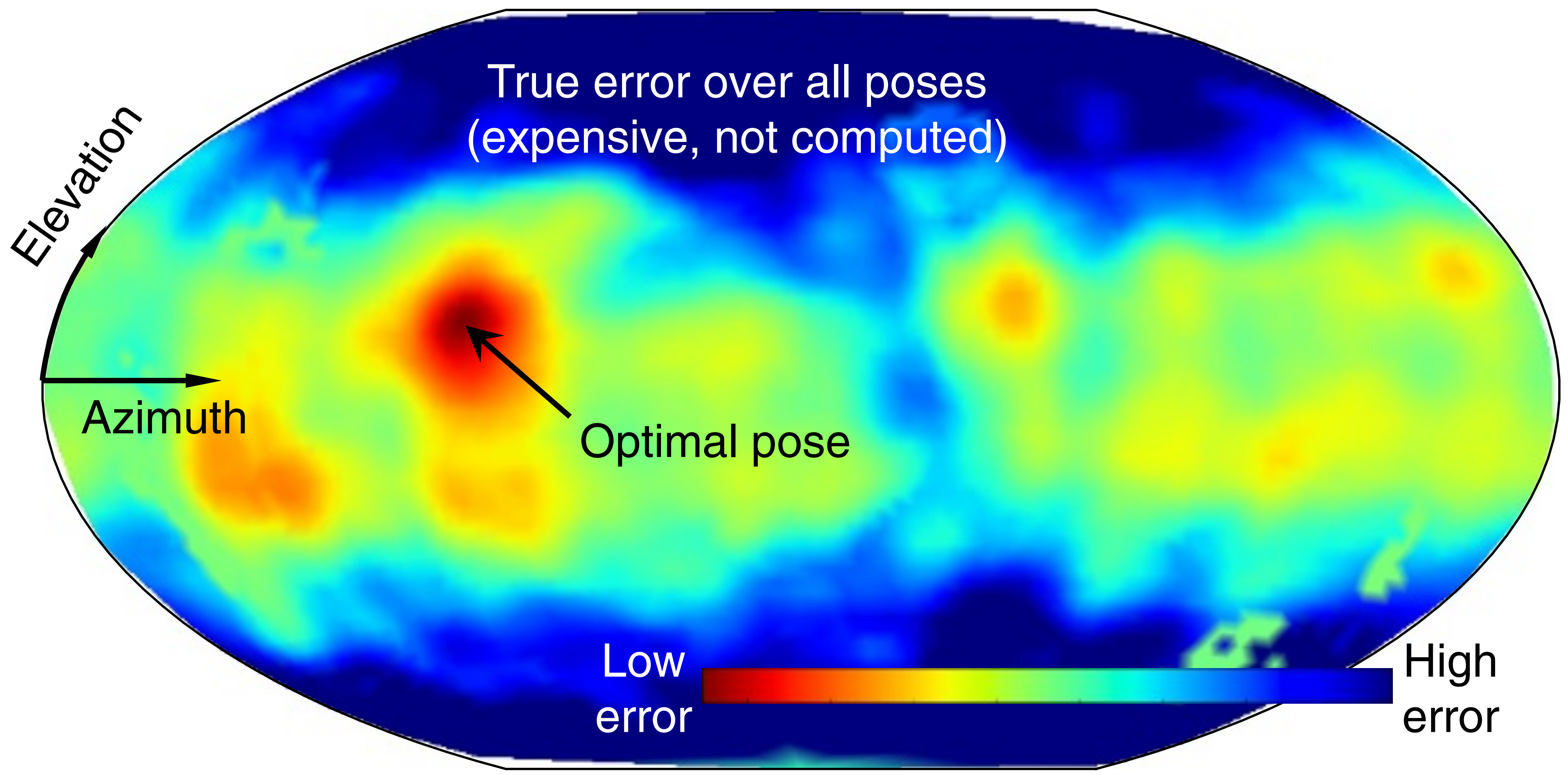
Branch-and-bound in cryoSPARC for integrating over orientations

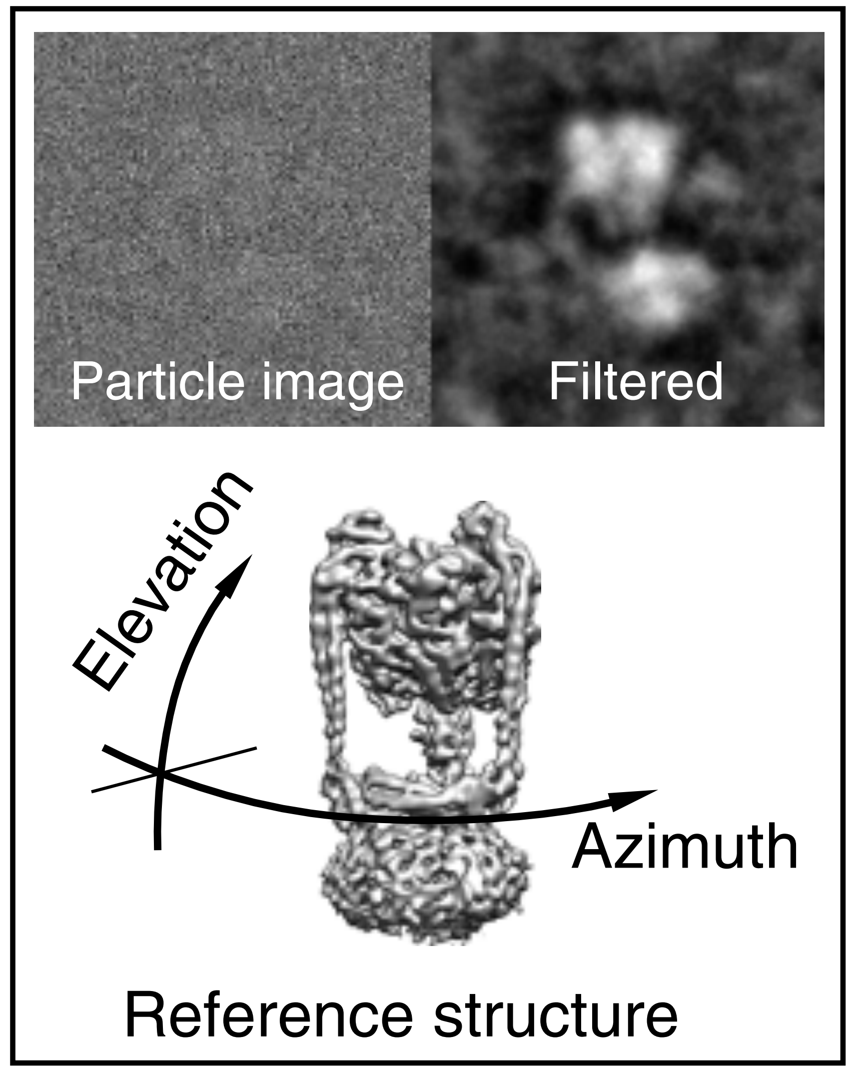
Here, “error” is just
$-\log[P(\mathbf{X}\,|\,\mathbf{V})]$

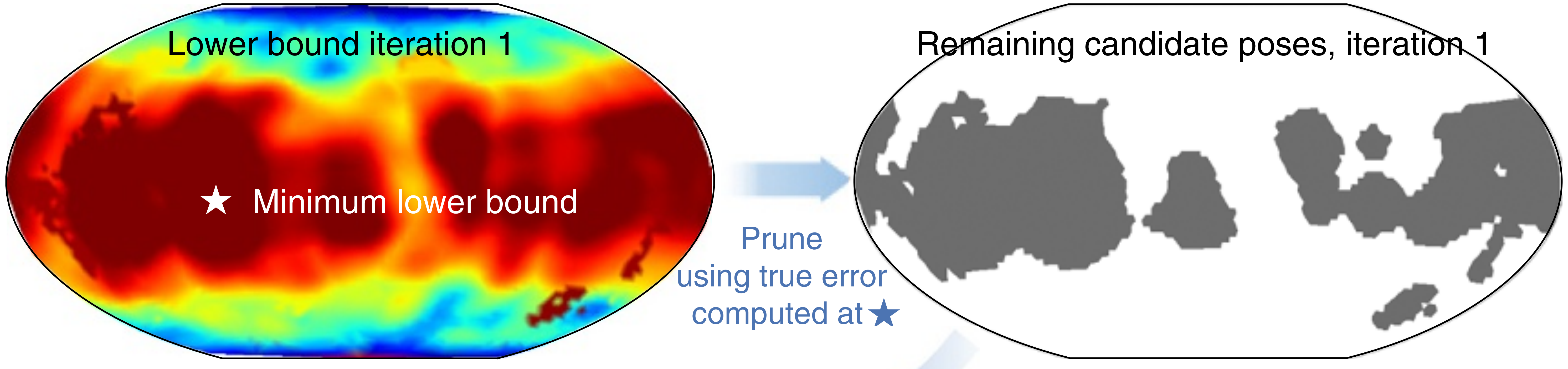




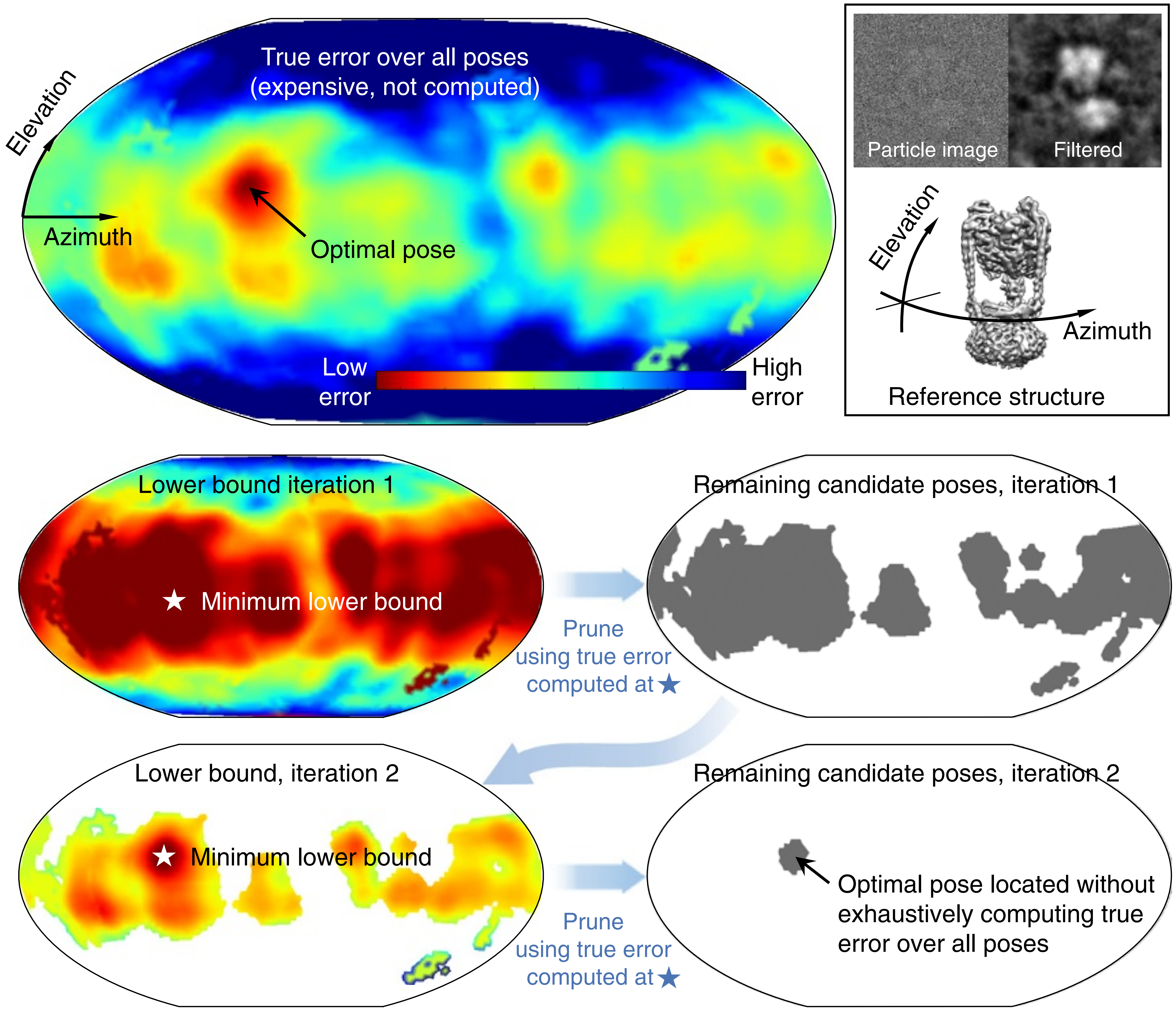
Branch-and-bound in cryoSPARC for integrating over orientations



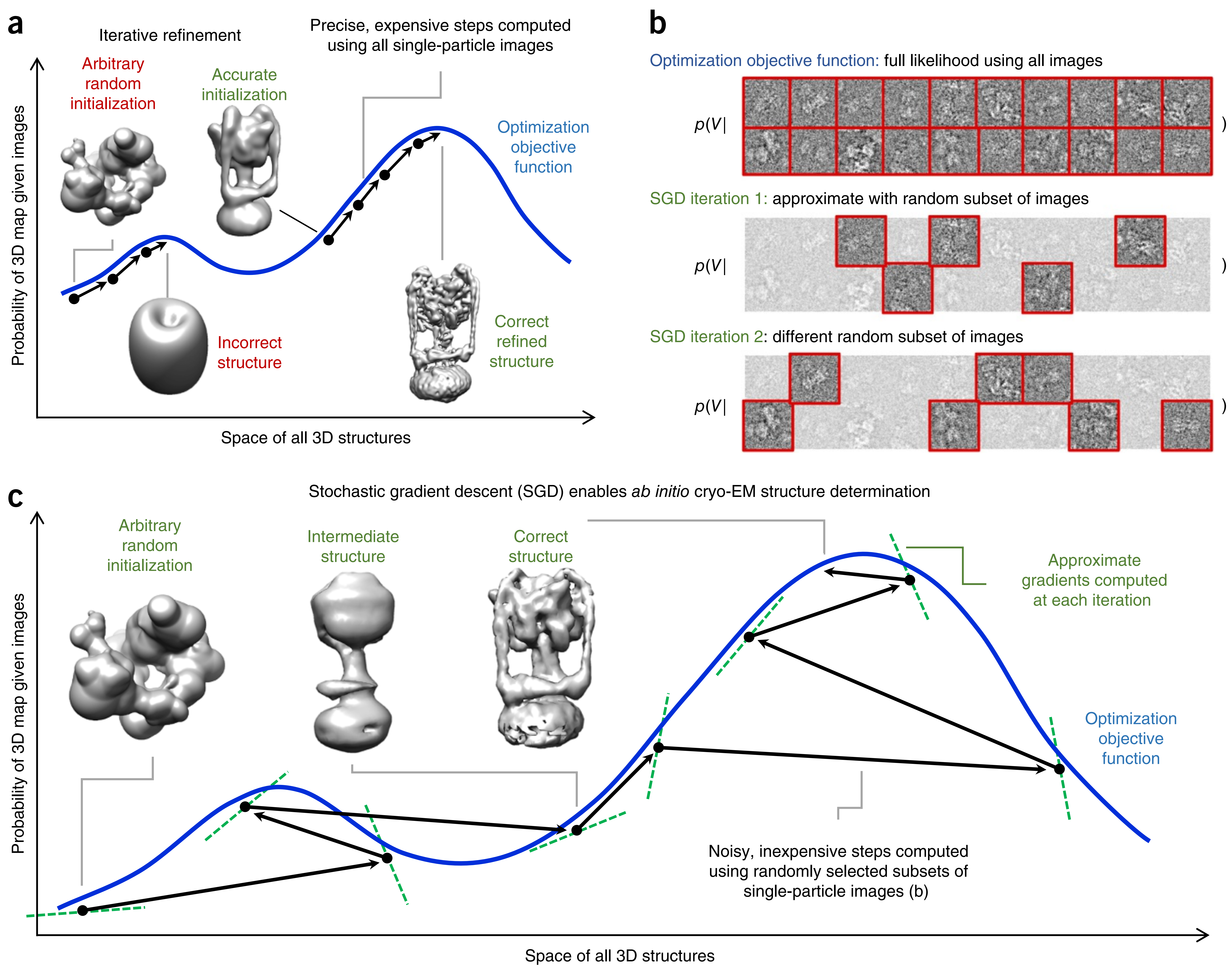


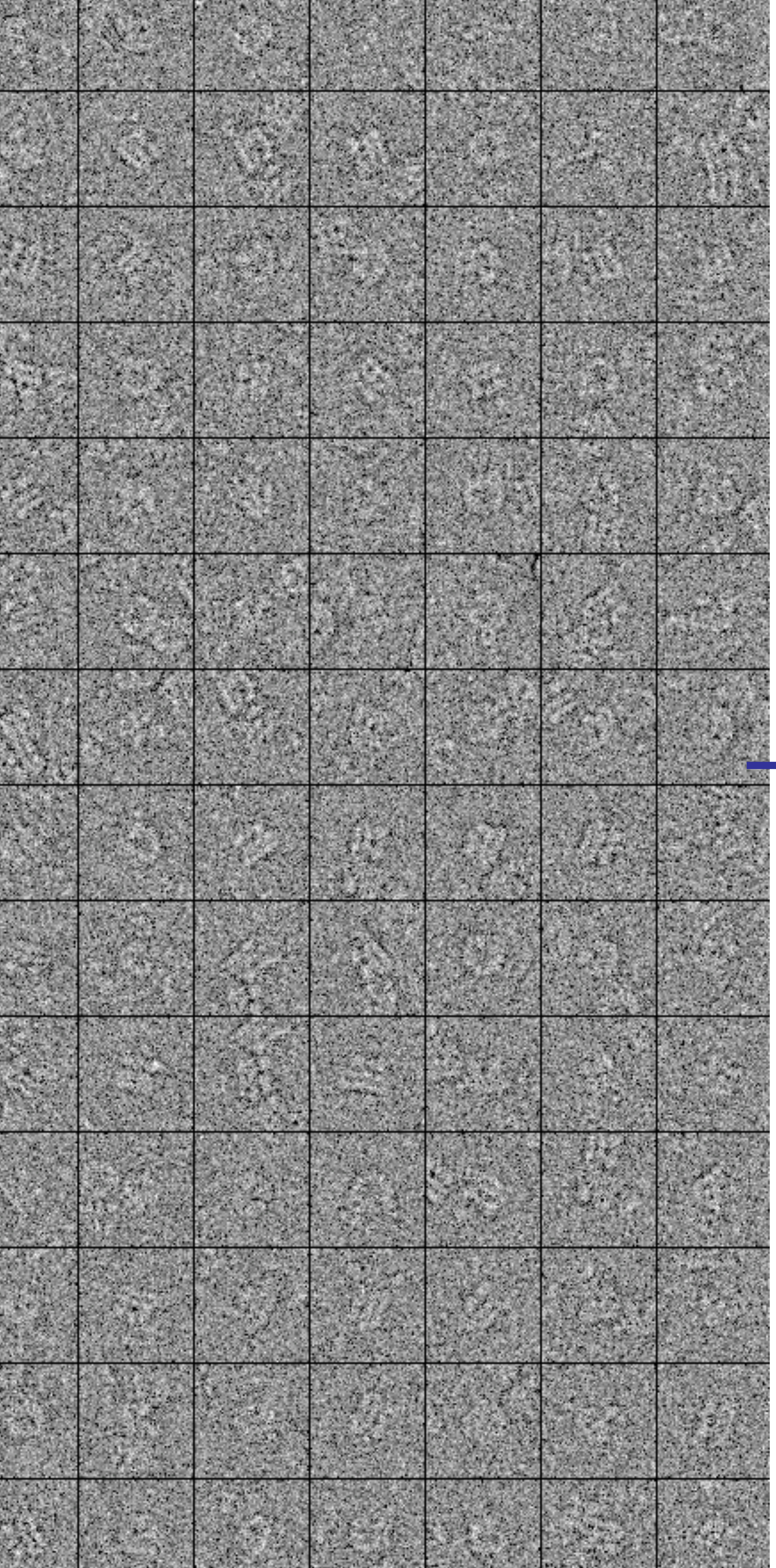


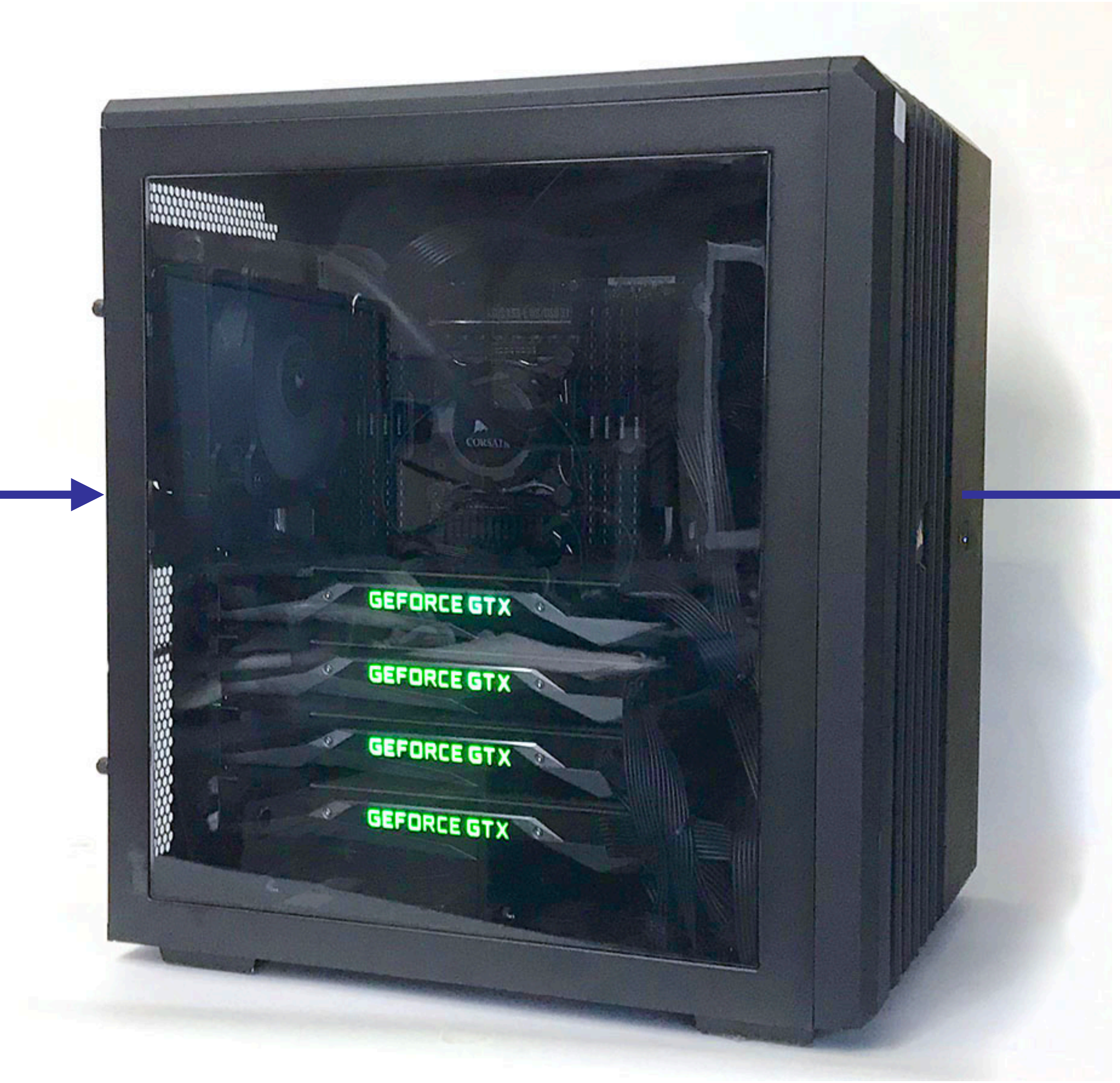
Branch-and-bound in cryoSPARC for integrating over orientations

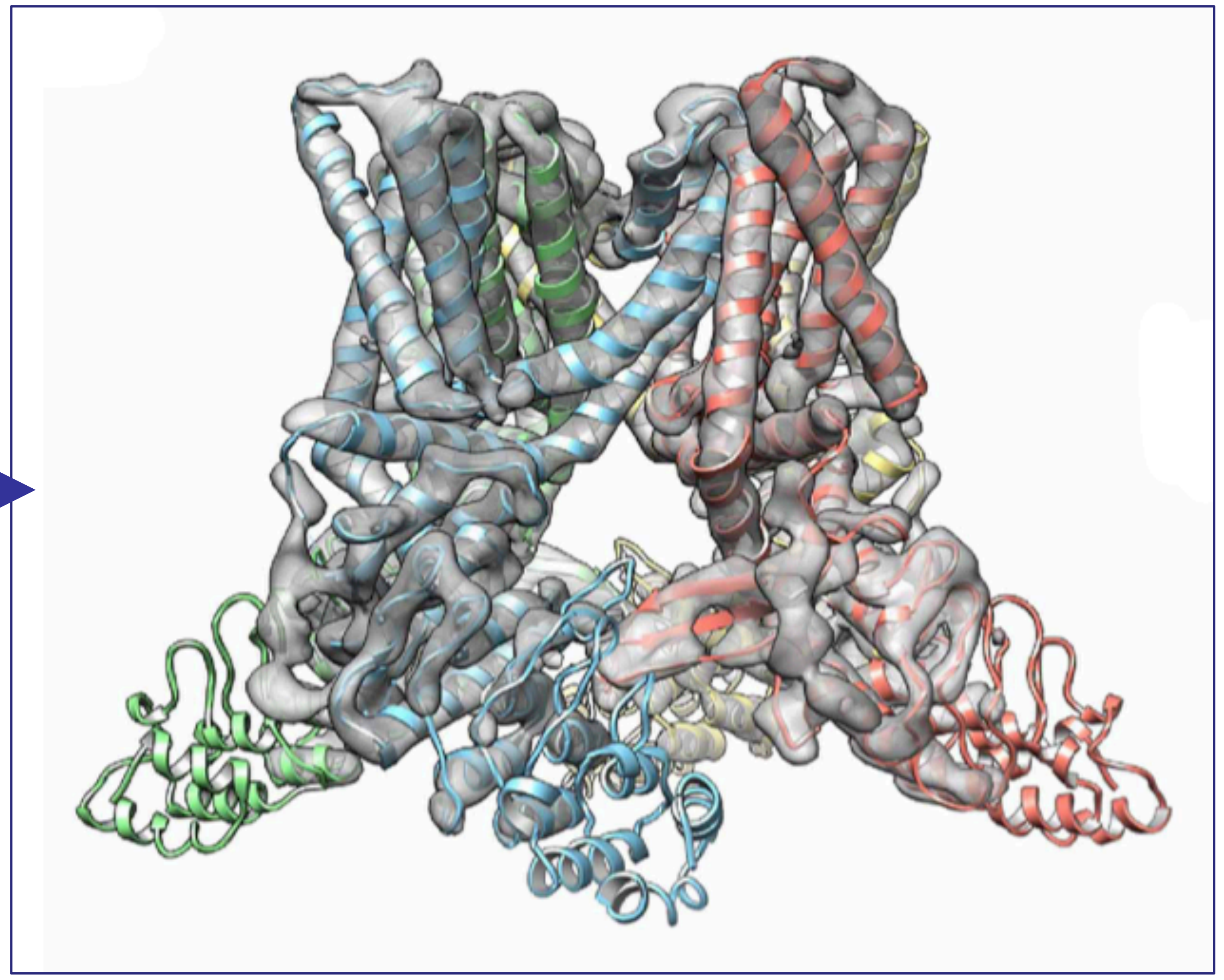


Stochastic gradient descent (ascent) to avoid model bias









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