Algorithms and Foundational Math

Part II

Theoretical basis of single-particle reconstruction

Correlation and particle picking

Single-particle reconstruction

Maximum-likelihood methods

Image processing with Fourier transforms

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

Convolution and correlation

Convolution

$$f(x, y) = g \star h$$

$$f(x, y) = \int\int g(x - s, y - t)\, h(s, t)\, ds\, dt$$

$$\rightarrow F(u, v) = G(u, v)H(u, v)$$

Correlation

$$c(x, y) = g \otimes h$$

$$c(x, y) = \int\int g(x + s, y + t)\, h(s, t)\, ds\, dt$$

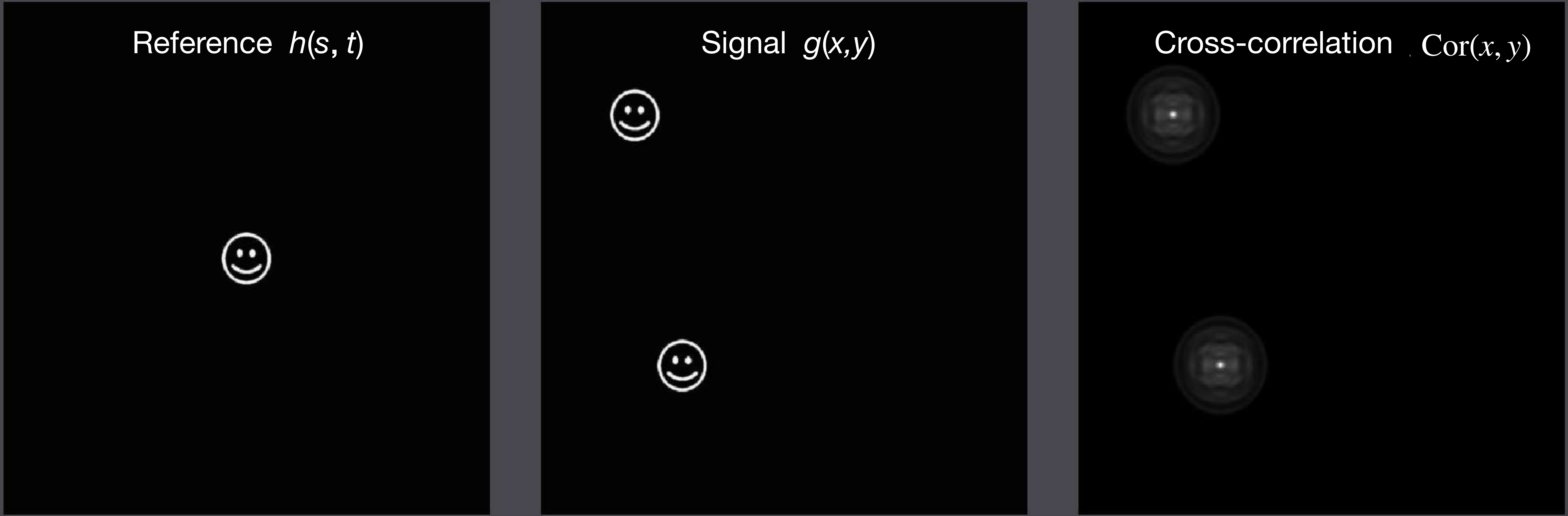
$$\rightarrow C(u, v) = G(u, v)H^*(u, v)$$

Correlation locates motifs in images

Translational cross-correlation function

$$\mathrm{Cor}(x,y) = X \otimes R$$
$$= \sum_{s,t} h(s,t)\, g(x+s, y+t)$$

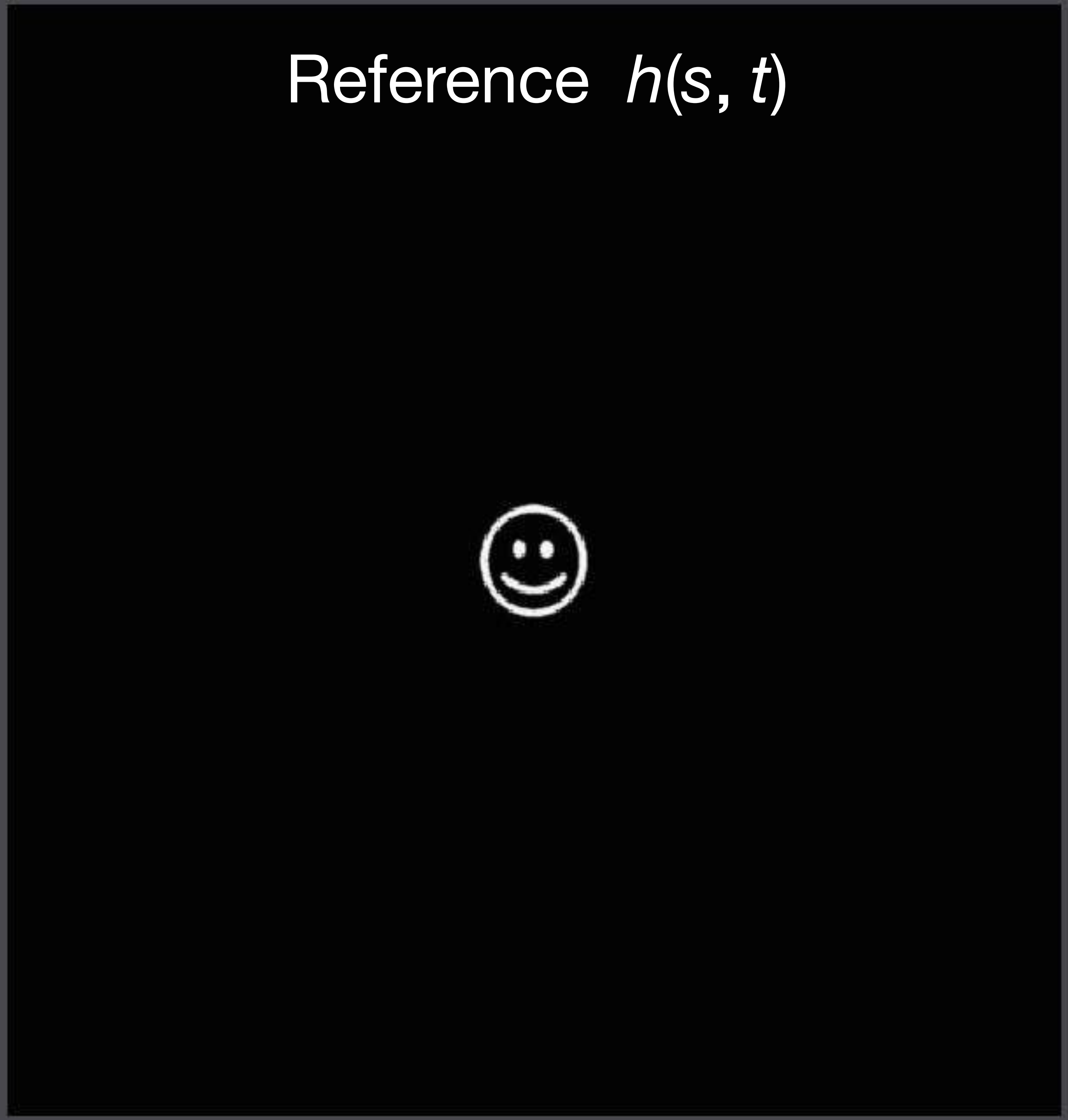
Correlation is like convolution.
The FT pair is: $g \otimes h \rightarrow GH^*$

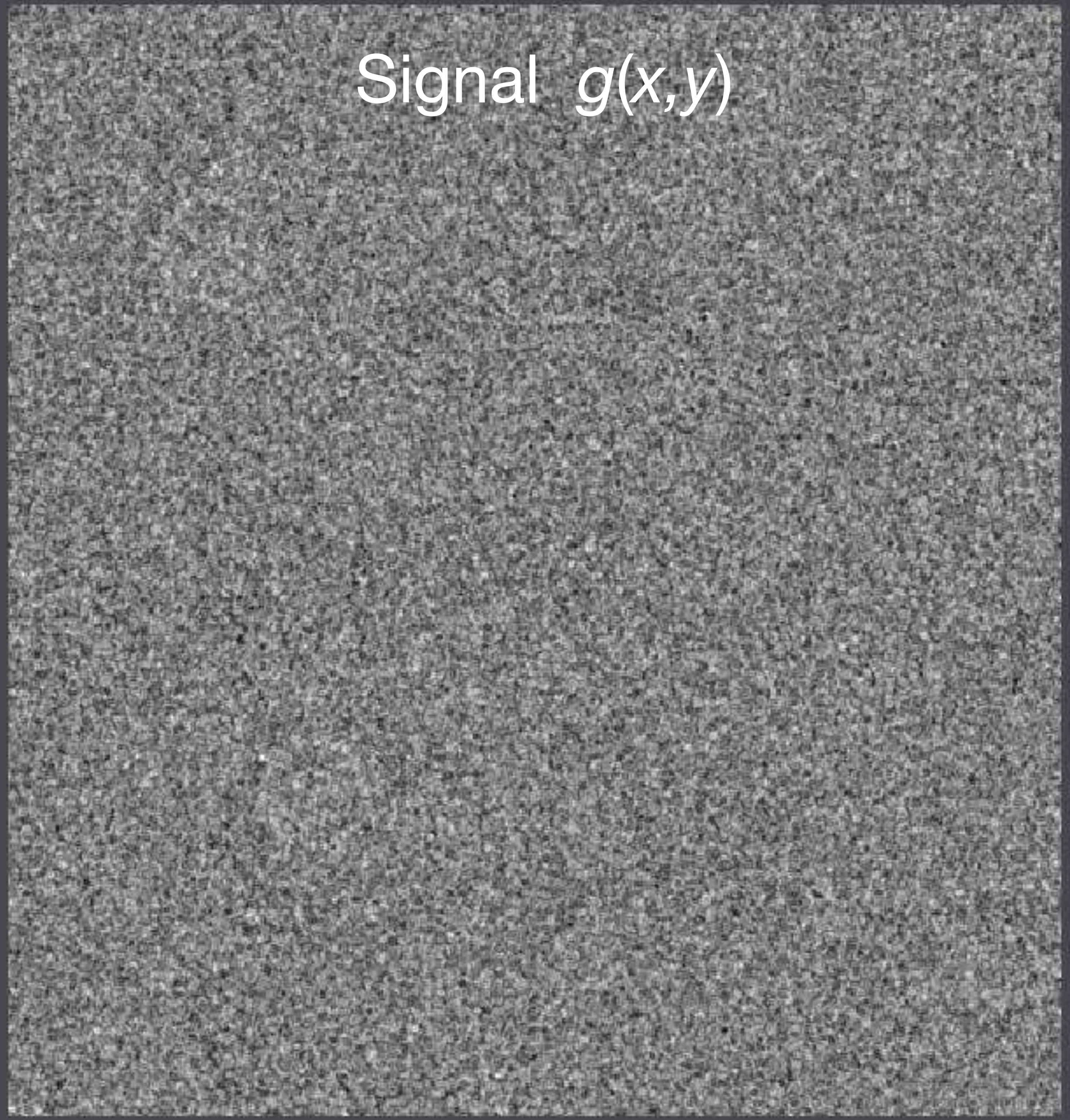


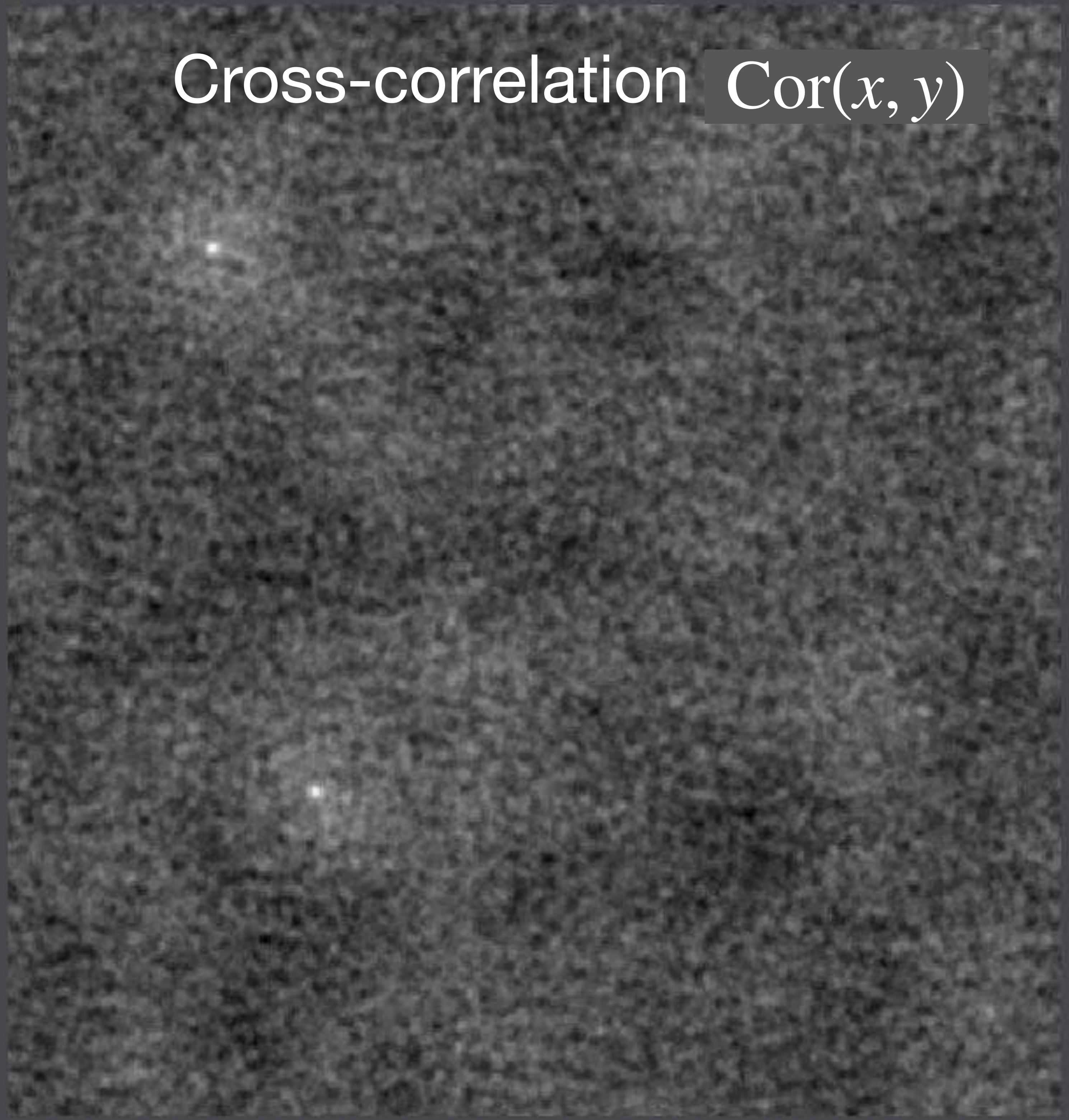
Correlation locates motifs in images

Translational cross-correlation function

$$\mathrm{Cor}(x, y) = X \otimes R = \sum_{s,t} h(s, t)\, g(x+s, y+t)$$

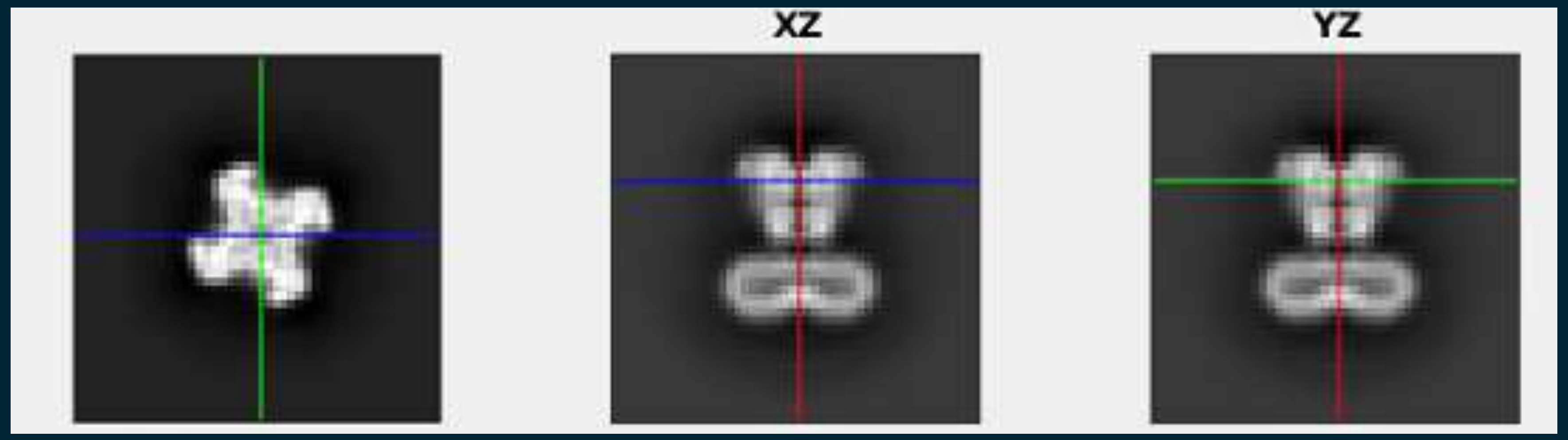




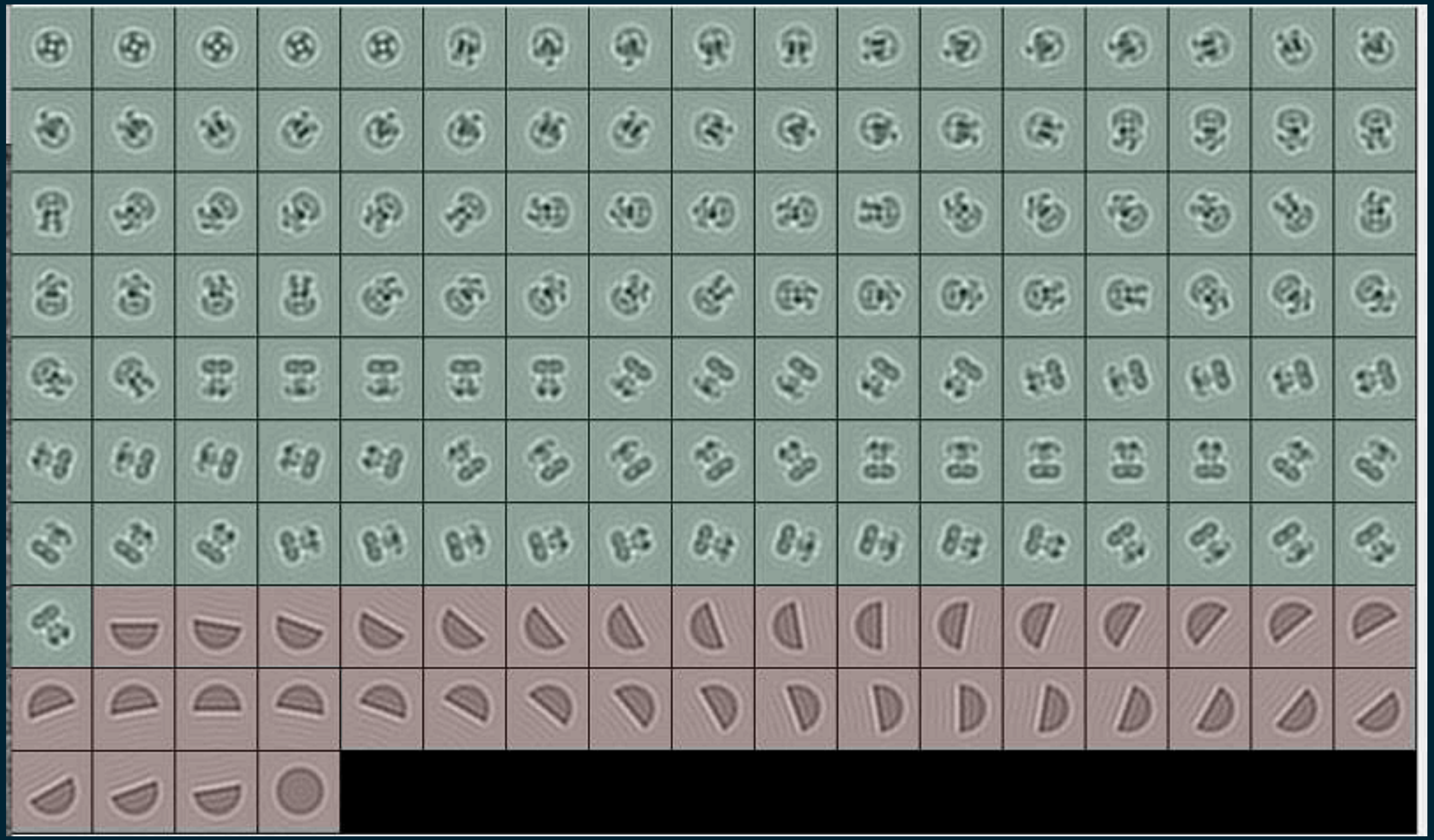


A correlation-based particle picker

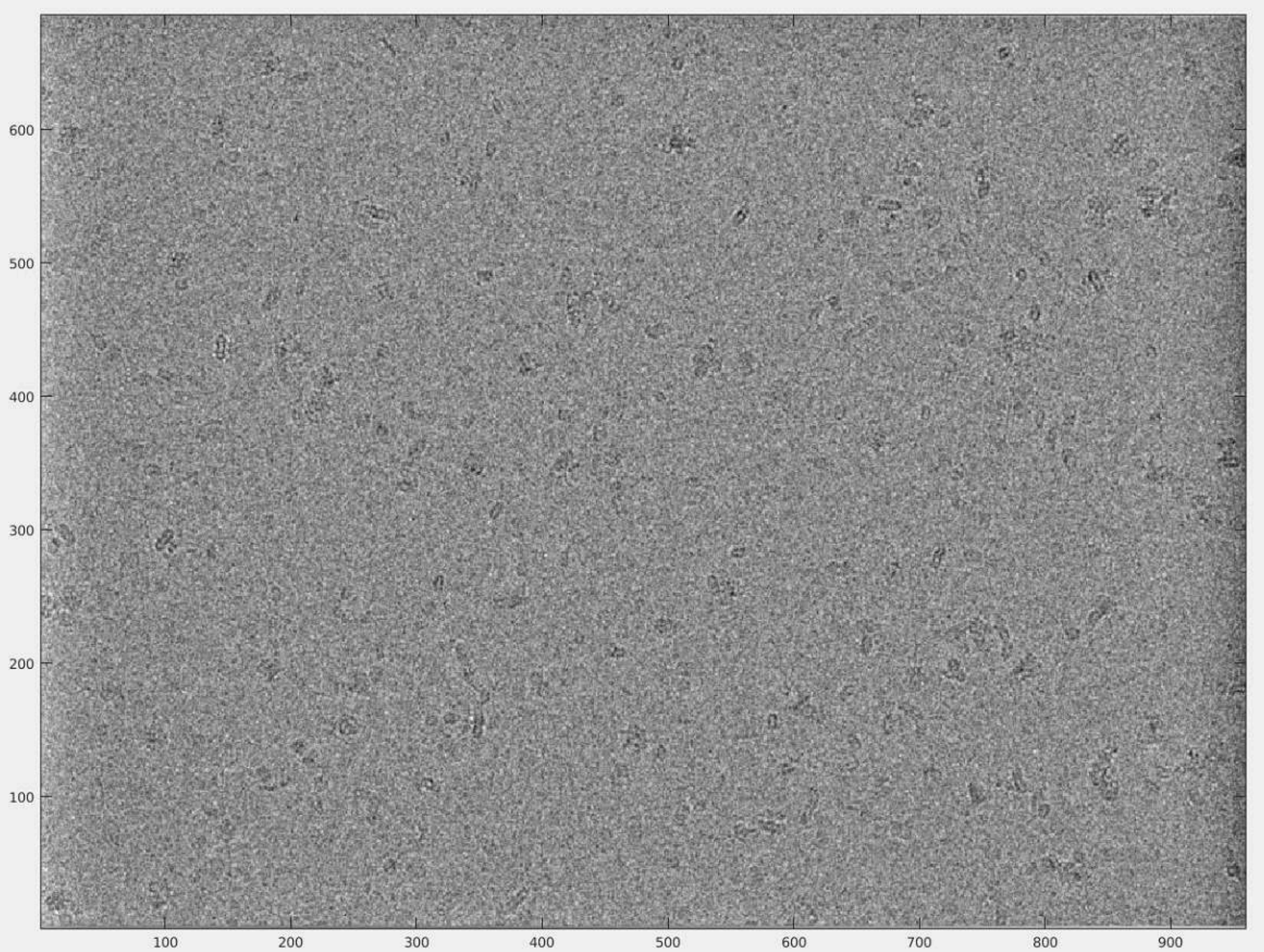
3D Reference



CTF-filtered projections and decoys

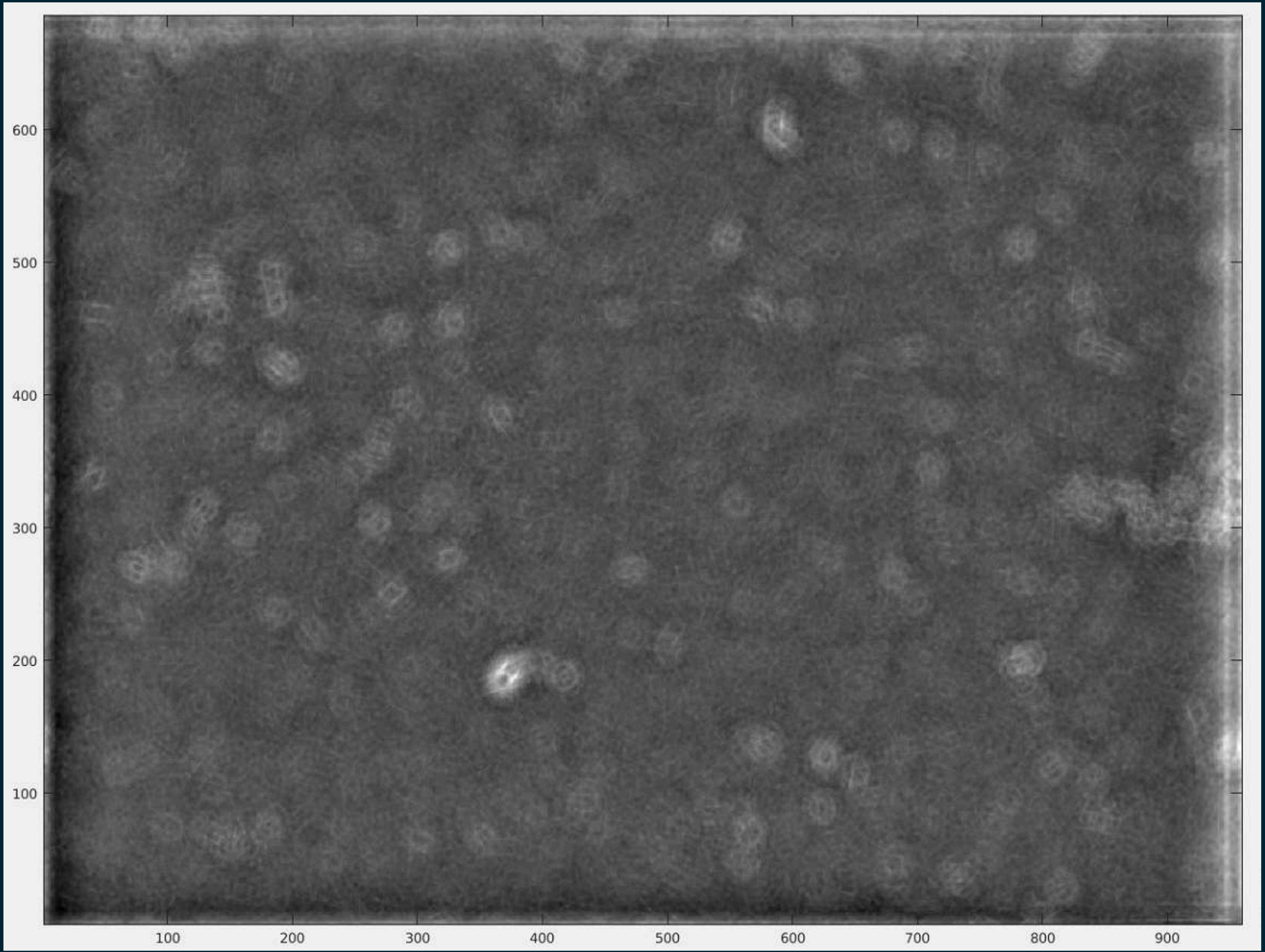


A correlation-based particle picker



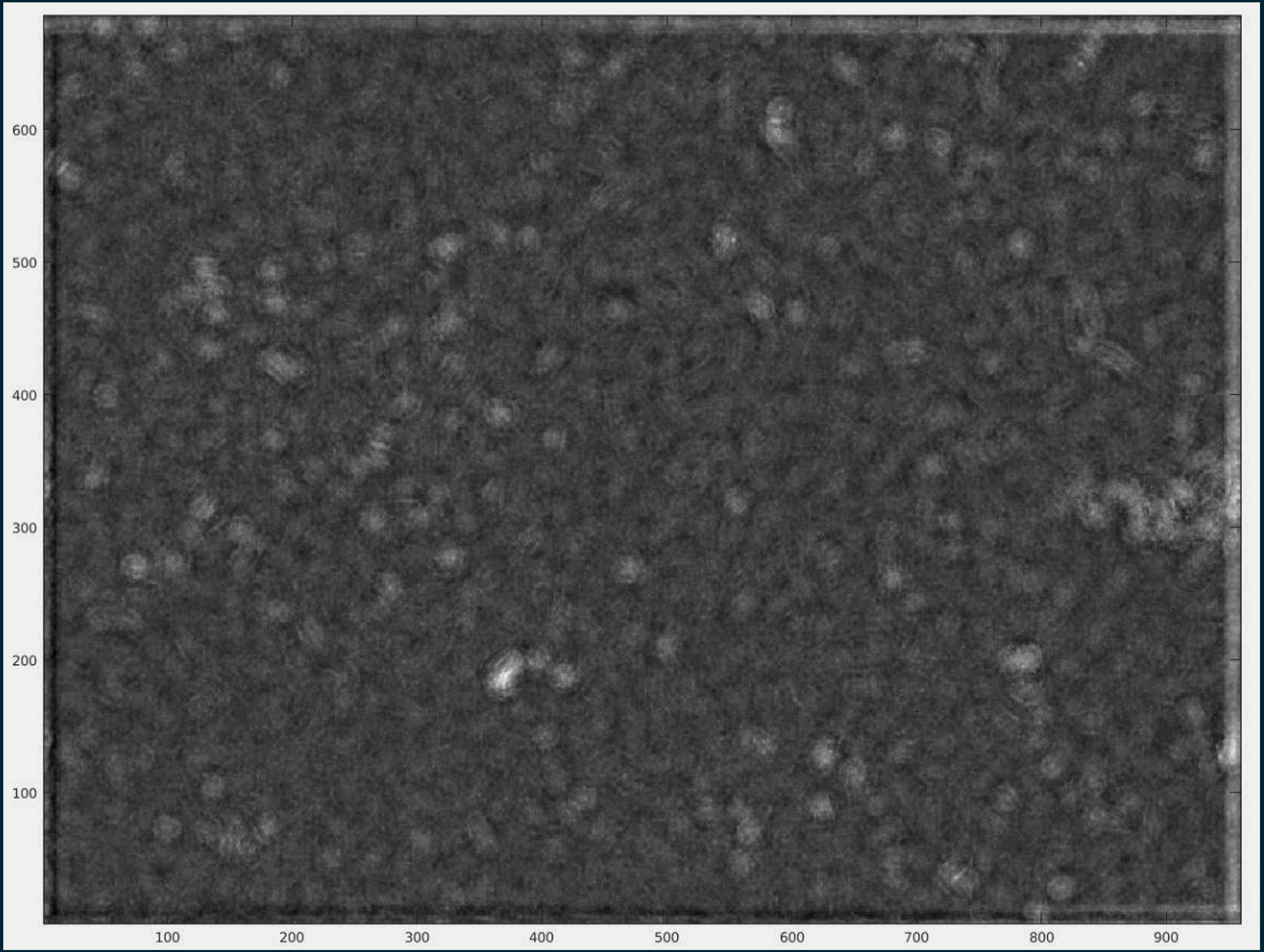
A micrograph

A correlation-based particle picker



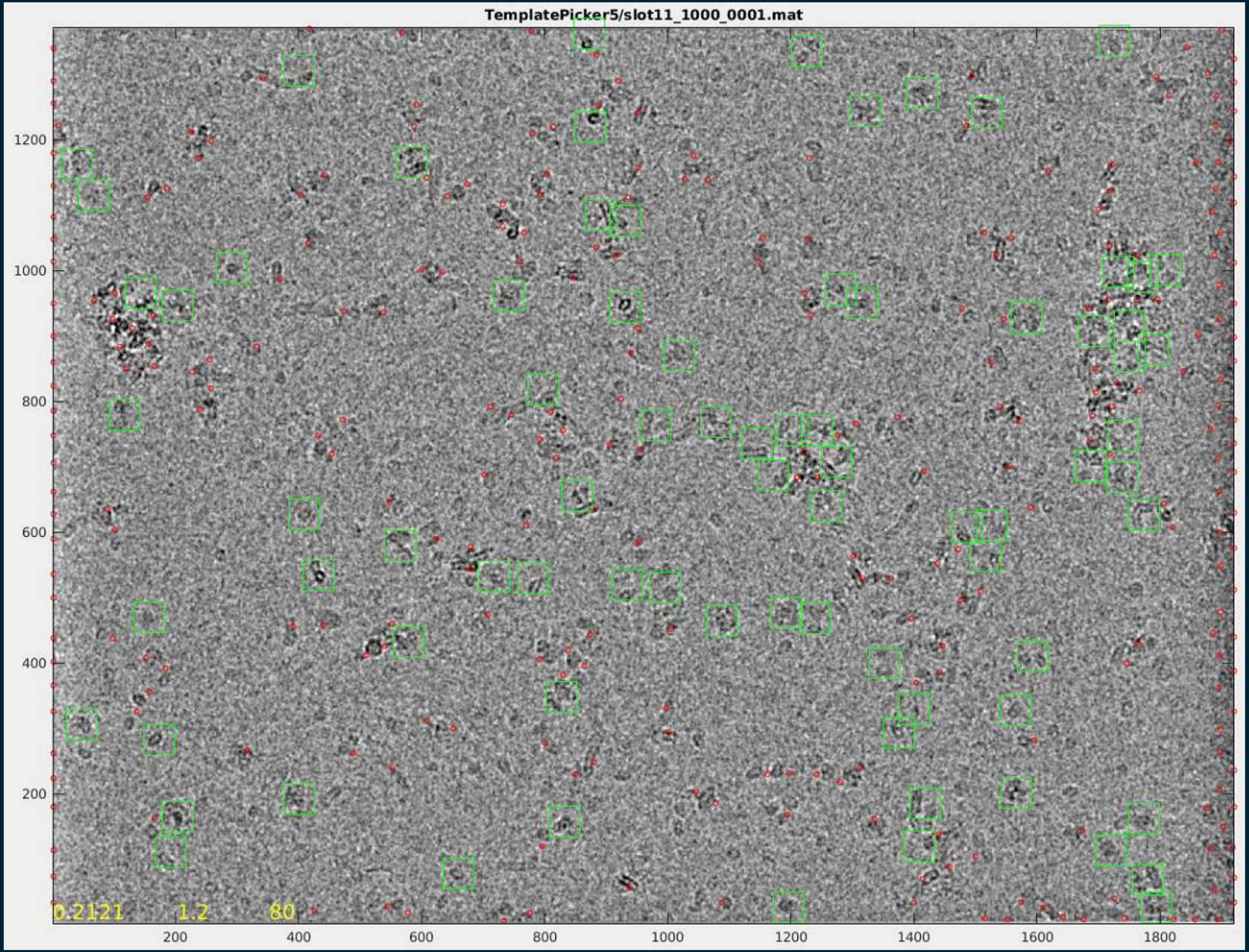
Max of correlations with decoy references

A correlation-based particle picker



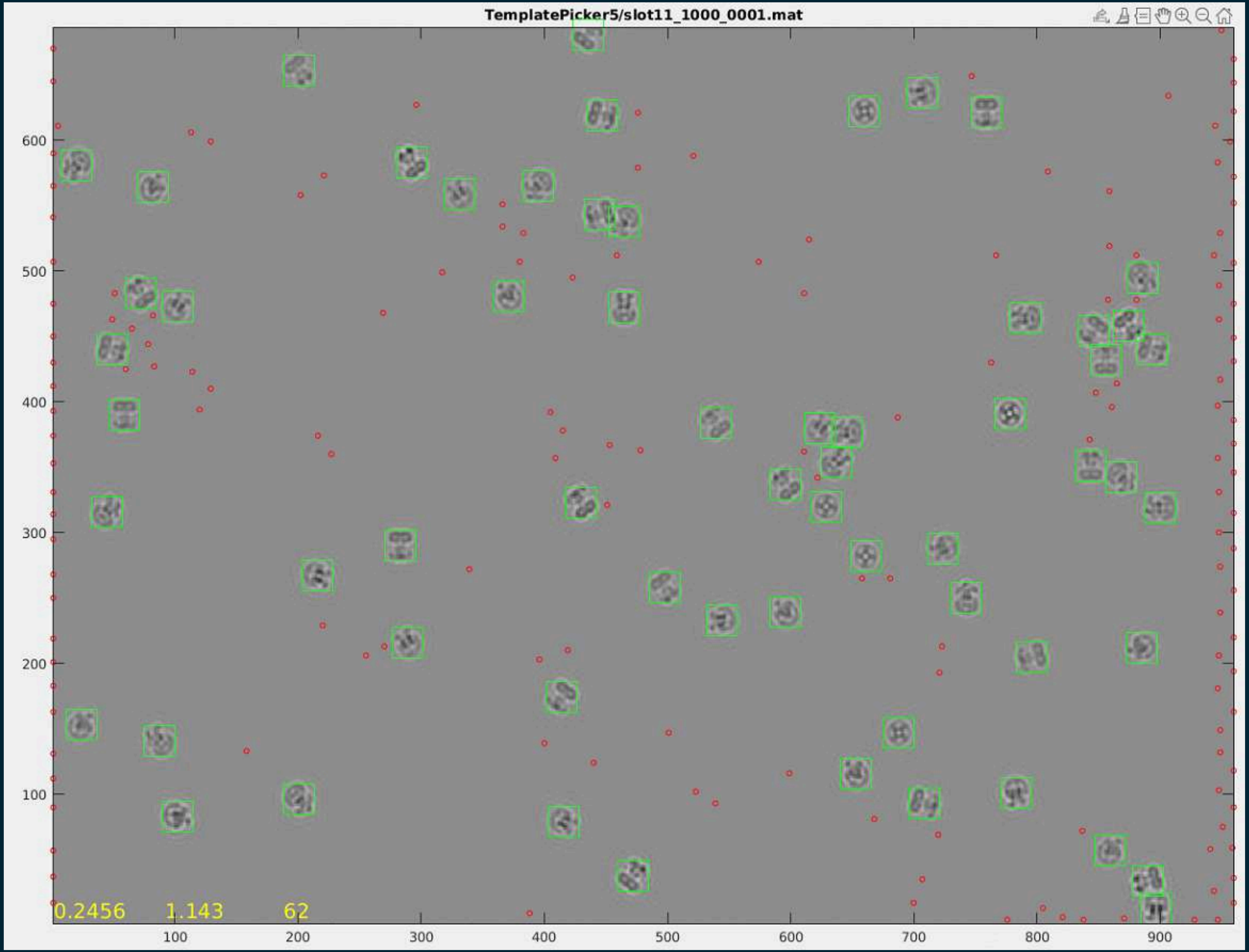
Max of correlations with particle references

A correlation-based particle picker



Green: found particles
Red dots: decoys

A correlation-based particle picker



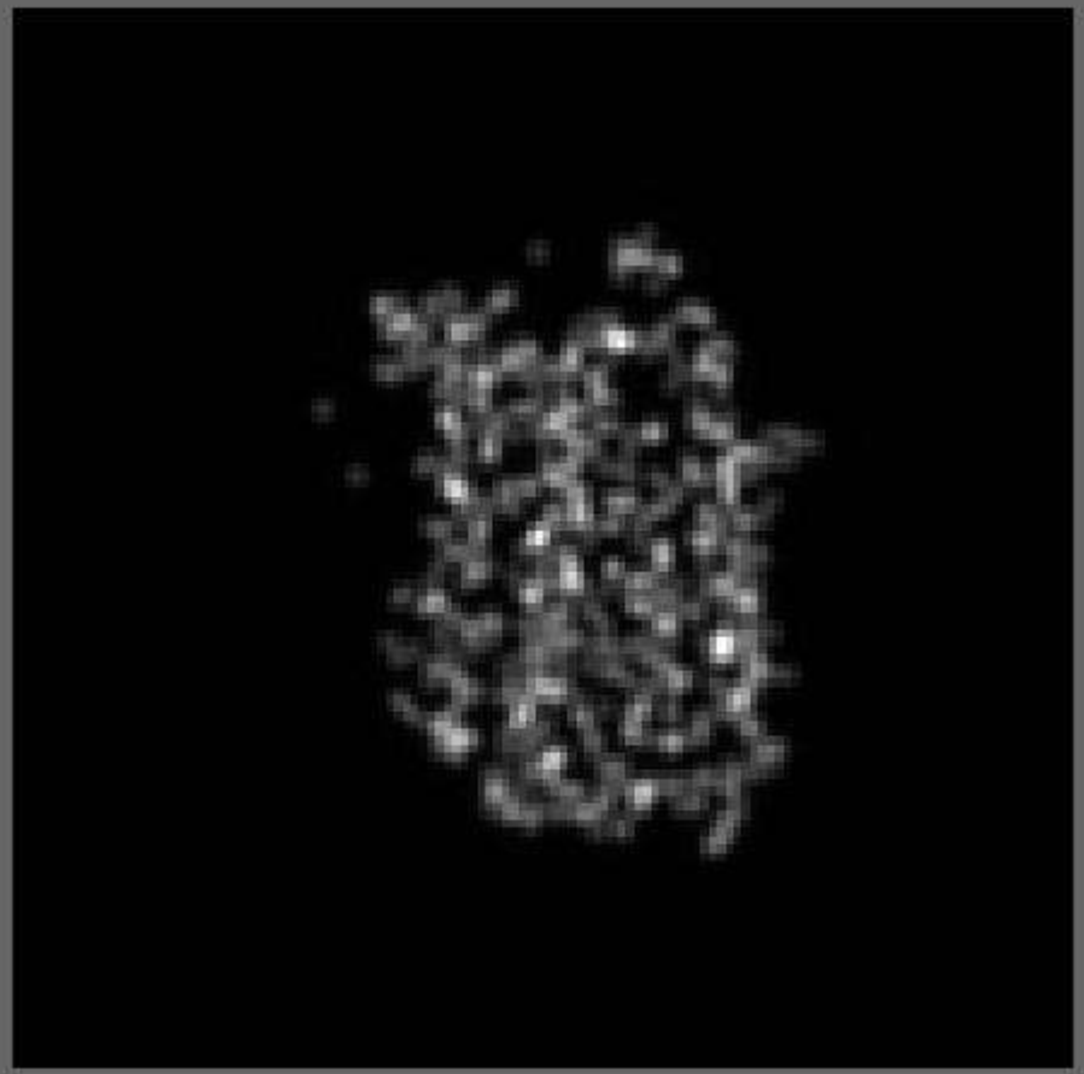
Best-matching references

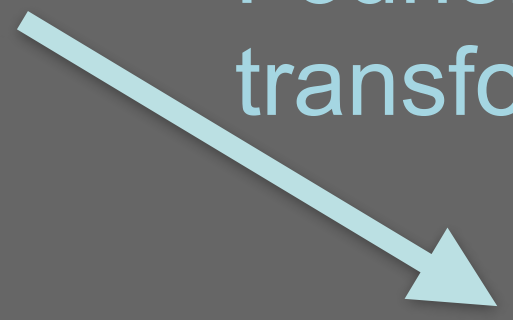
Correlation and particle picking

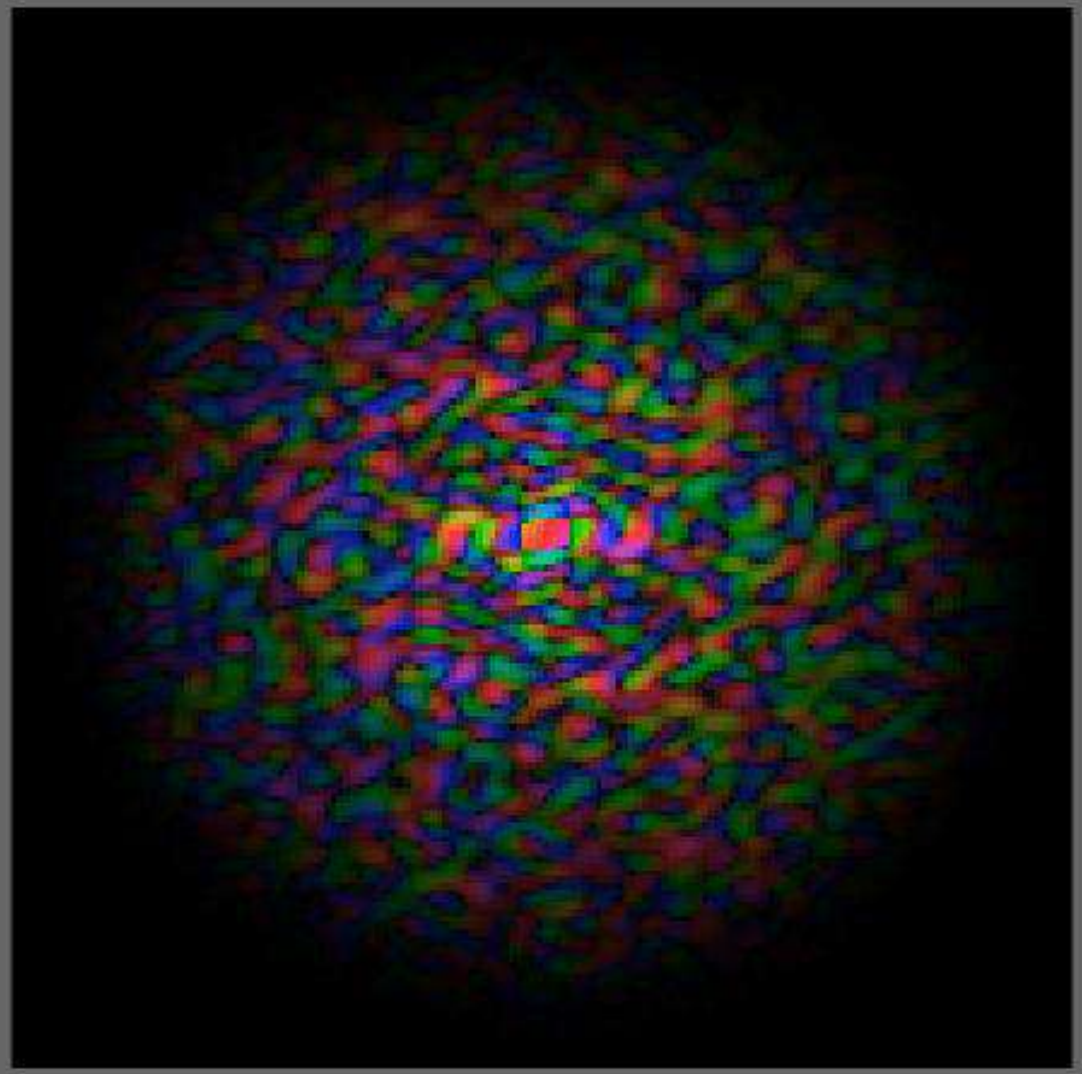
Single-particle reconstruction

Maximum-likelihood methods

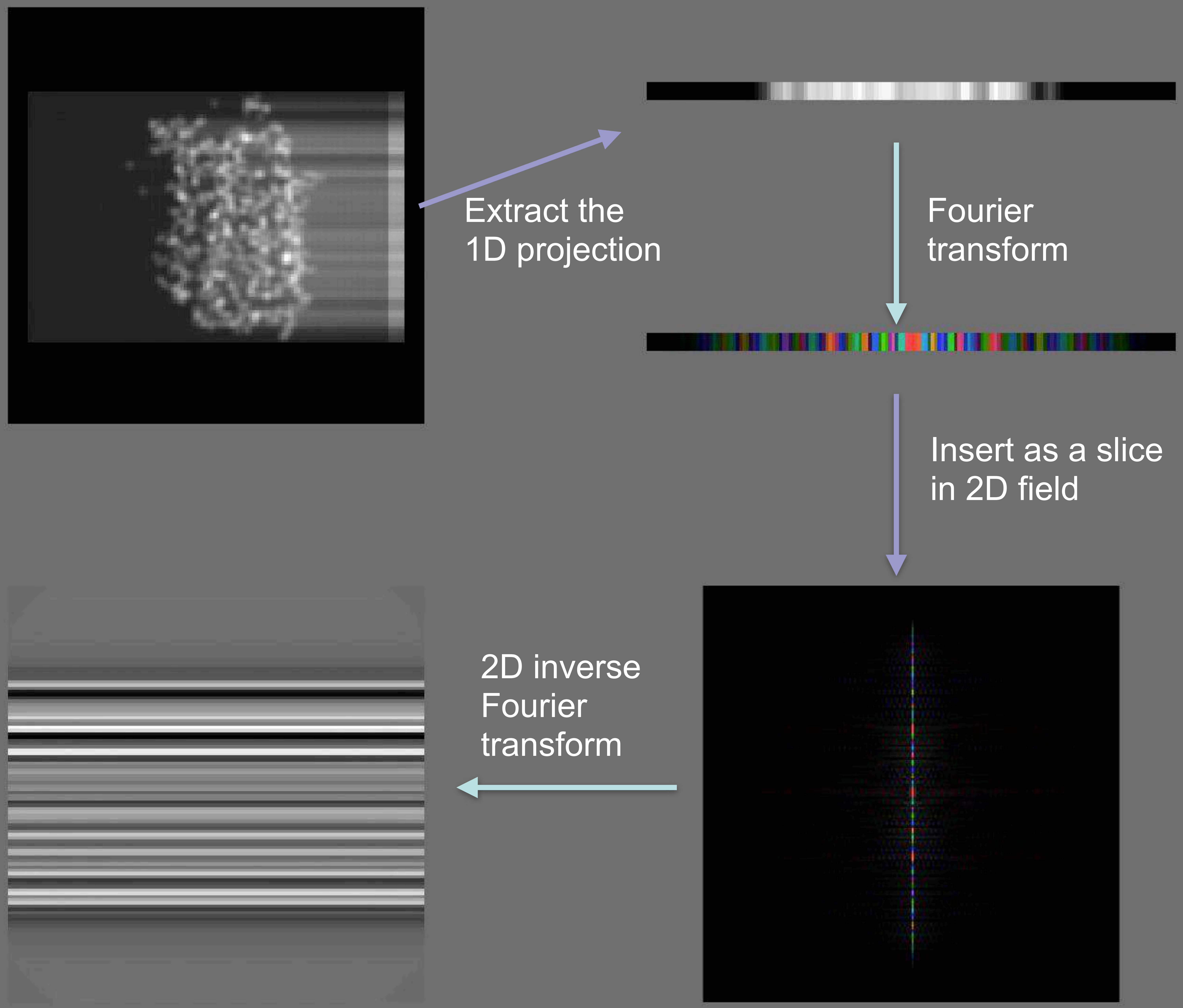
How to get 3D structures from 2D images? The Fourier slice theorem



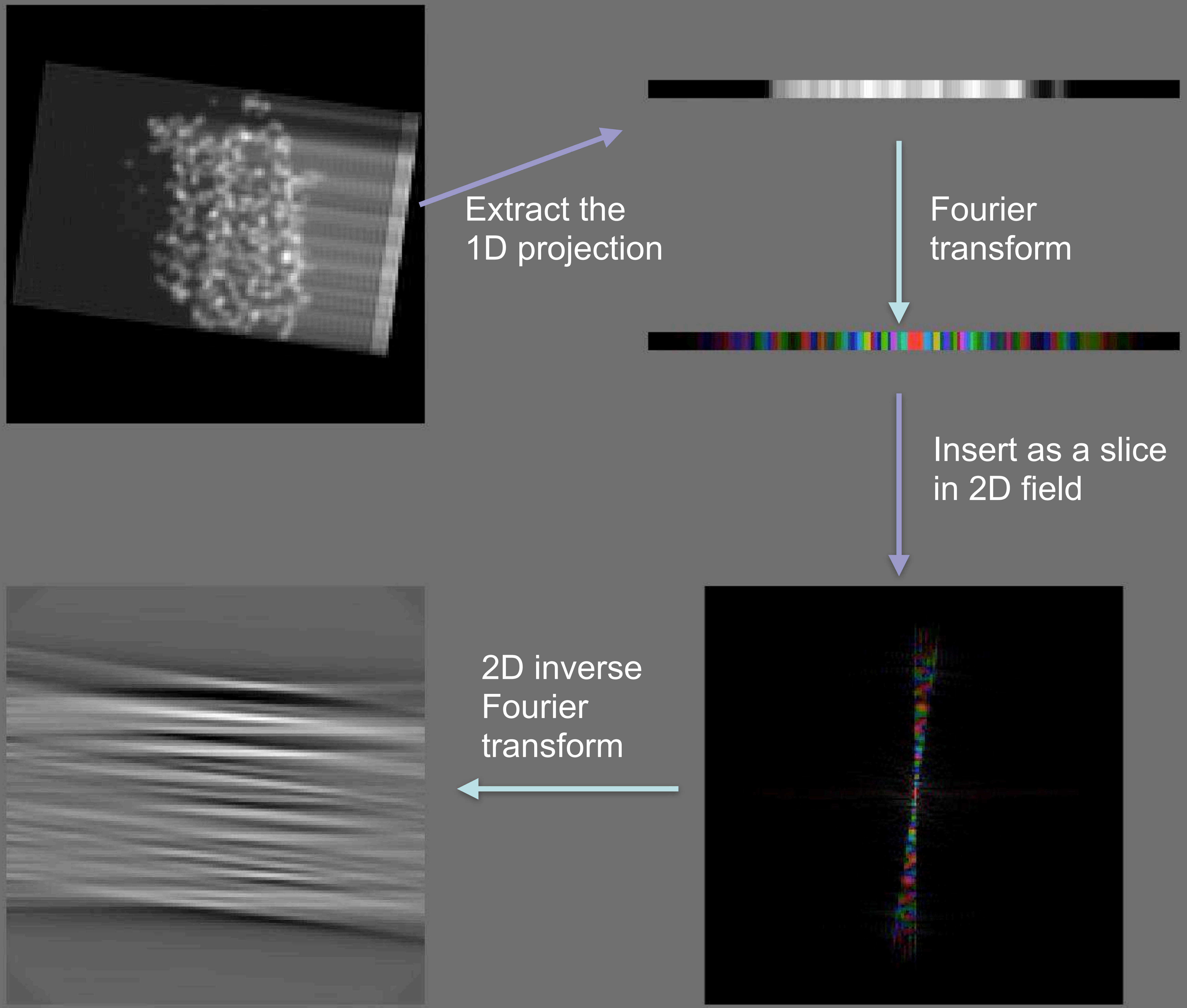




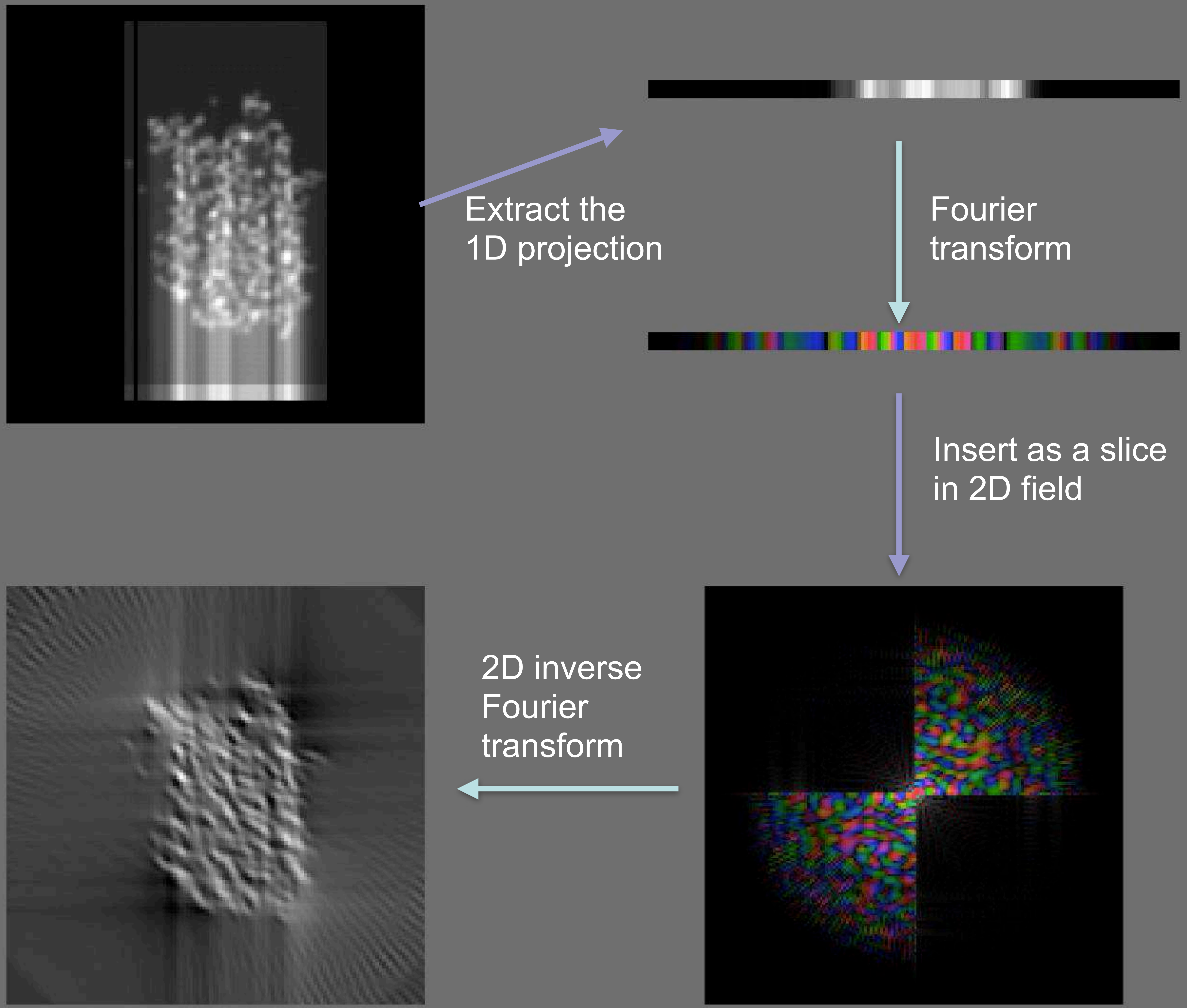
Tomographic reconstruction: 2D image from 1D projections



Tomographic reconstruction: 2D image from 1D projections



Tomographic reconstruction: 2D image from 1D projections

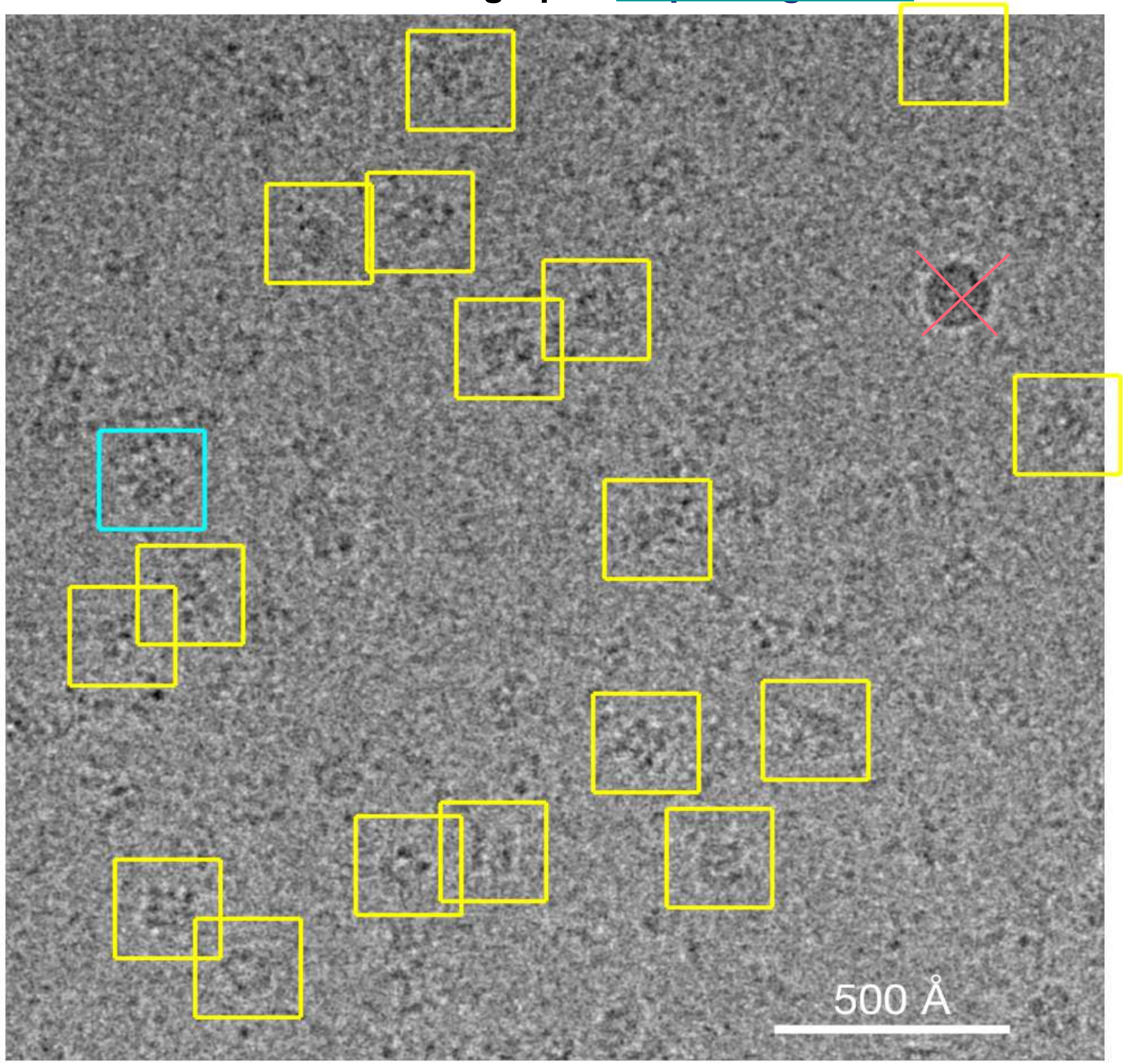


Determining the orientation angles: example from the TRPV1 dataset

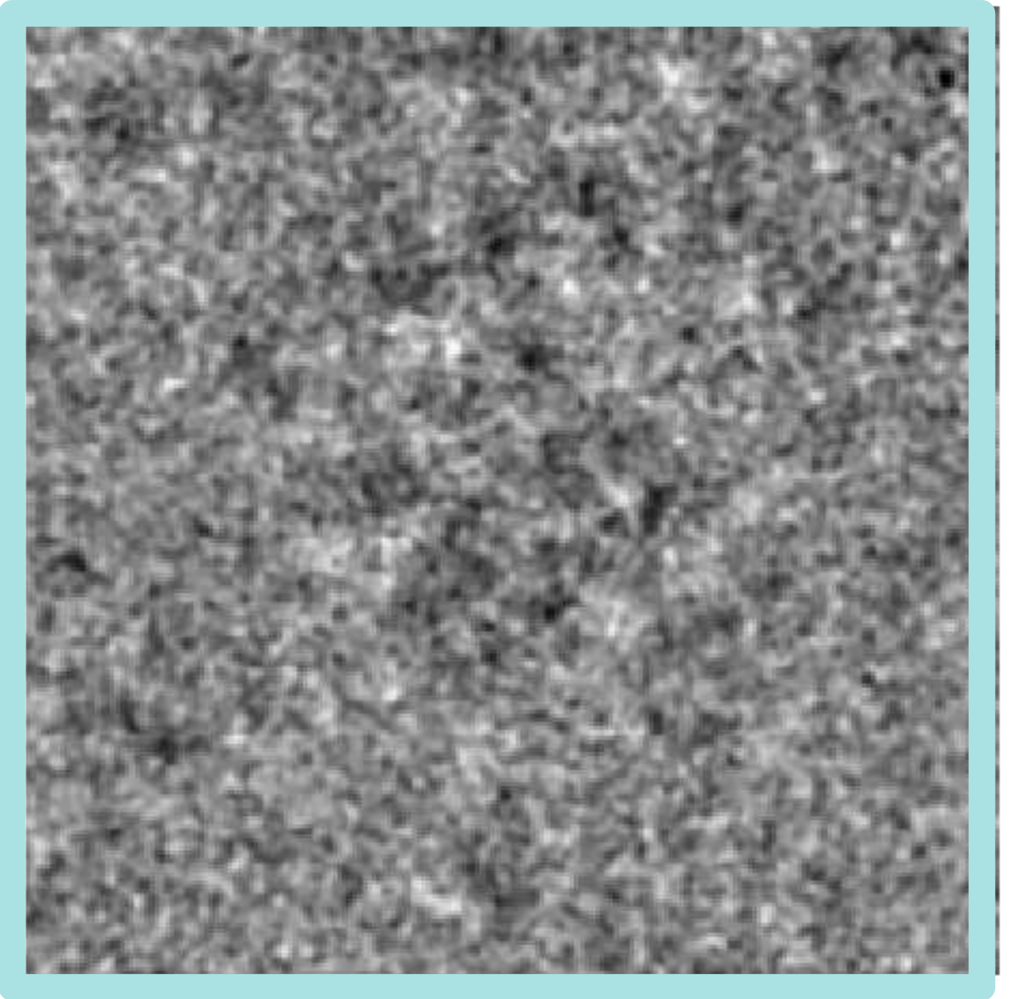
Structure of the TRPV1 ion channel determined by electron cryo-microscopy

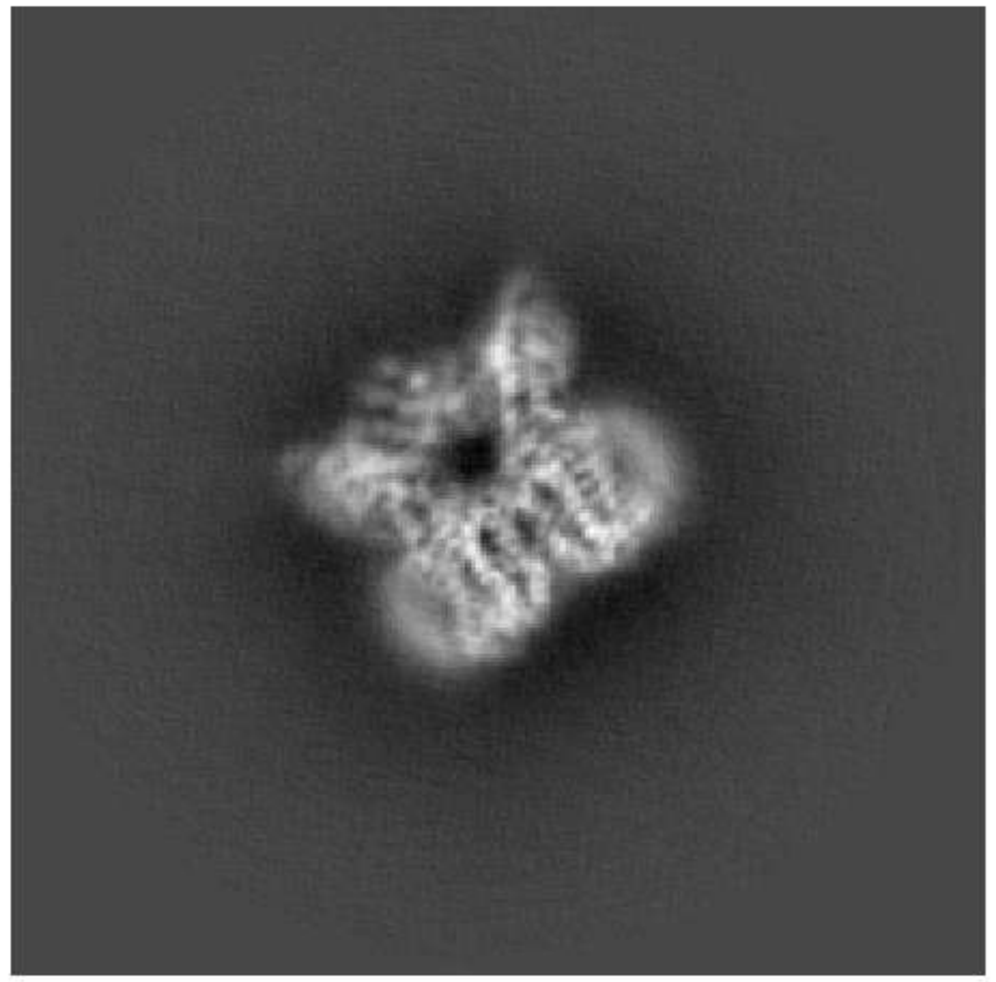
Maofu Liao[1]*, Erhu Cao[2]*, David Julius[2] & Yifan Cheng[1]

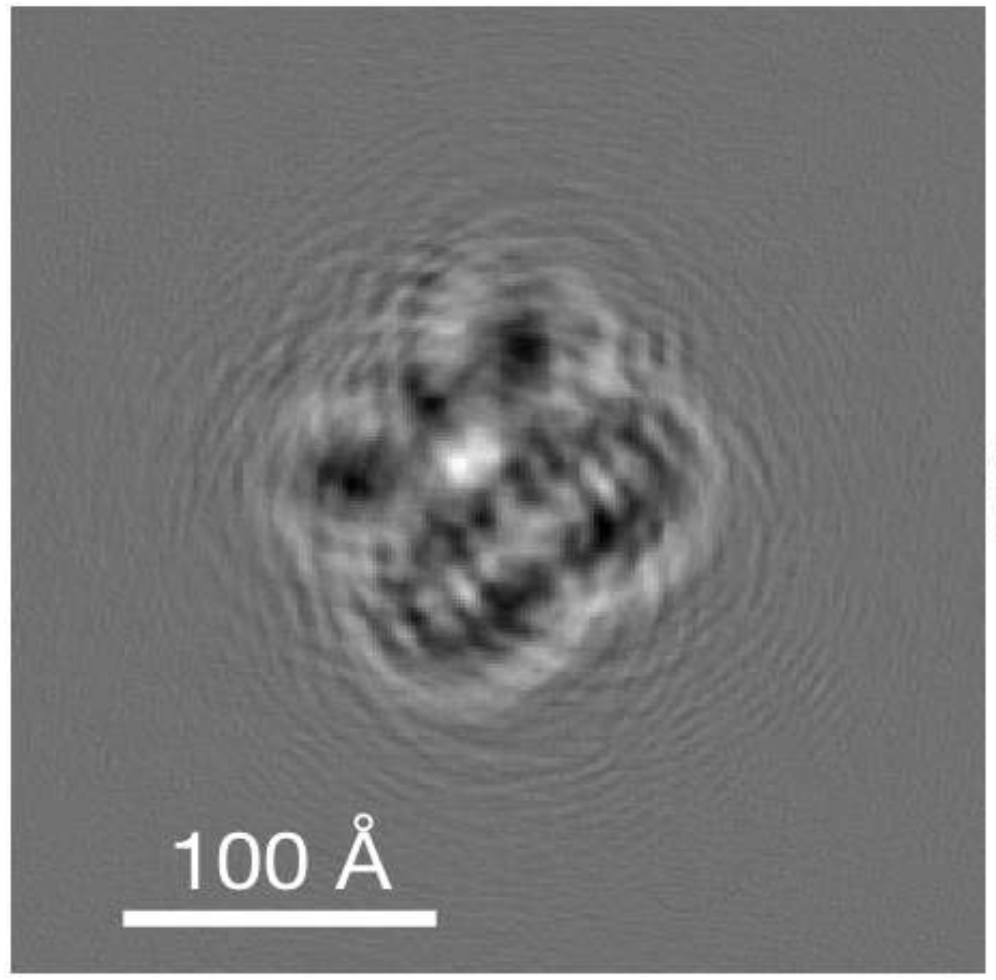
1/4 of a micrograph - empiar.org/10005

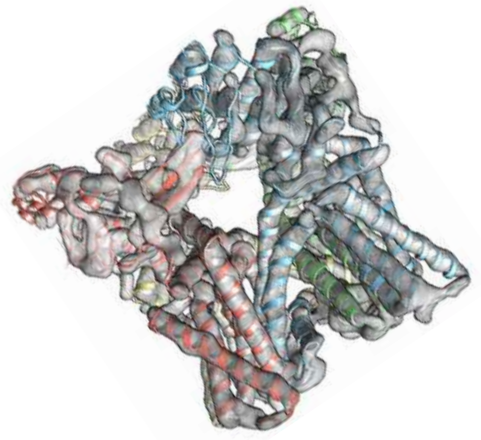


One particle image

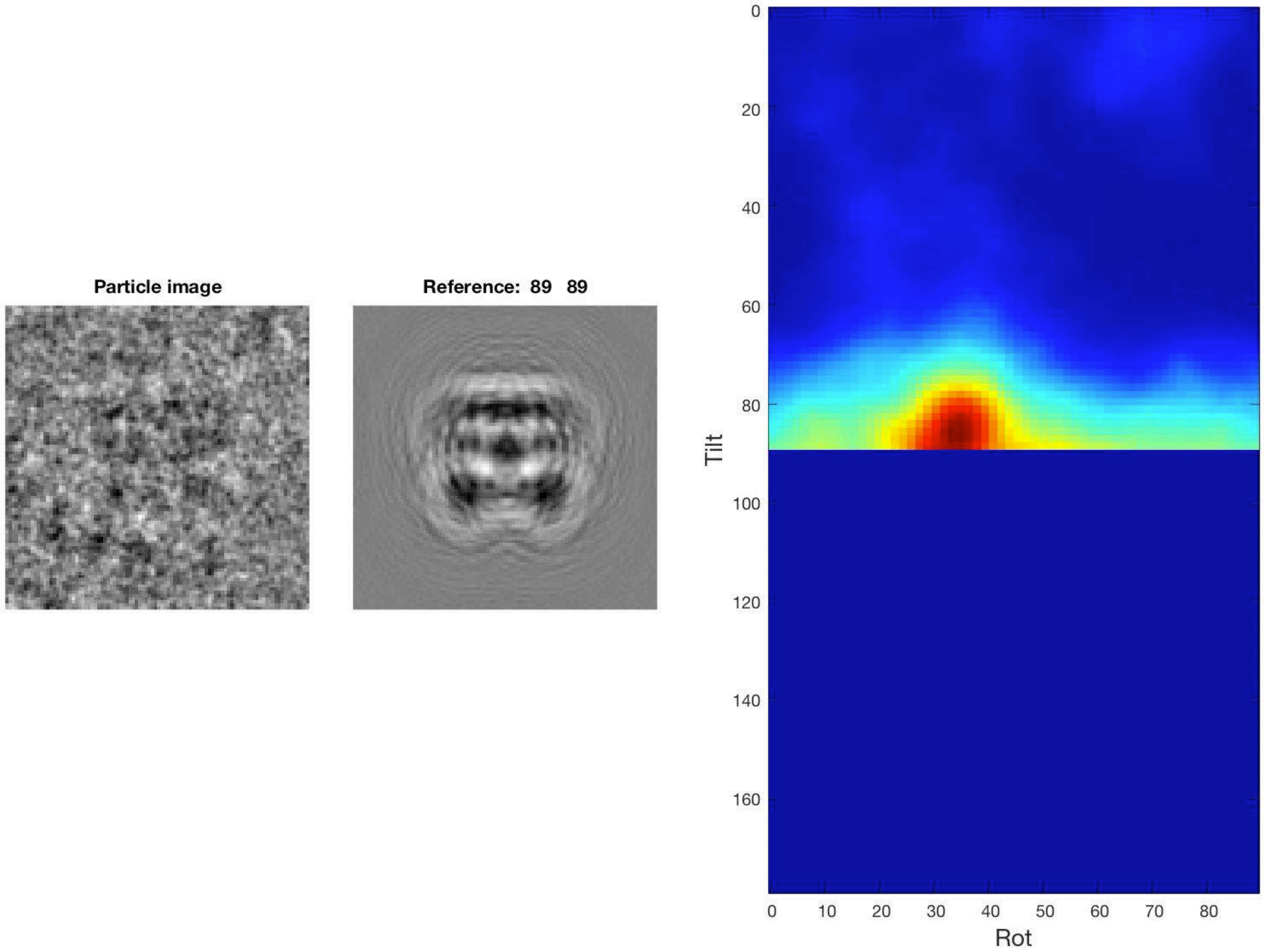








The probability of orientations $P(\phi | X, V)$ is remarkably sharp

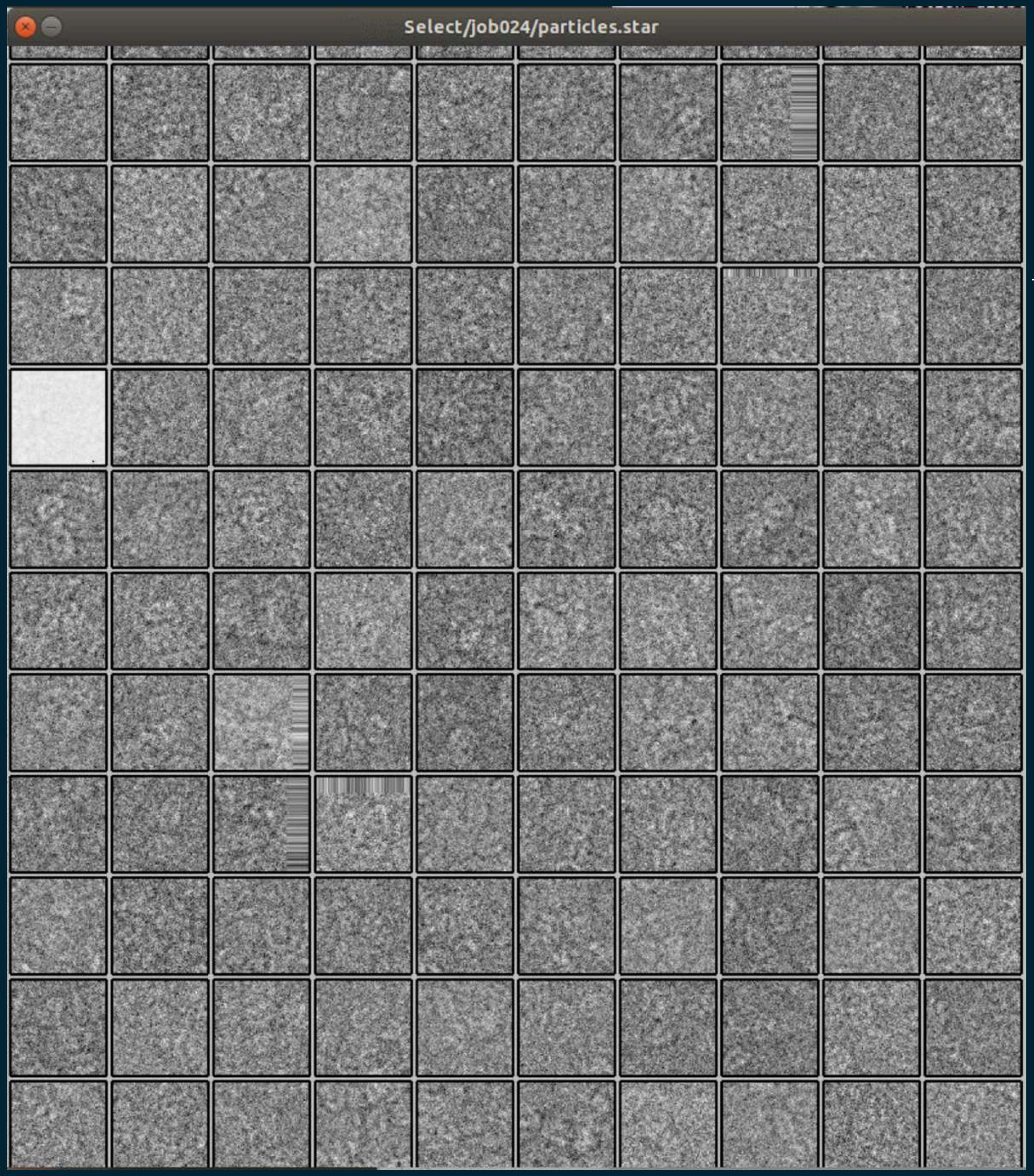


Single-particle reconstruction

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

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

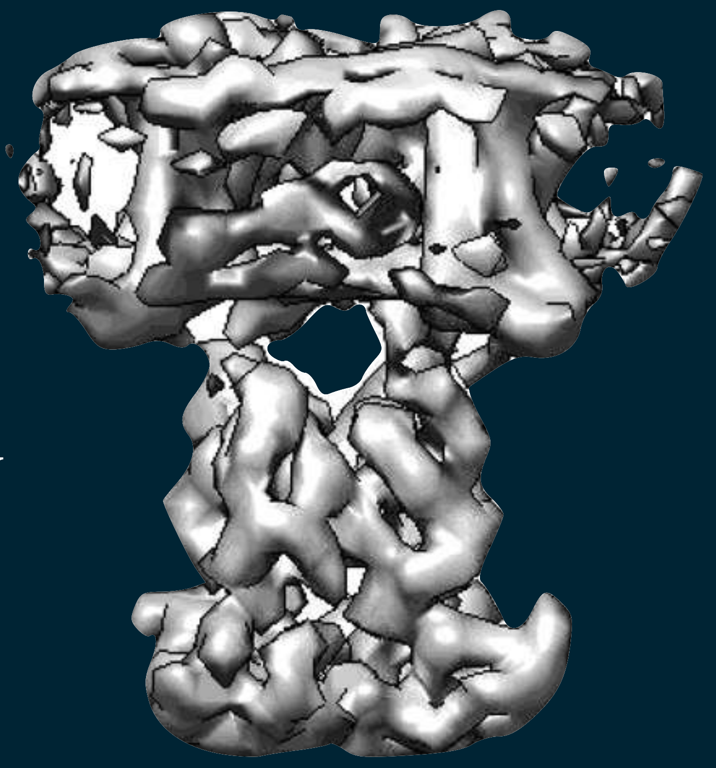
The goal is to discover the volume V



X_i

Project along ϕ_i

V



The first step is to compare images to determine orientations…

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

$$\|X - R\|^2 = \sum_j (X_j - R_j)^2$$
$$= \|X\|^2 - 2X \cdot R + \|R\|^2$$

Correlation

$$\text{Cor} = X \cdot R$$
$$= \sum_j X_j R_j$$

Correlation coefficient

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

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

First the 2D problem: reconstruct an image

Model of an image

$$X = CA + N$$

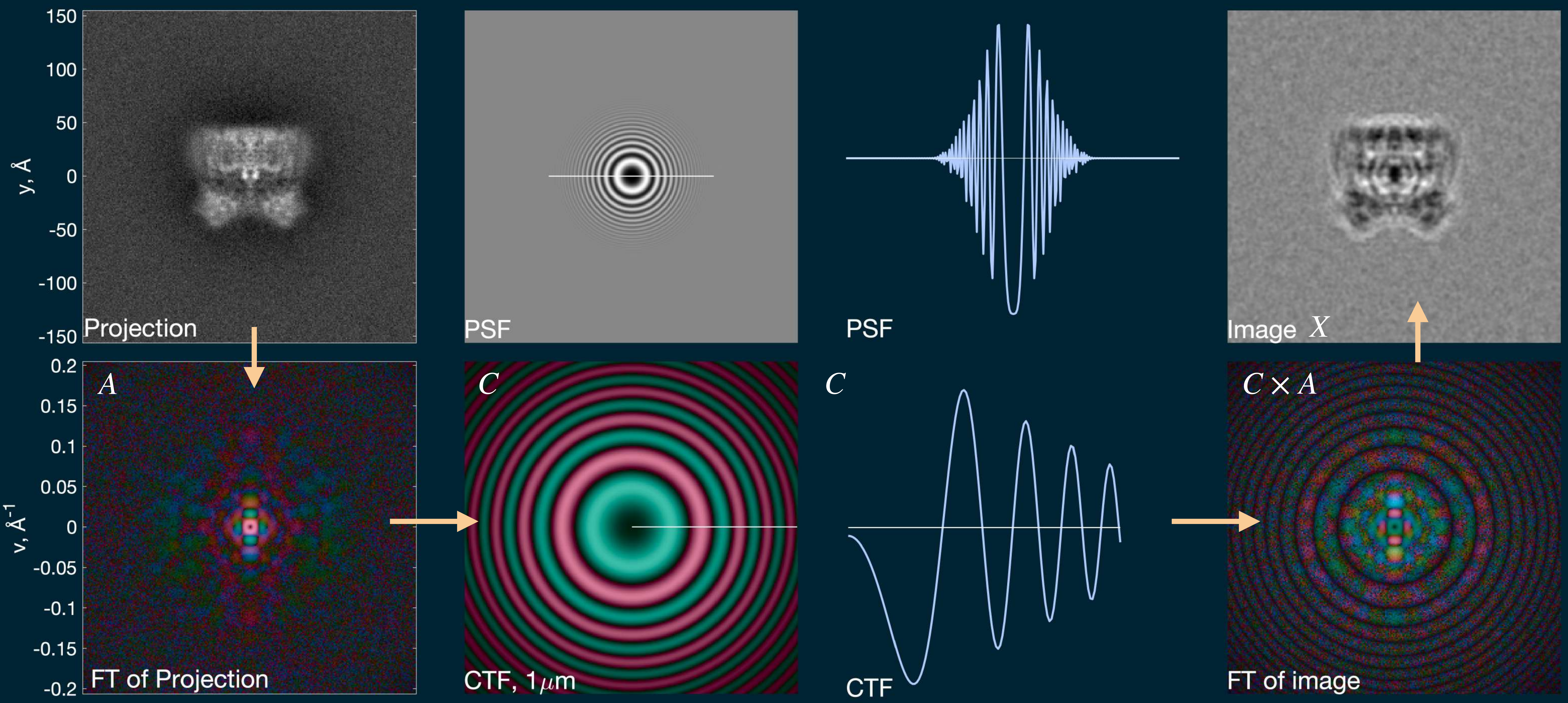
A	"true" image
C	contrast-transfer function
N	noise image

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

$X = CA + N$

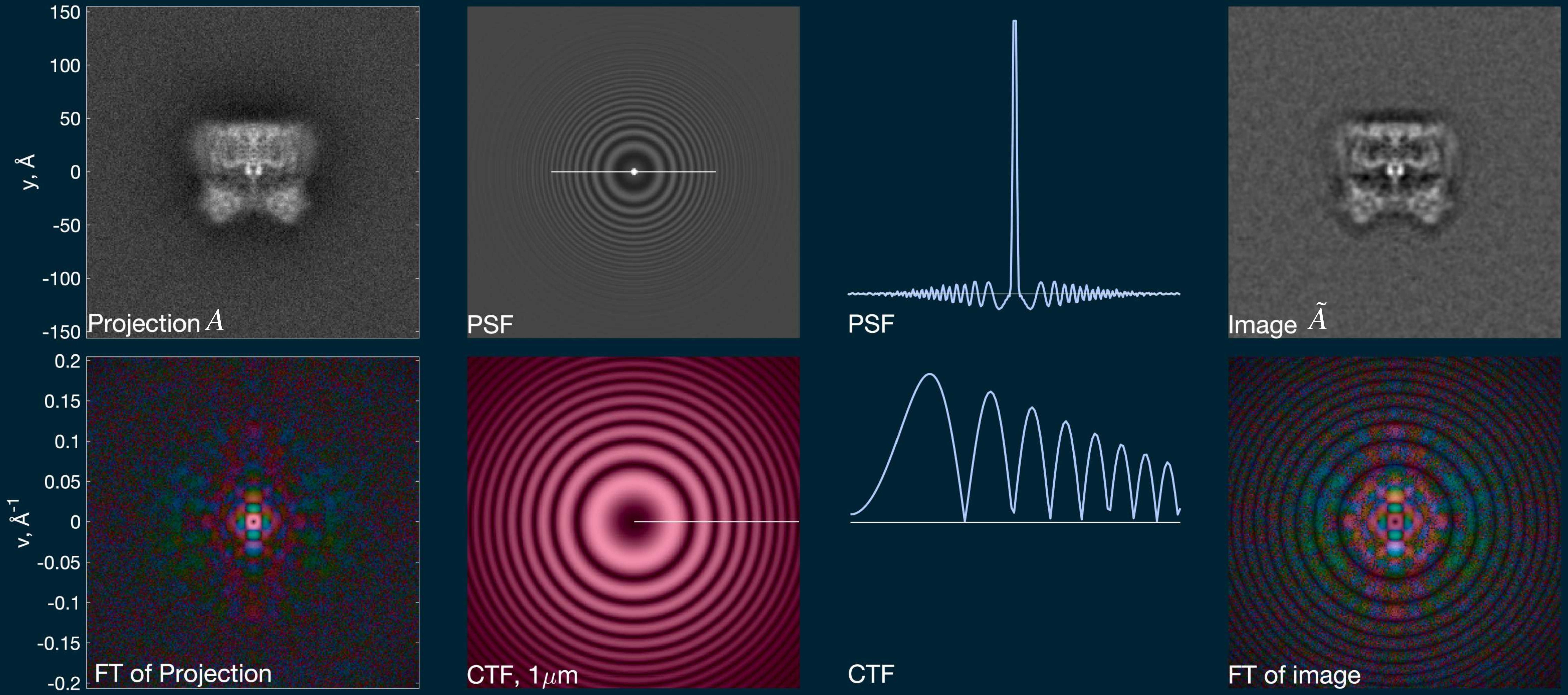
Can we do the deconvolution: $\tilde{A} = X/C$??



How to undo the CTF effects?

1. **Phase flipping**

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



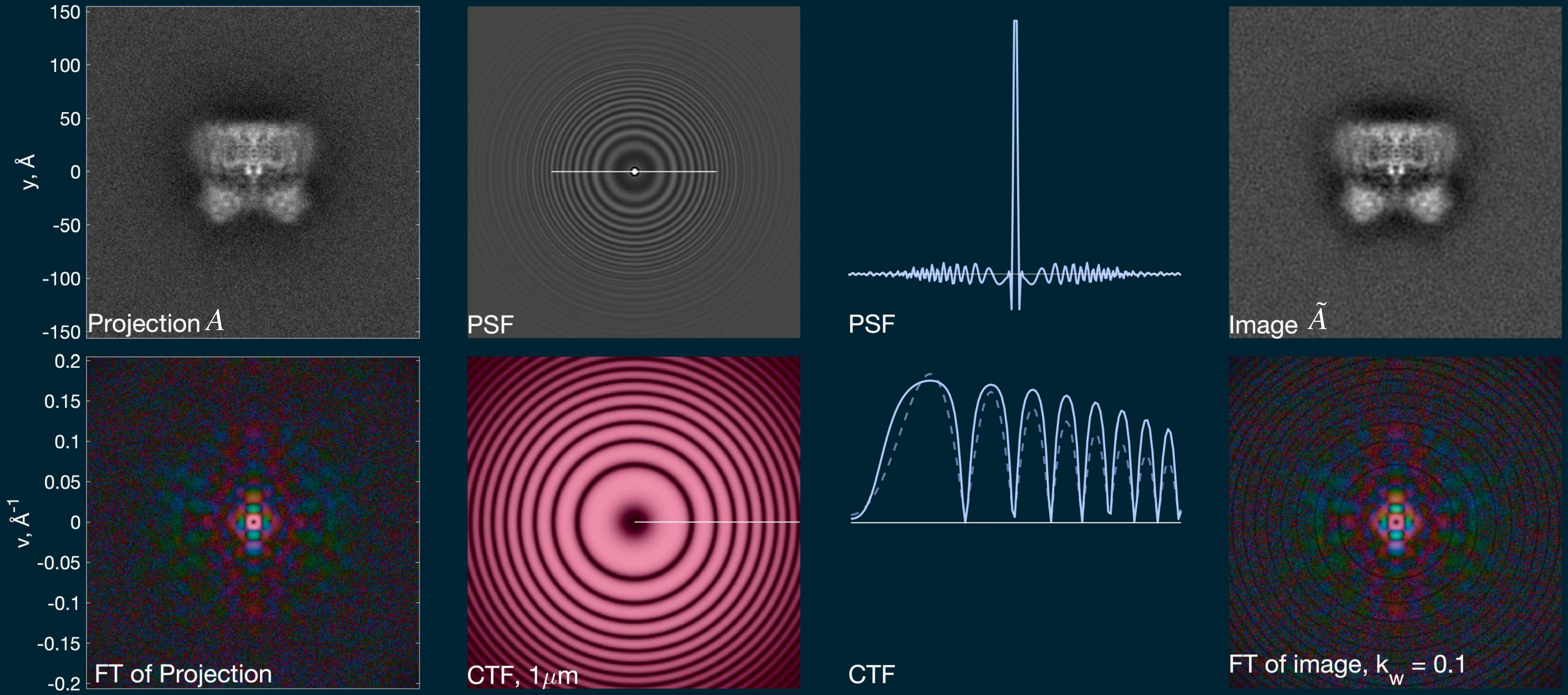
How to undo the CTF effects?

1. **Phase flipping**

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

2. **Wiener filter**

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



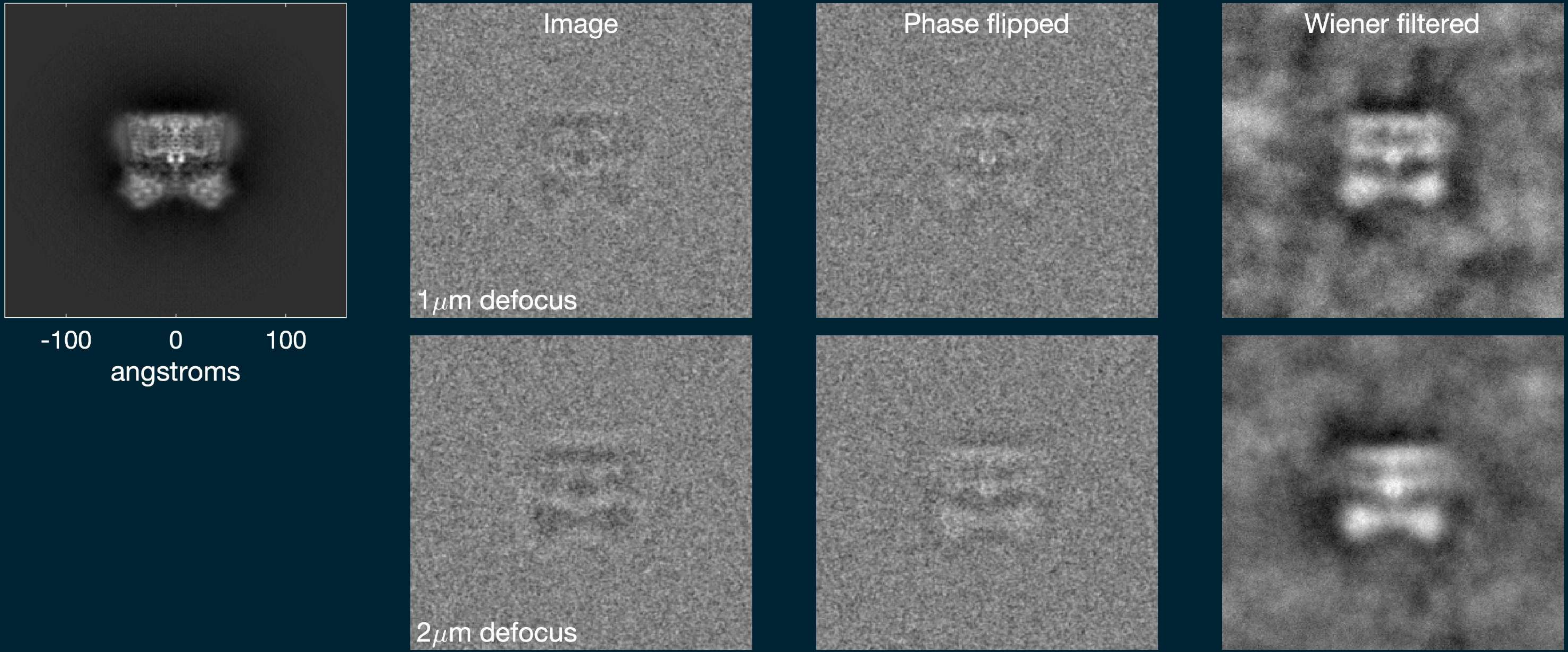
How to undo the CTF effects in noisy images?

1. Phase flipping

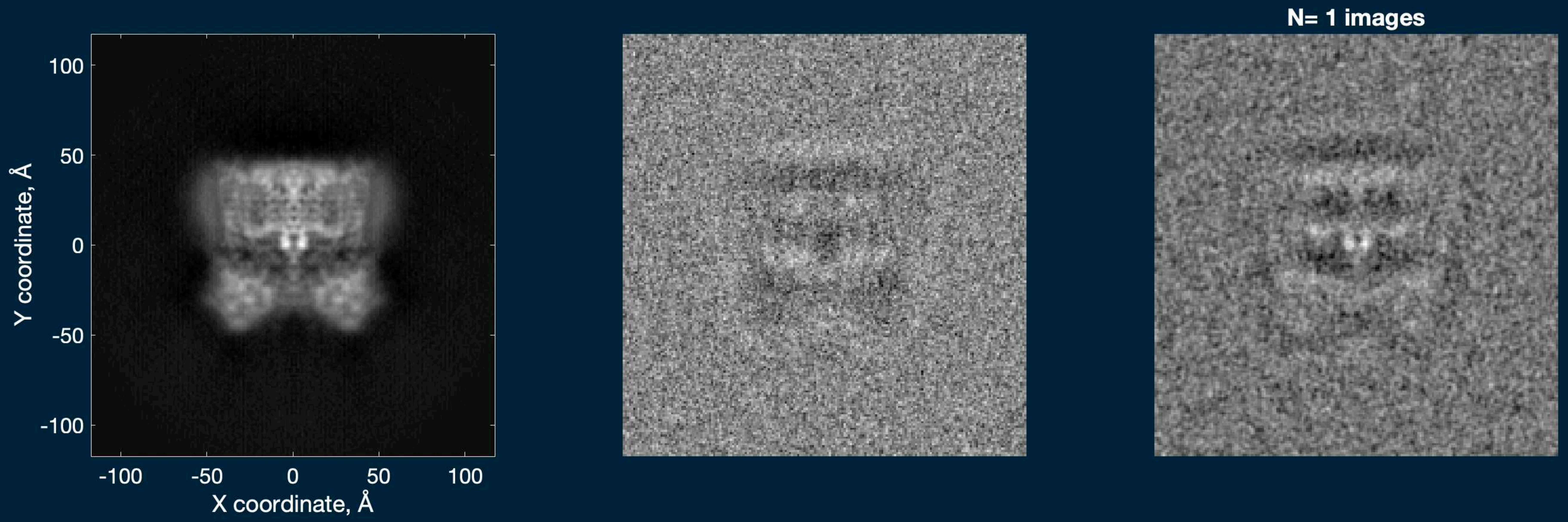
$$\tilde{A} = \mathrm{sgn}(C)X$$

2. Wiener filter

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



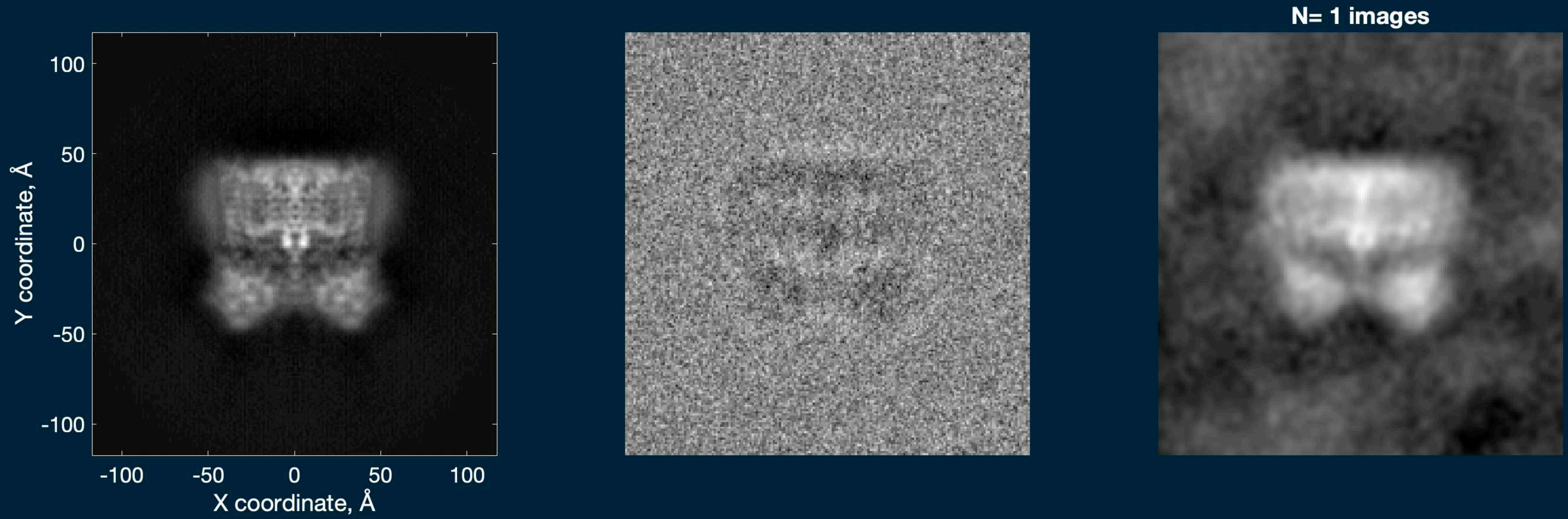
How to undo the CTF effects in noisy images?



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

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

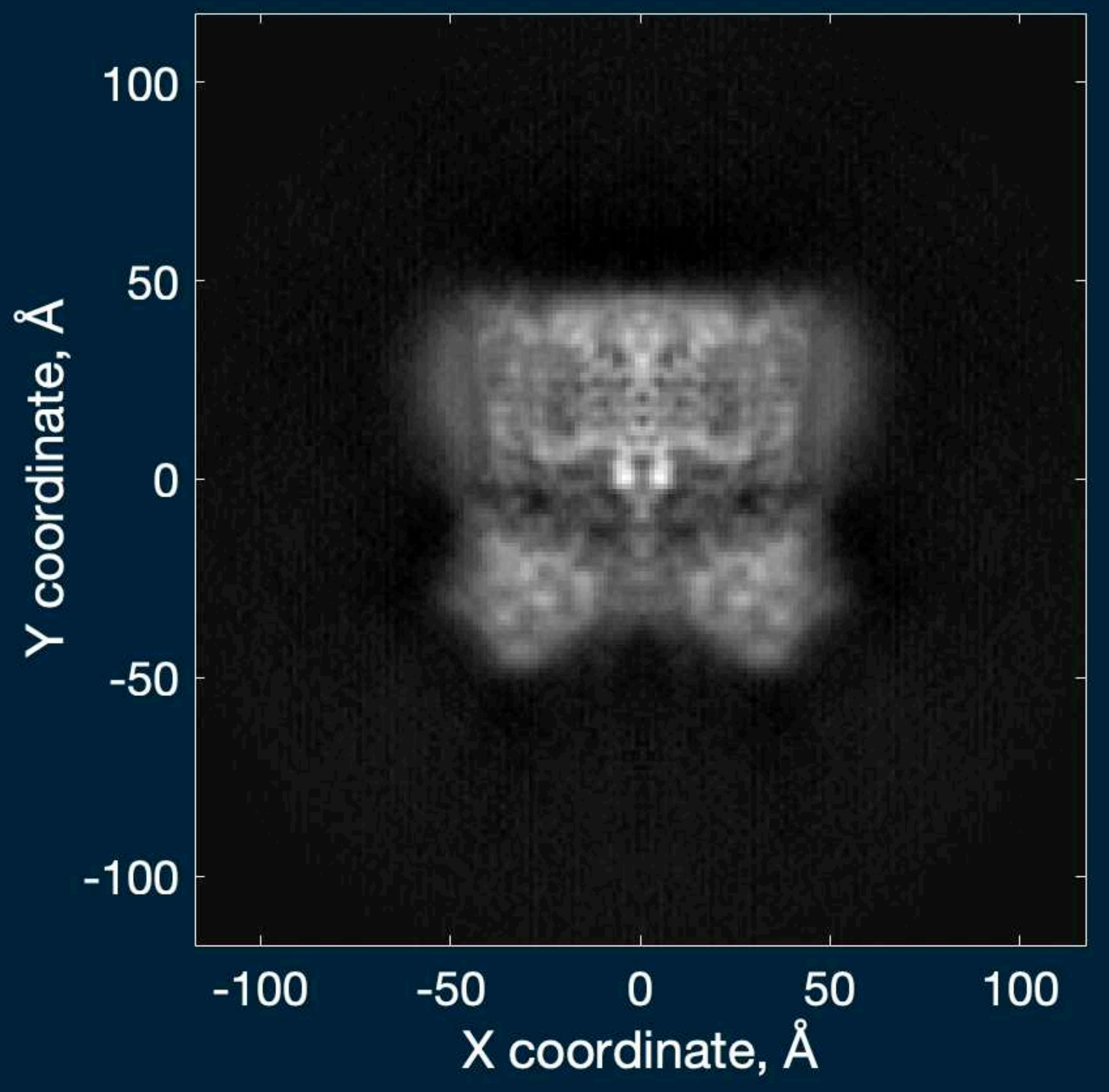
$$k(s) = 1/\mathrm{SNR} = \frac{|N|^2}{|A|^2}$$

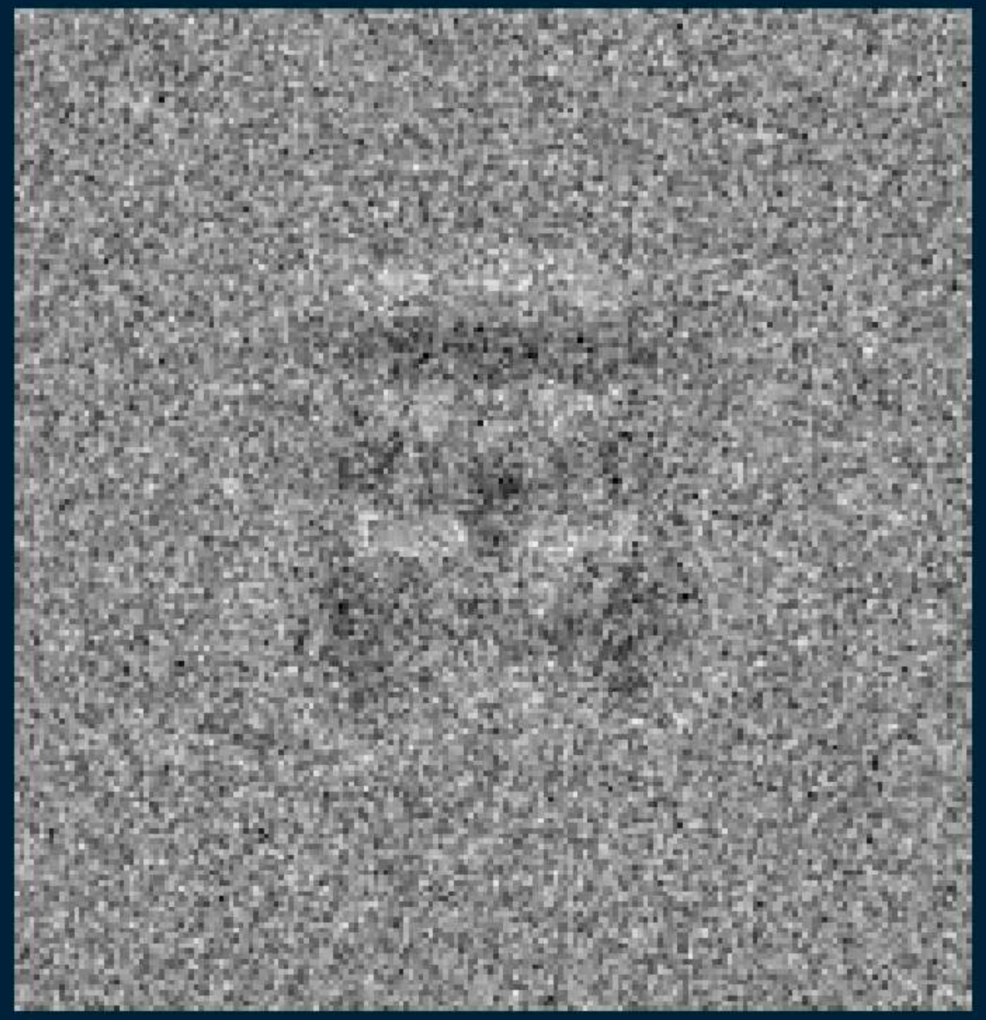
Image restoration when spectral SNR is known

Restoration from multiple images

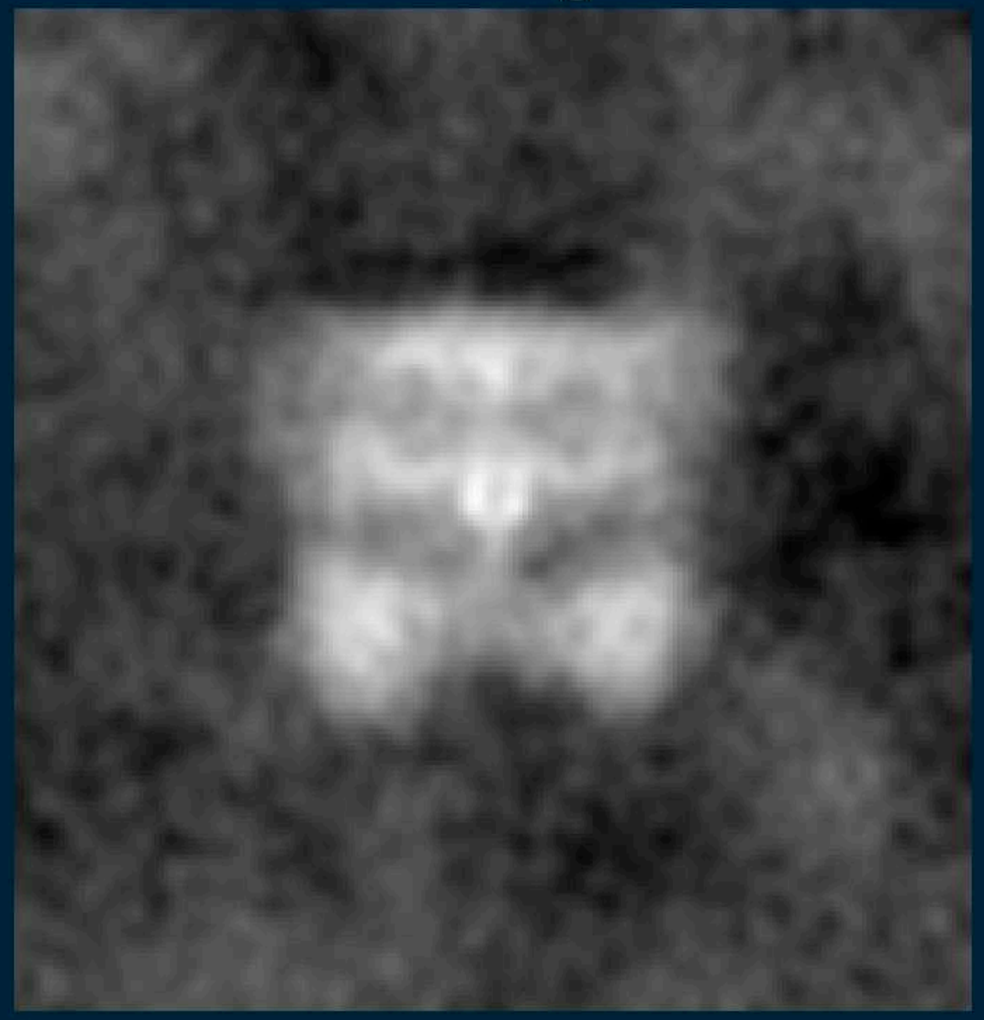
$$\tilde{A} = \frac{\sum_i^N C_i X_i}{\frac{1}{\text{SSNR}} + \sum_i^N C_i^2}$$

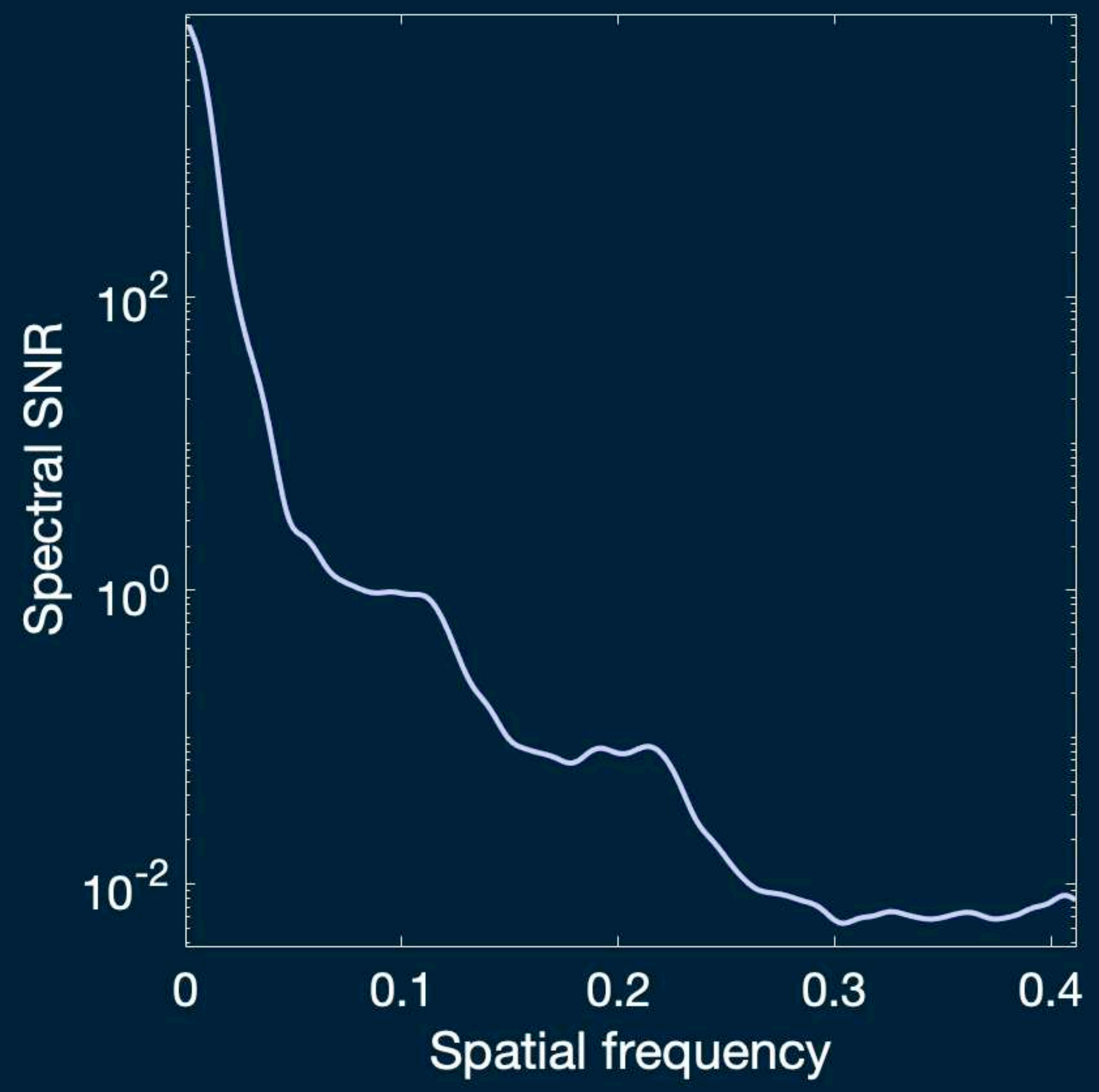
The defocus varies to fill in CTF zeros





N= 1 images





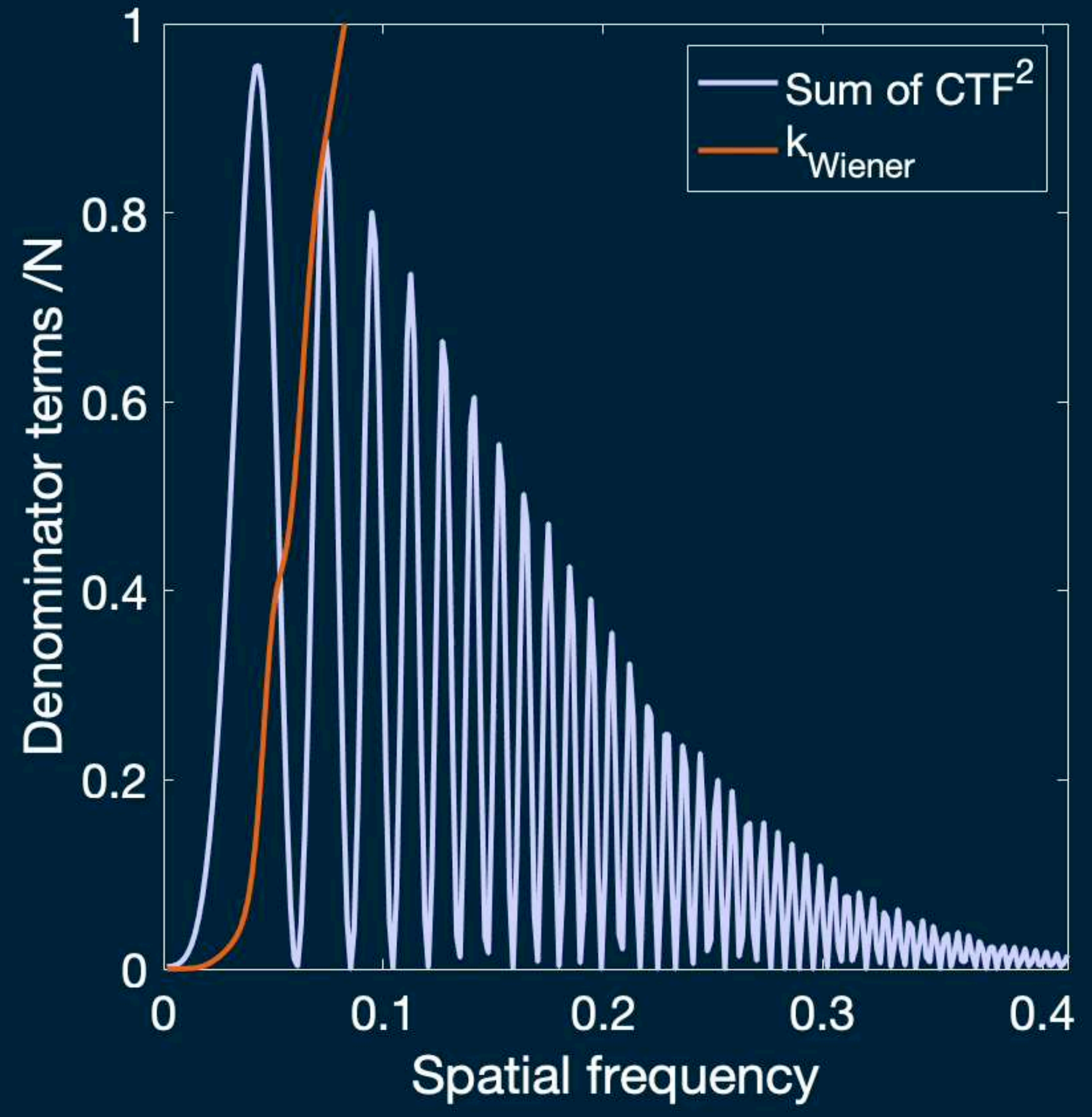
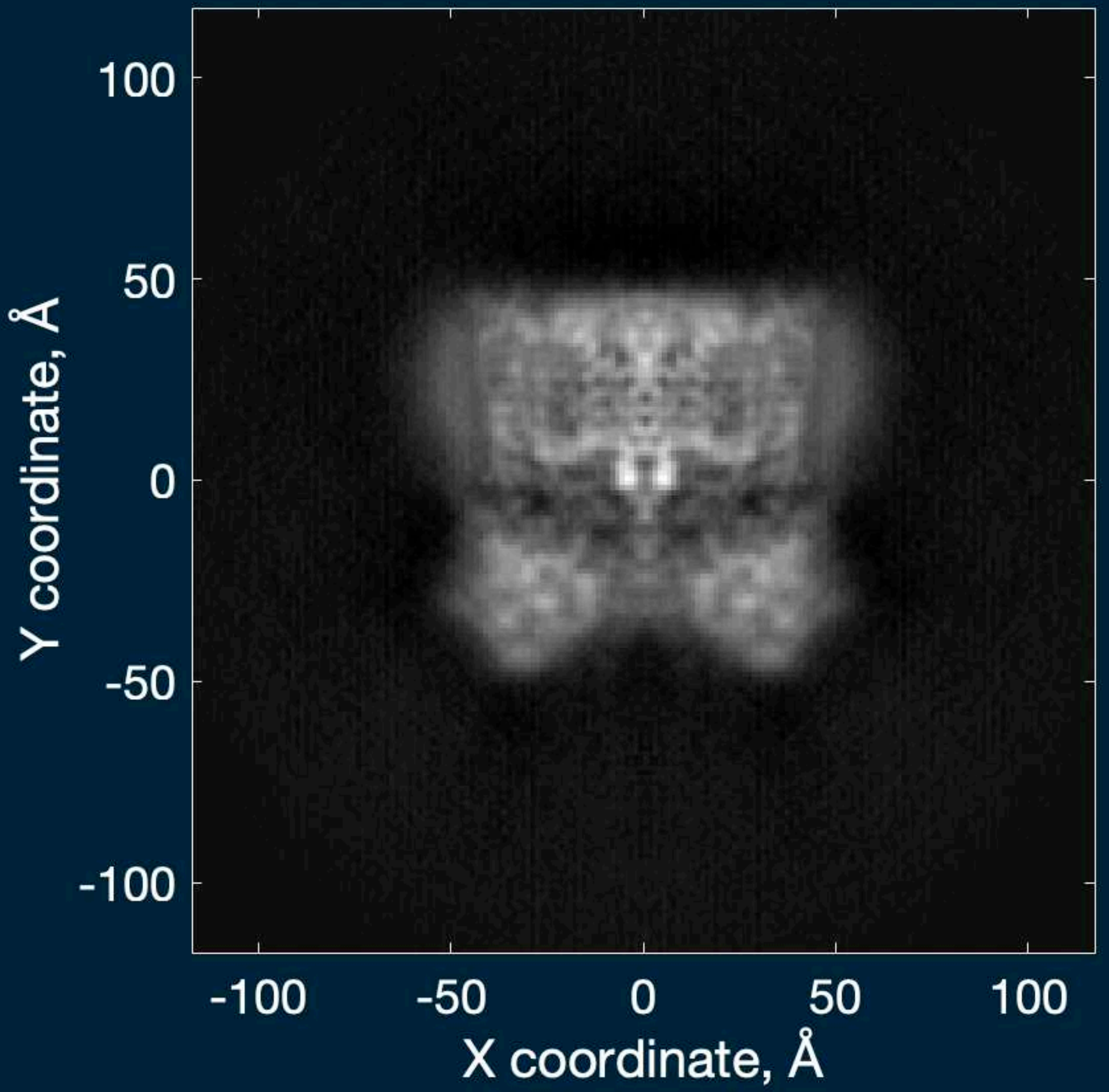


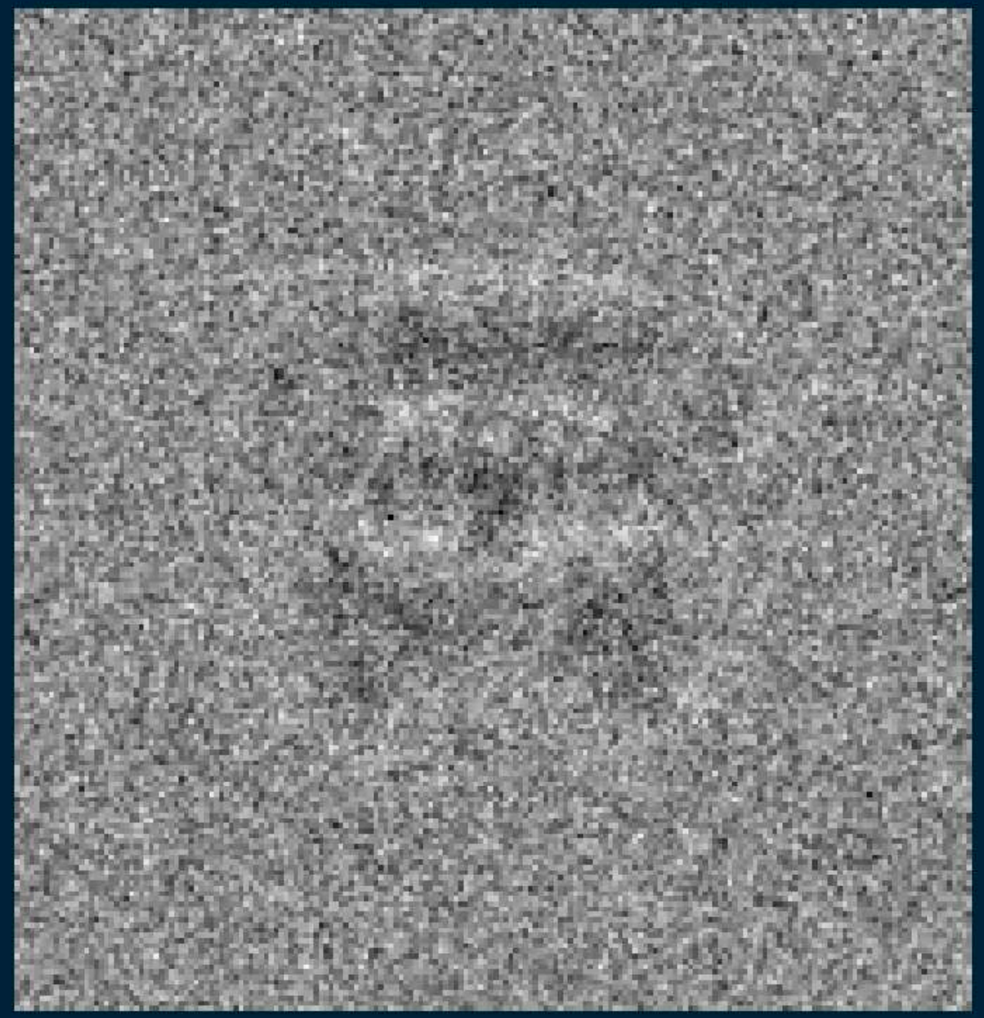
Image restoration when spectral SNR is known

Restoration from multiple images

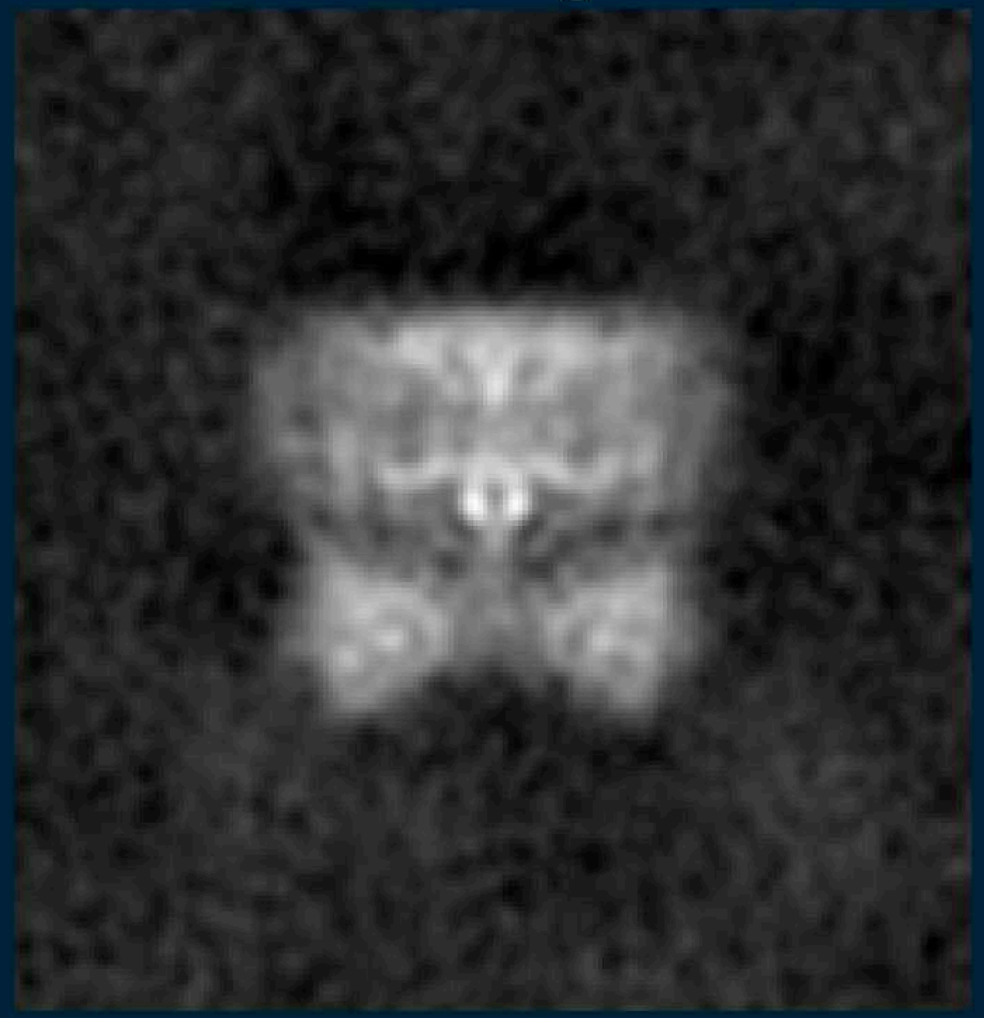
$$\tilde{A} = \frac{\sum_i^N C_i X_i}{\frac{1}{\mathrm{SSNR}} + \sum_i^N C_i^2}$$

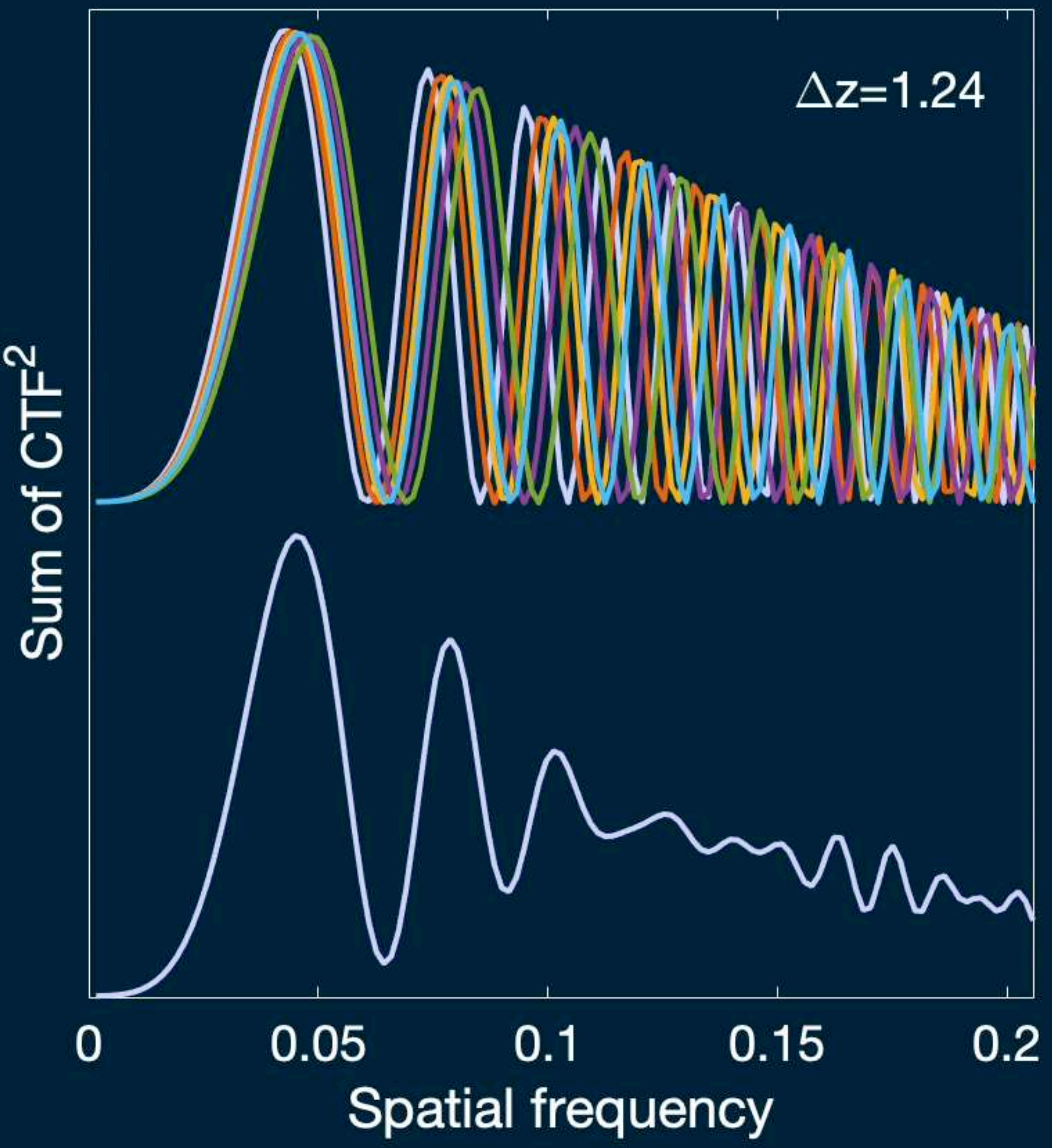
The defocus varies to fill in CTF zeros





N= 6 images





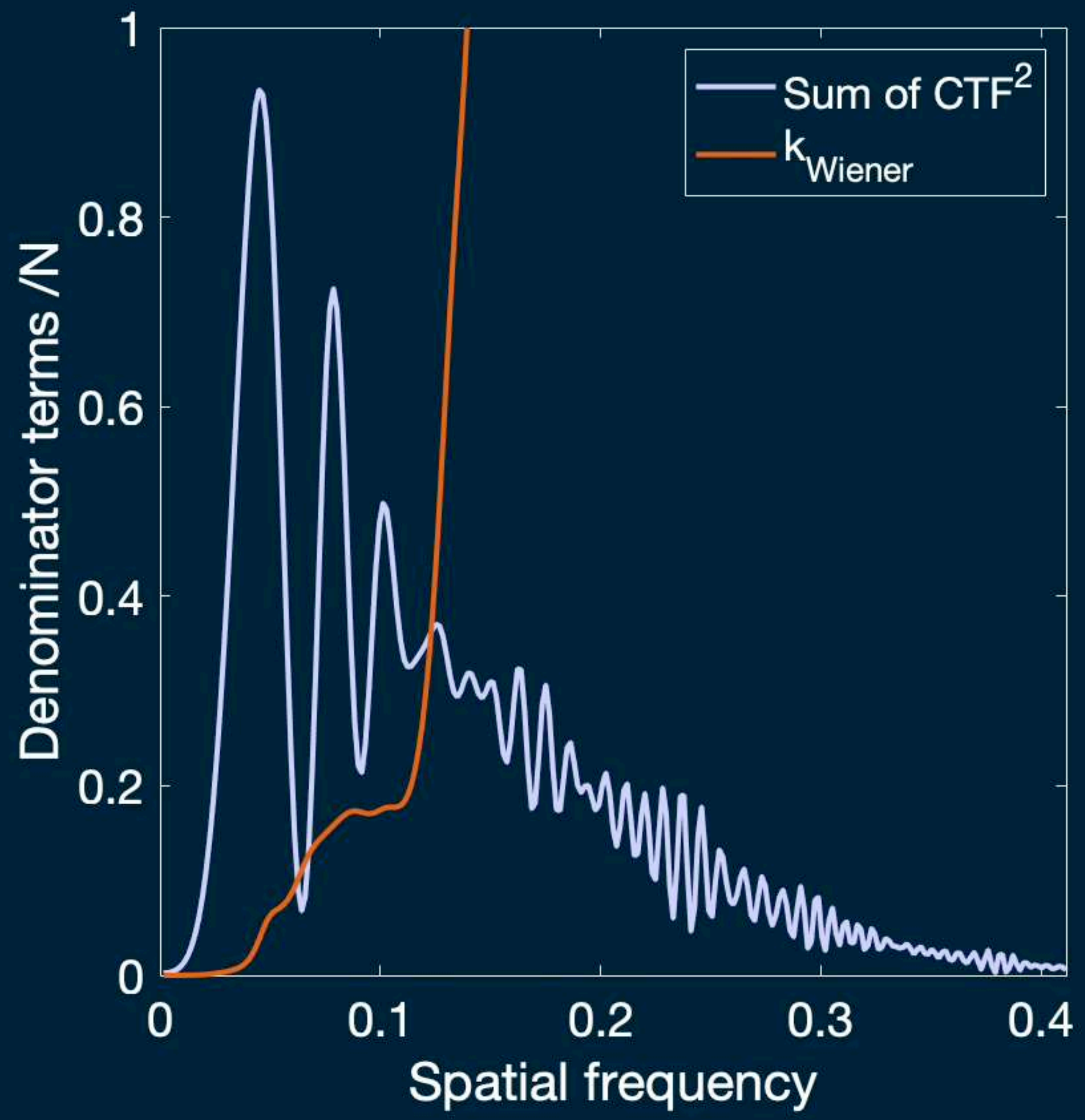
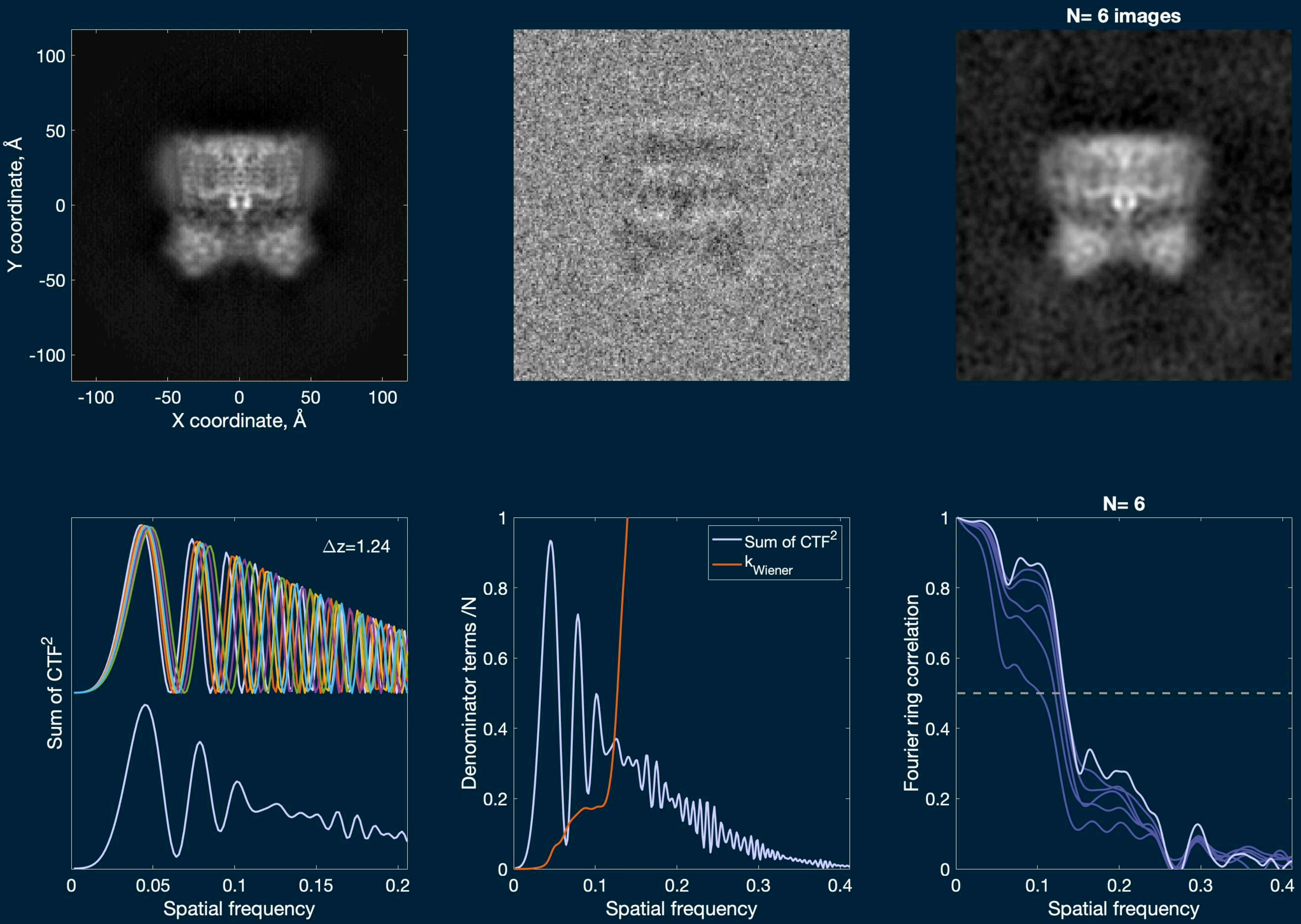


Image restoration when spectral SNR is known

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



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

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

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. $\mathbf{P}^{\mathbf{T}}_{\phi_i}$ 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

$$\sum_i^N \mathbf{P}^{\mathbf{T}}_{\phi_i} C_i X_i$$

is therefore the insertion of *N* slices.

3D reconstruction in FREALIGN—iterations

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

Iterate

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 $V^{(n+1)}$

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

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

Correlation and particle picking

Single-particle reconstruction

Maximum-likelihood methods

Probabilities, another way to compare images

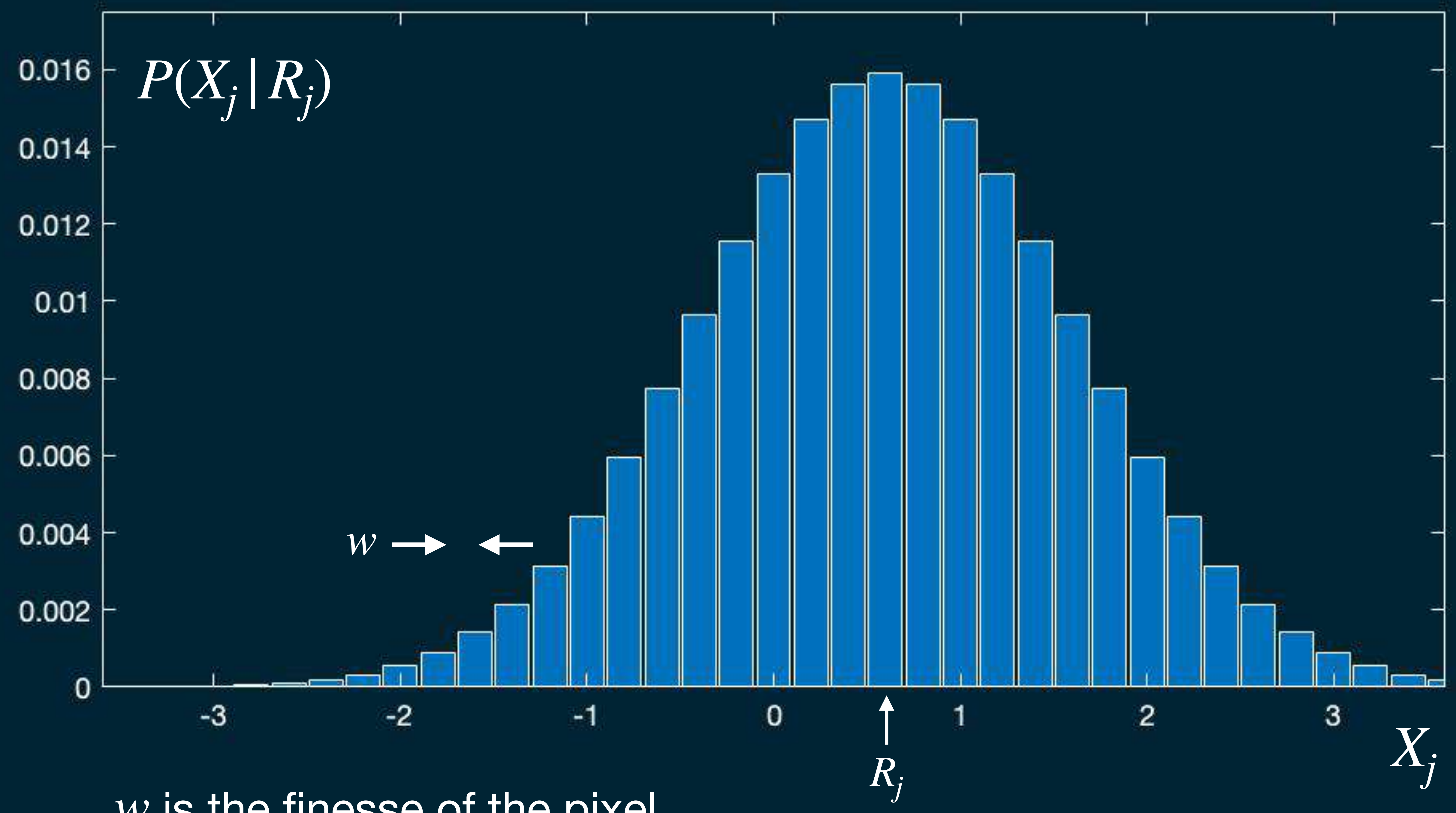
Image model: $X = R + N$

Probability of the j[th] pixel value:

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

Probability of observing an entire image that comes from R:

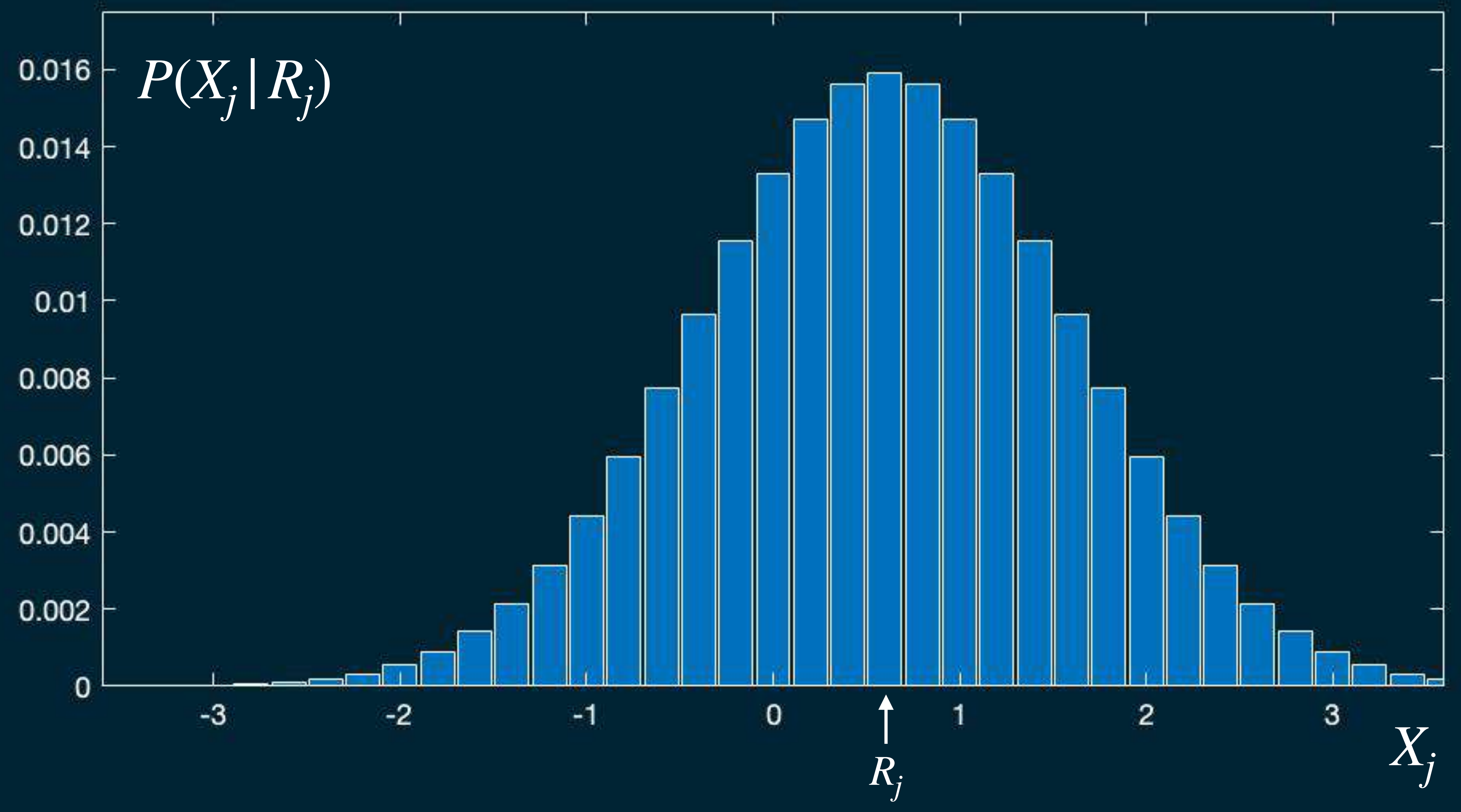
$$P(X \mid R) = \frac{\cancel{w^J}\ 1}{(2\pi\sigma^2)^{J/2}}\, e^{-||X-R||^2/2\sigma^2}$$



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

Simplified image probability

$$X = R + N$$



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

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

(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 V consistent with these data, say maximizing the posterior probability

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

According to Bayes' theorem,

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

- $P(\mathbf{X})$ 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.

- $P(\mathbf{X} \mid V)$ is something we can calculate. It's called the likelihood of V.

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

We know how to compute the likelihood

We know that

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

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

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

assuming $P(\phi)$ is uniform.

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

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

For numerical sanity, we compute the log likelihood,

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

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:

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

Probability of a projection direction for X_i:

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

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

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

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

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:

$$\Gamma^{(n)}_{\phi_i,k}$$

The iteration, guaranteed to increase the likelihood:

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

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

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

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

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

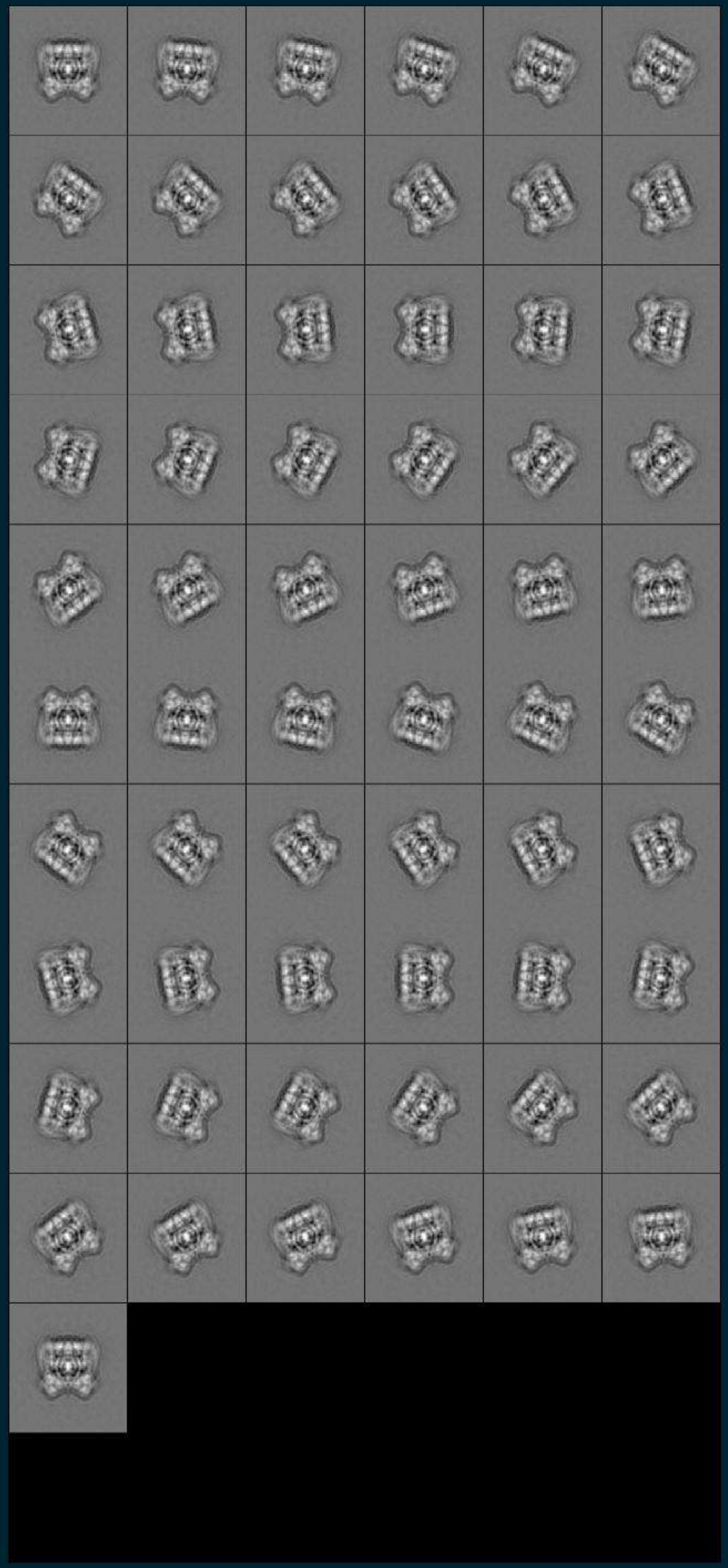
No. operations $\approx \underbrace{\frac{\pi^3}{8} t^2 n^5 N}_{\text{finding orientations}} + \underbrace{\pi n^4 + N n^2}_{\text{3D reconstruction}}$

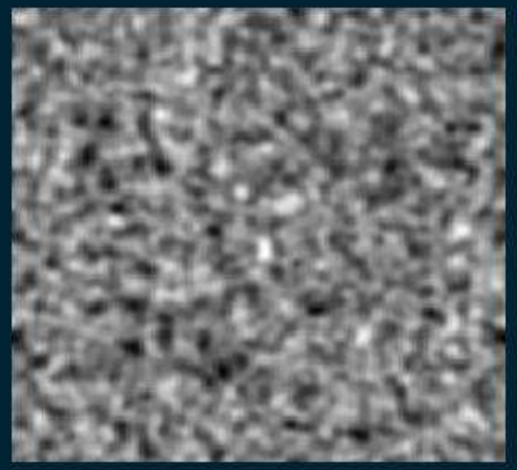
e.g. $N=10^5$, $n=128$, $t=7$

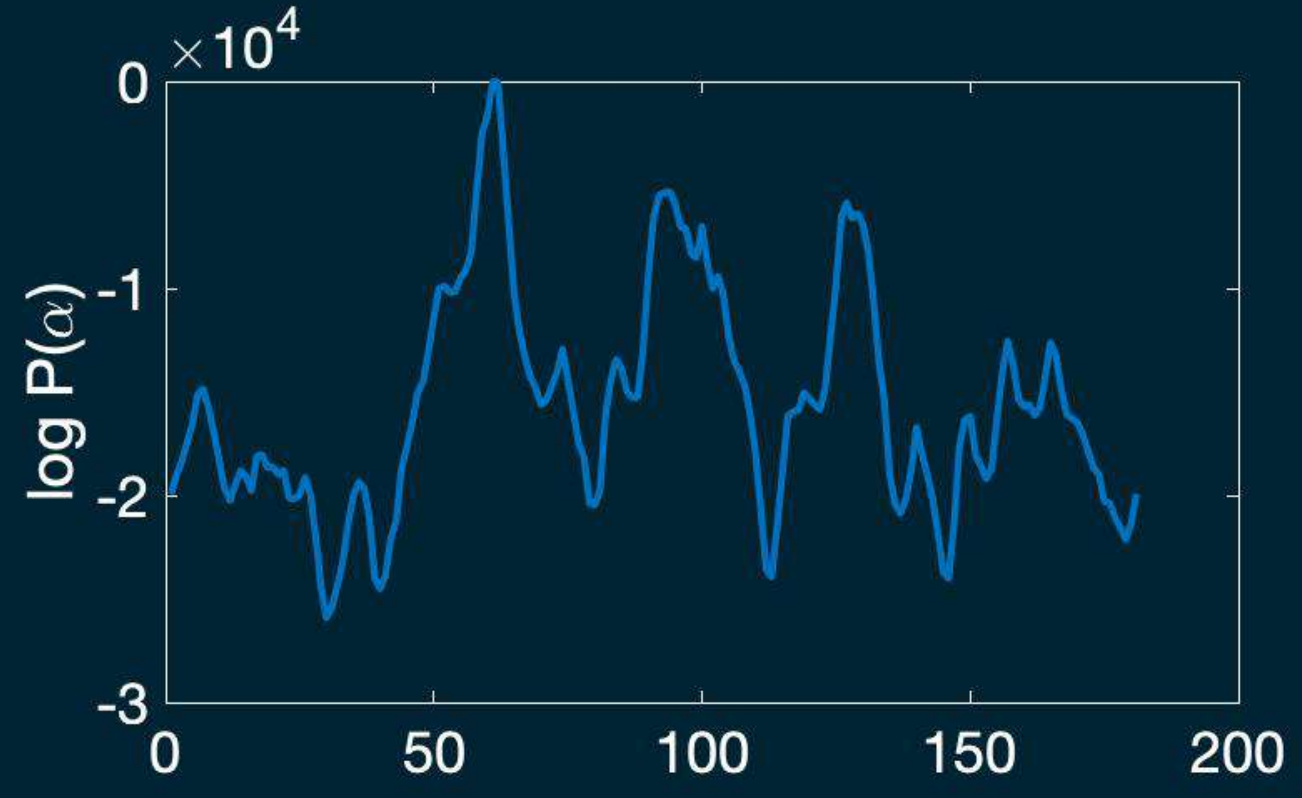
No. operations ≈ 6 x 10^{17} ≈ 19 CPU-years

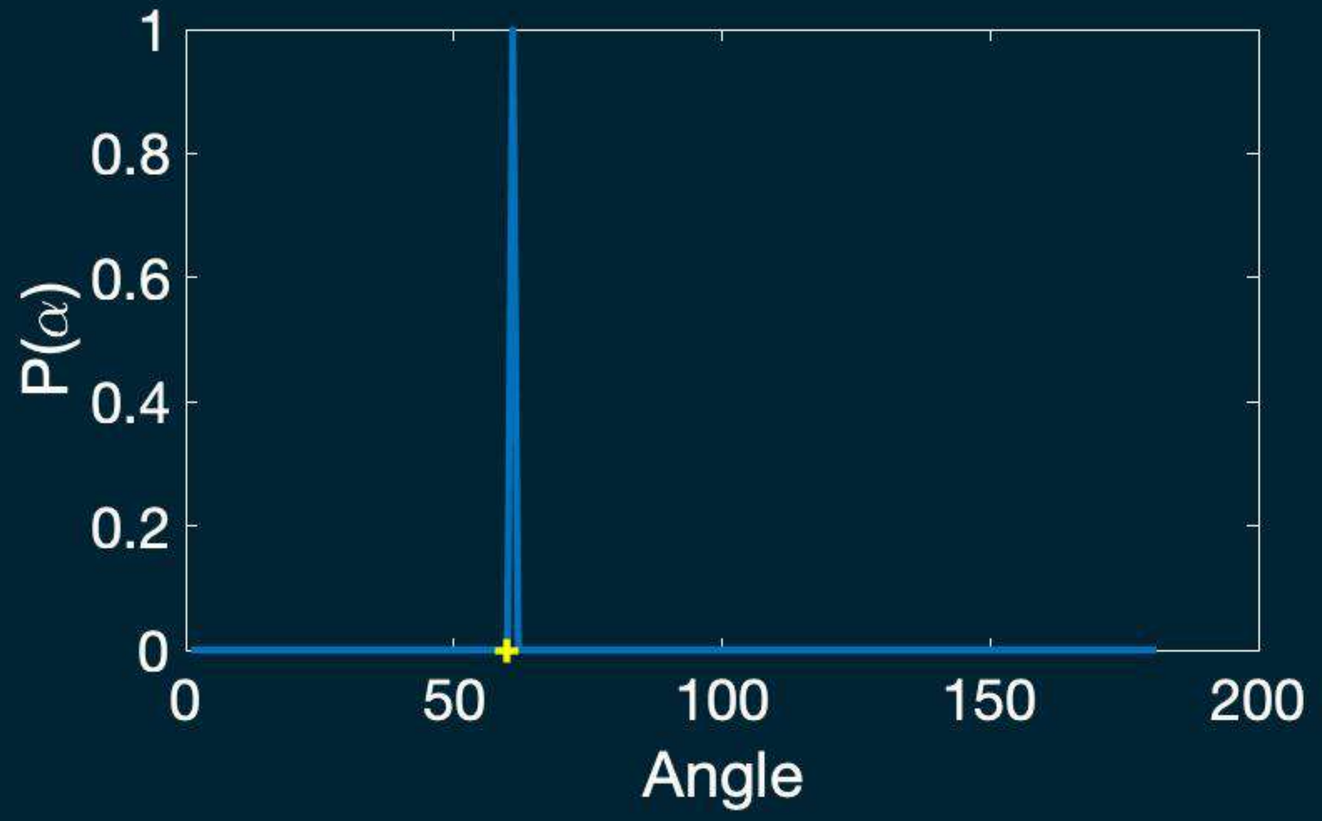
With efficient programs, ~ 1 CPU-day

Evaluating Γ_ψ is expensive: one of 5 parameters

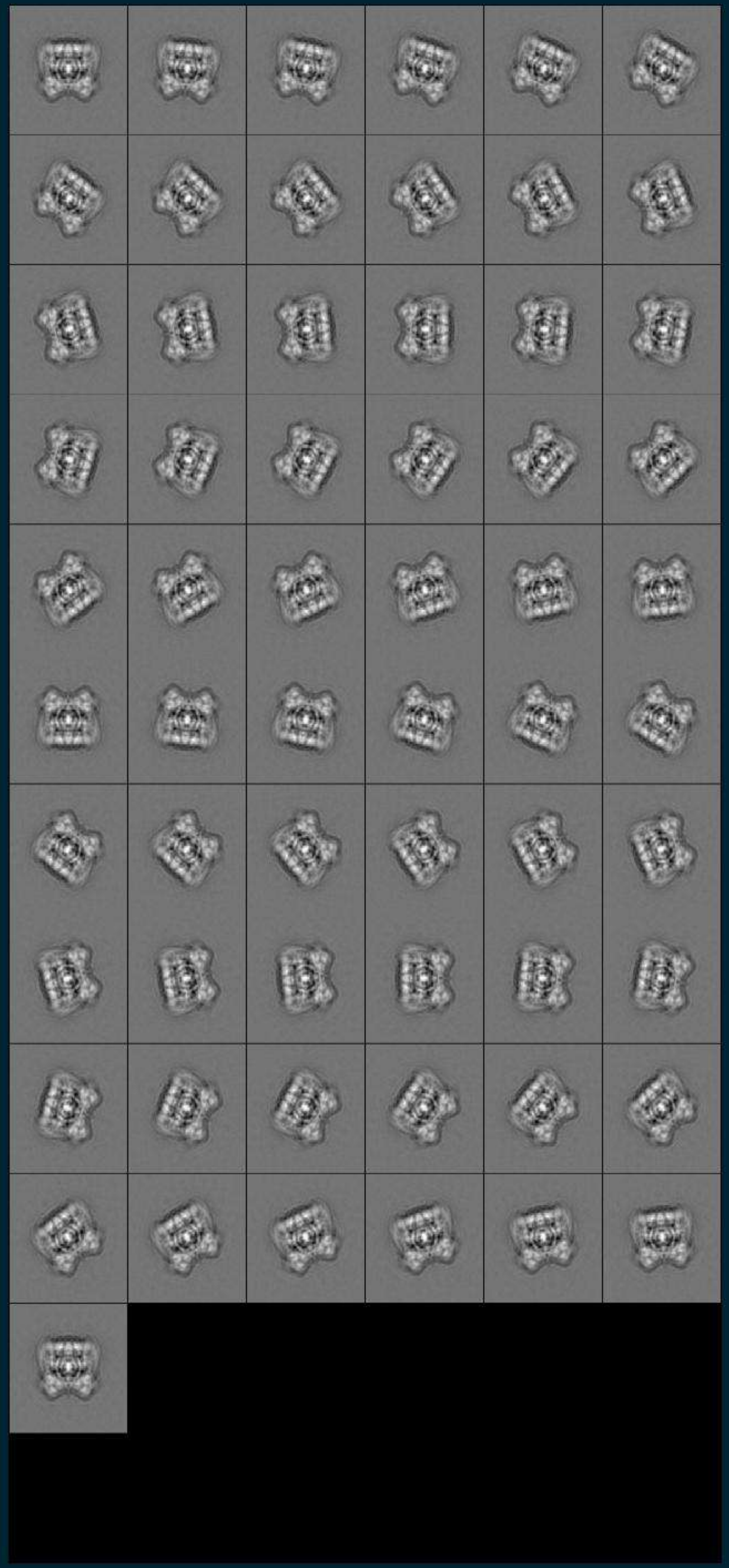


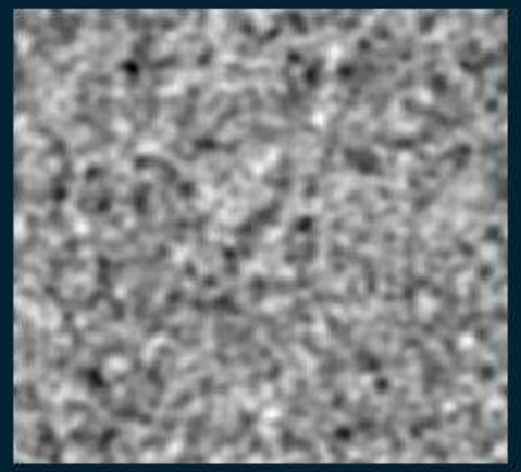


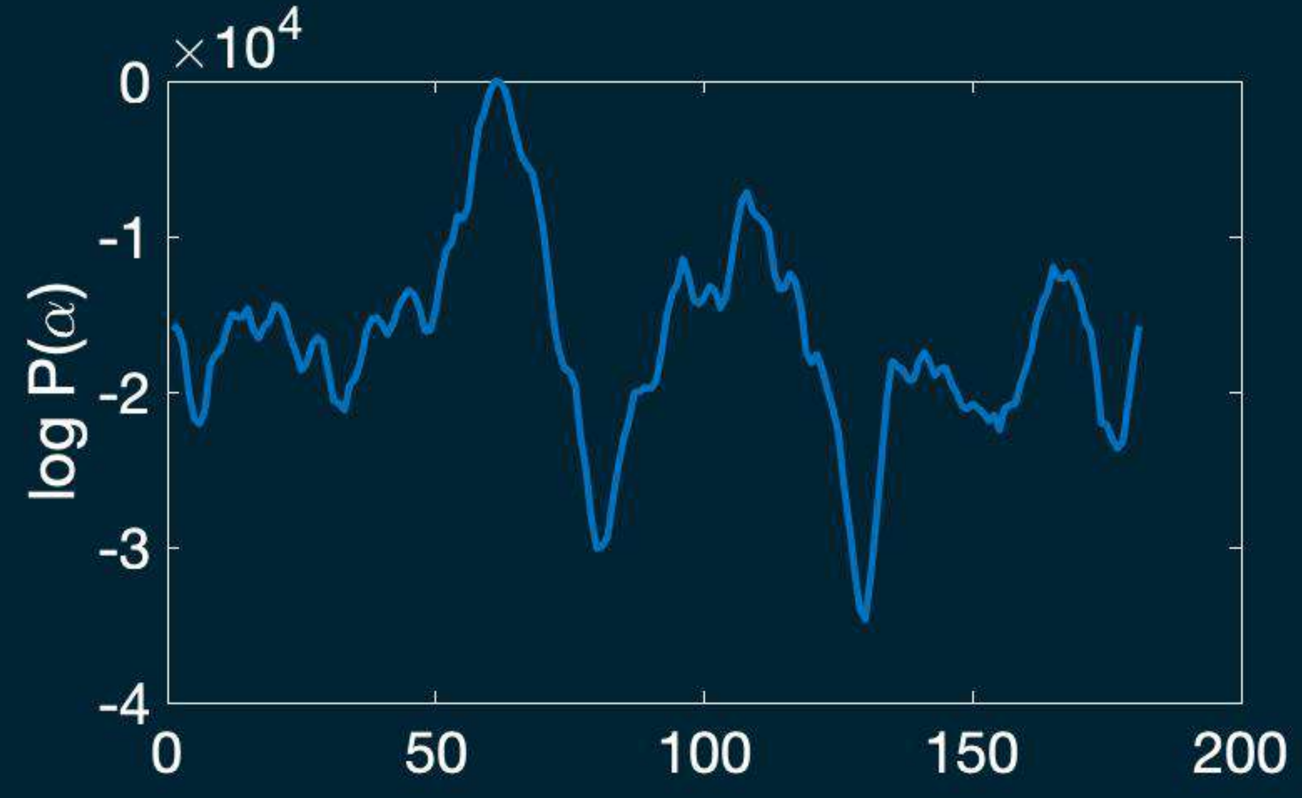


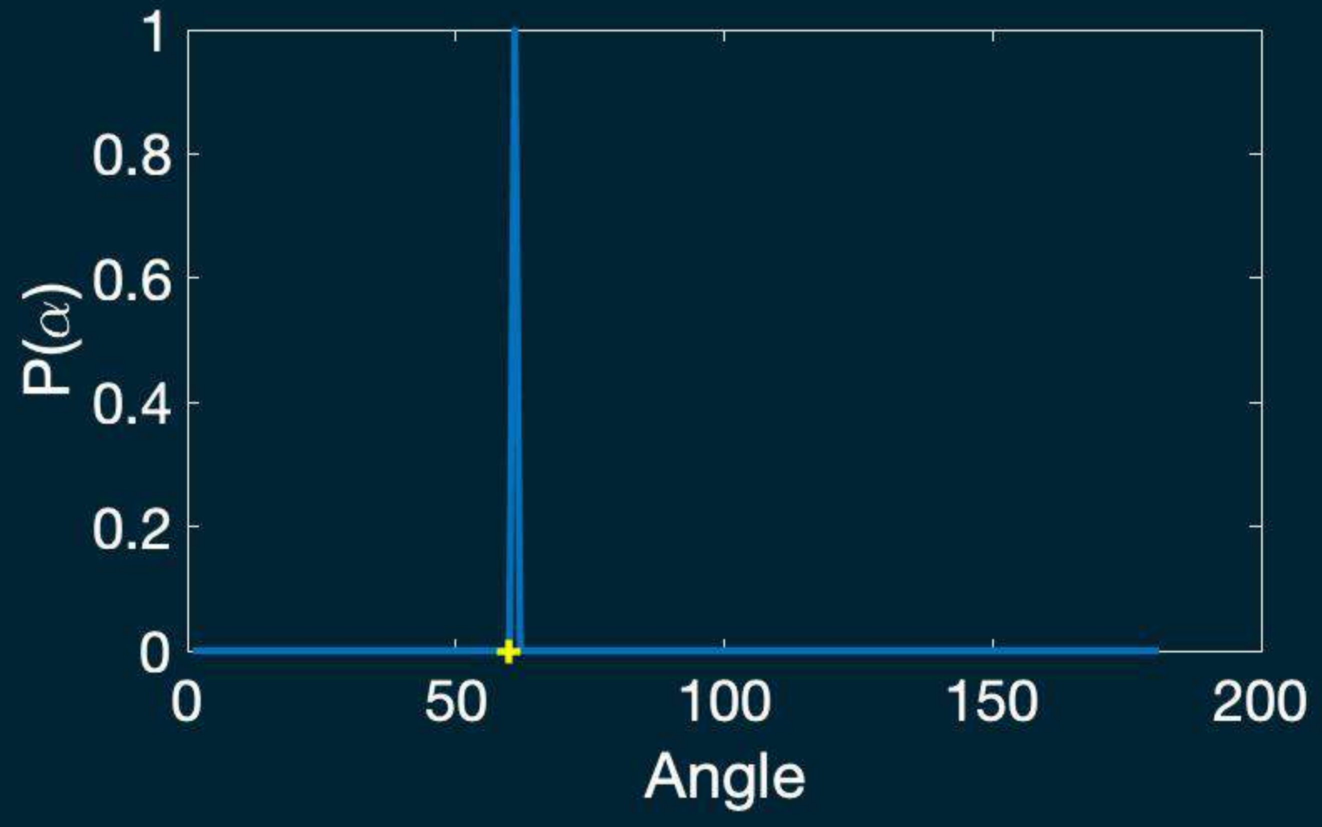


Evaluating Γ_ϕ is expensive: one of 5 parameters

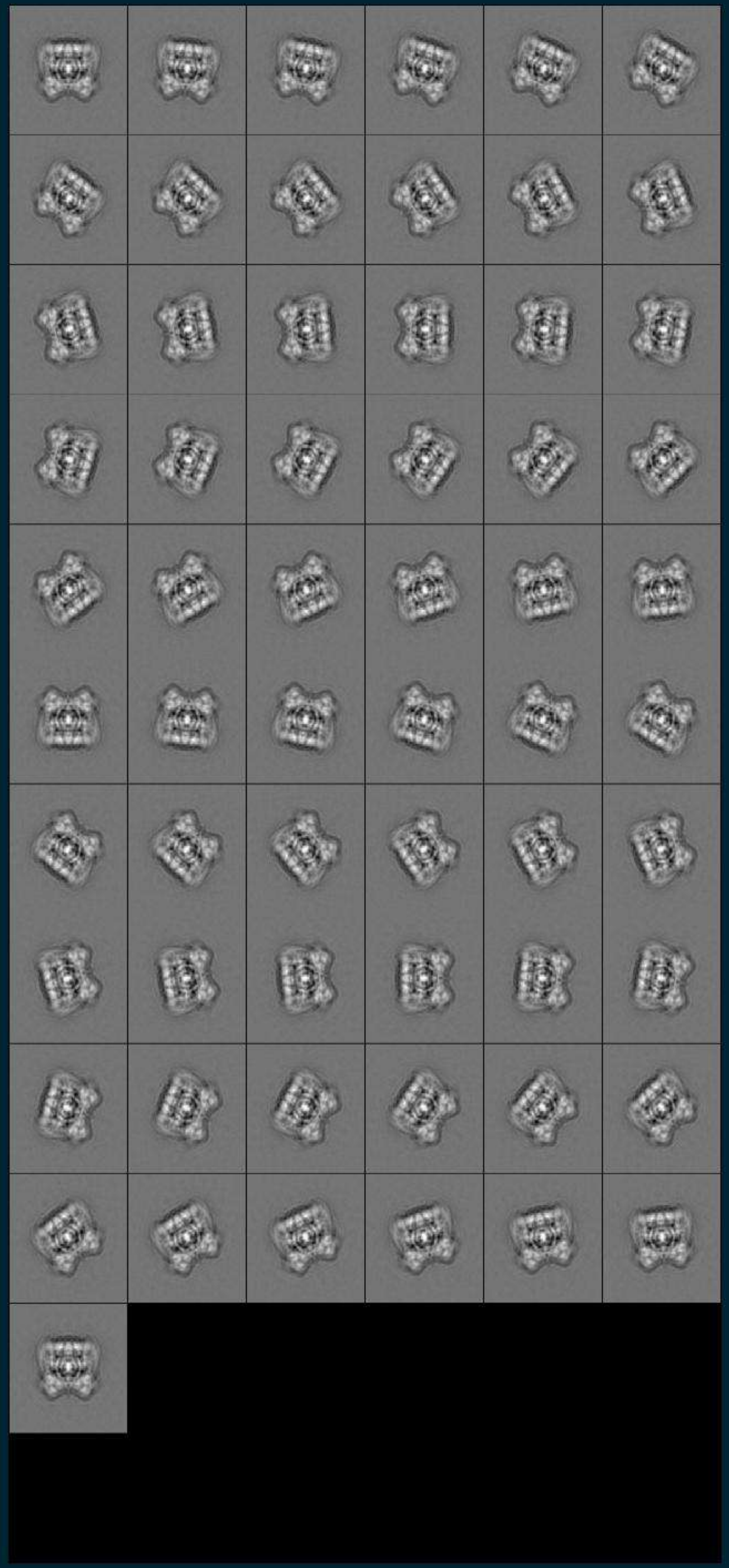


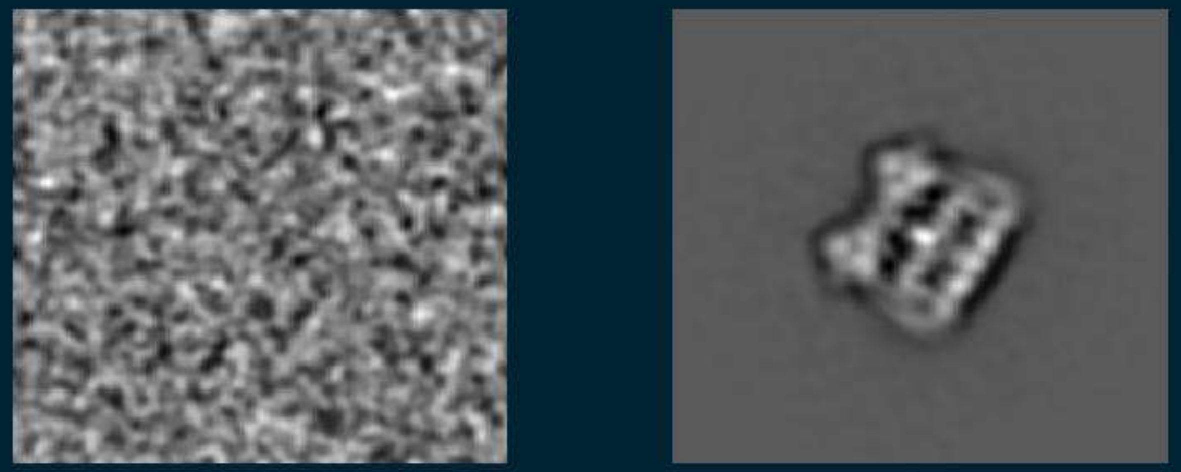


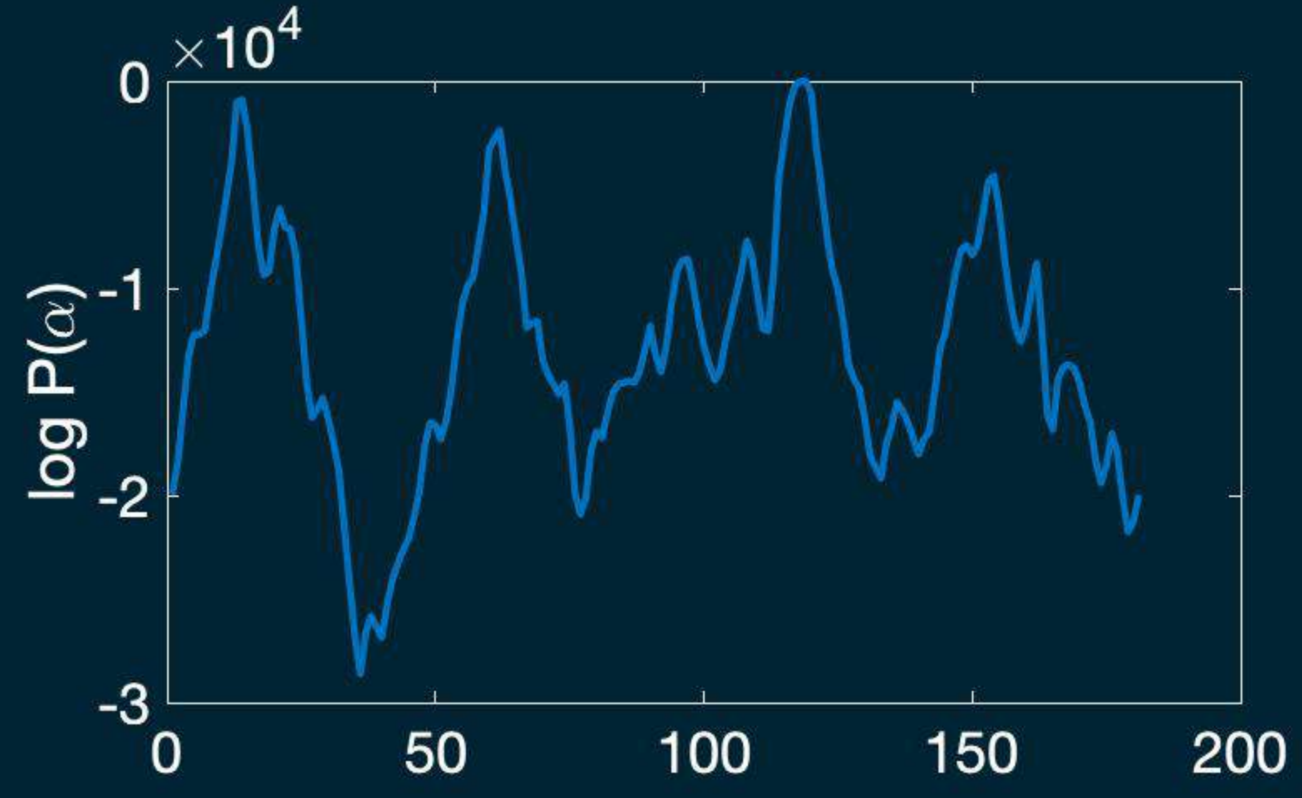


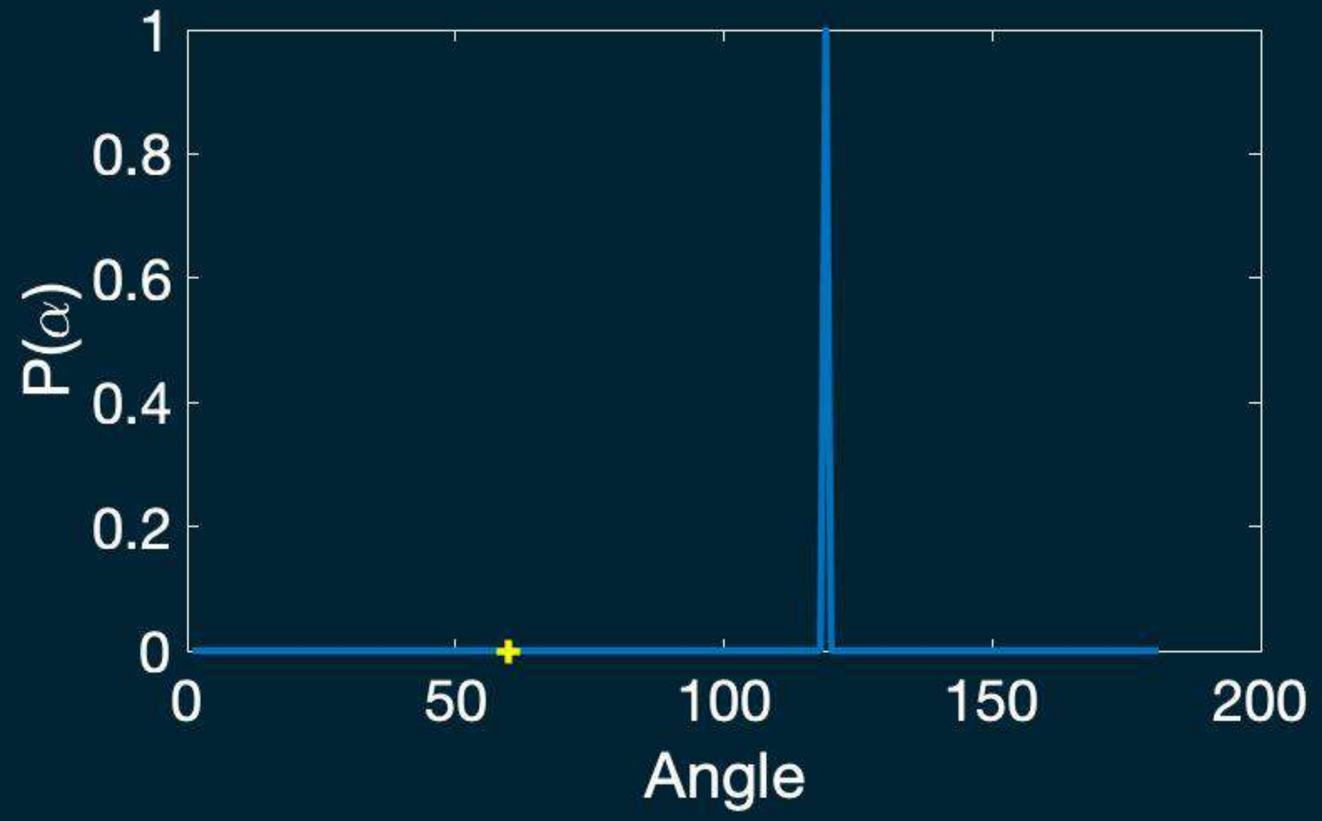


Evaluating Γ_ϕ is expensive: one of 5 parameters

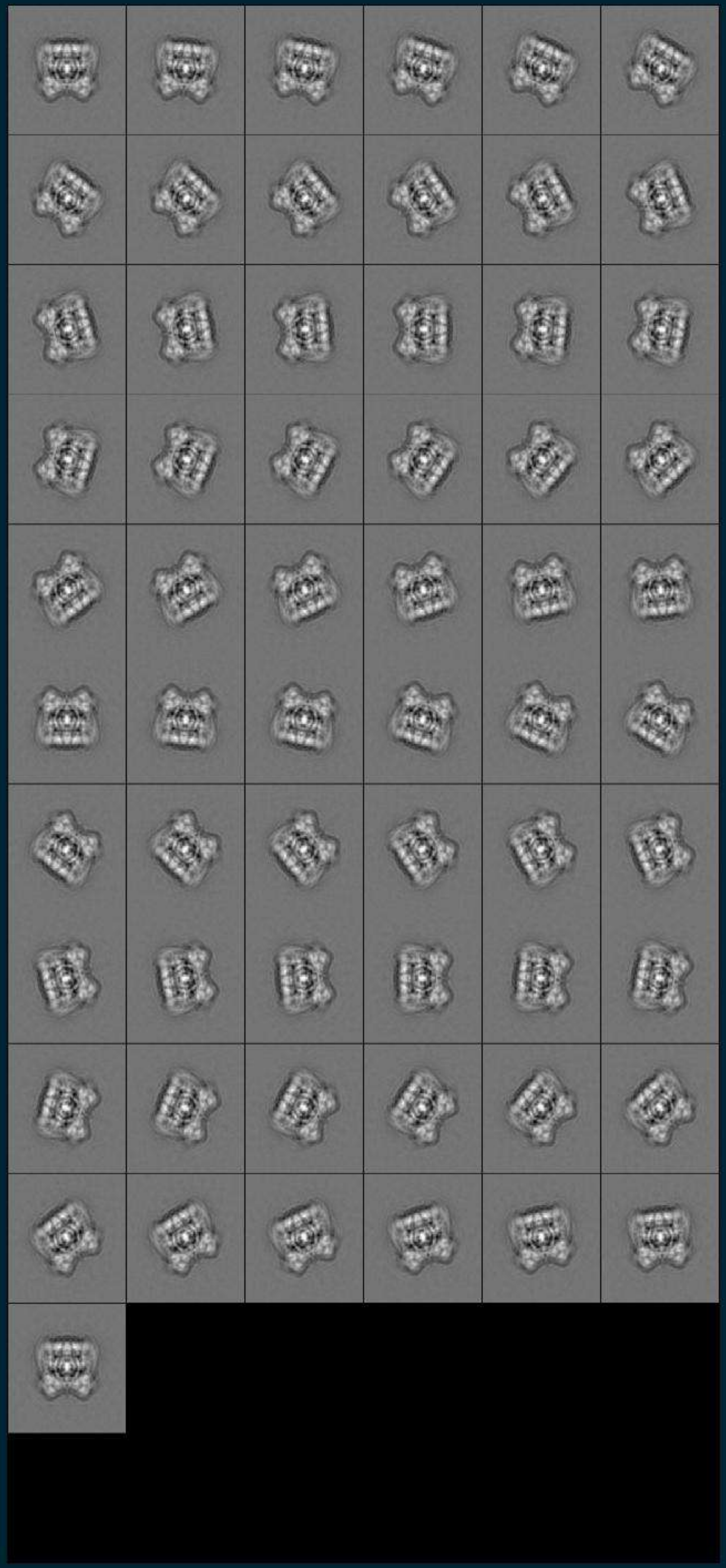


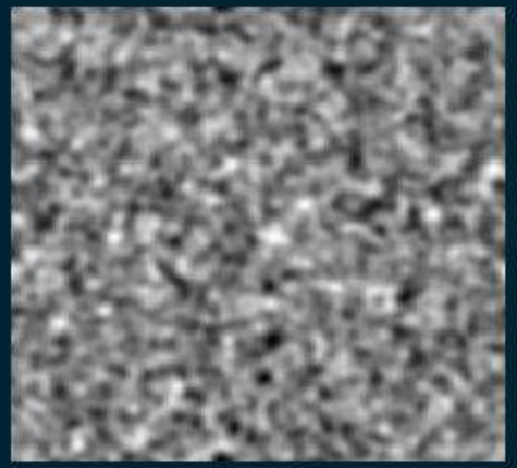


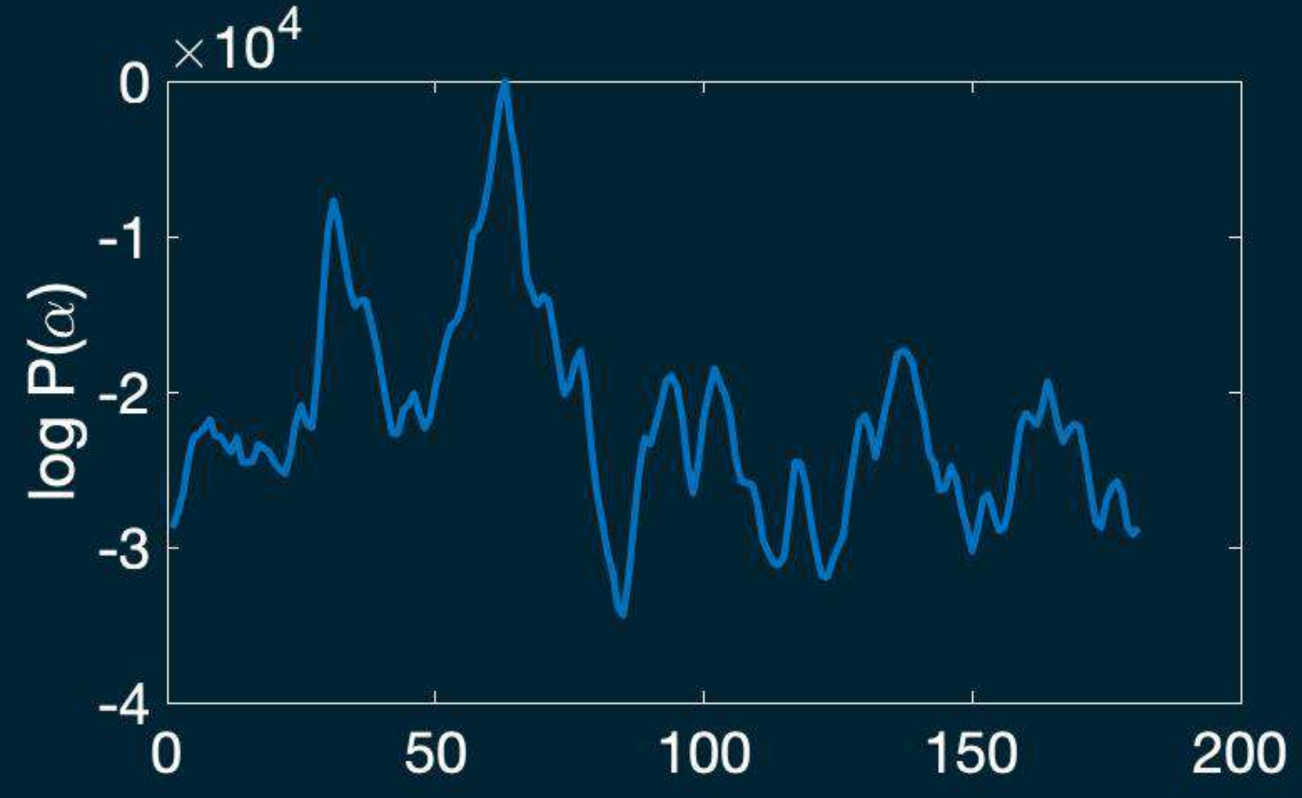


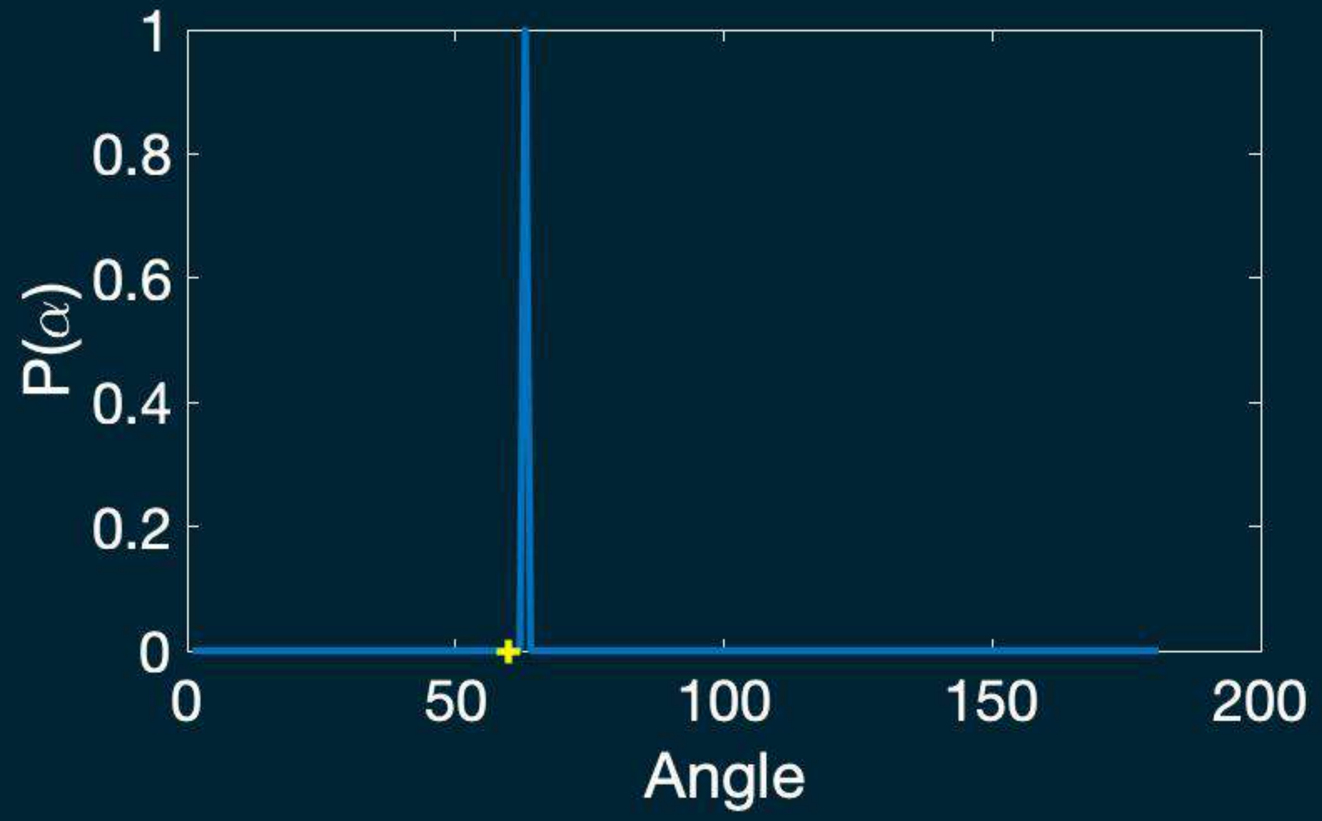


Evaluating Γ_ϕ is expensive: one of 5 parameters

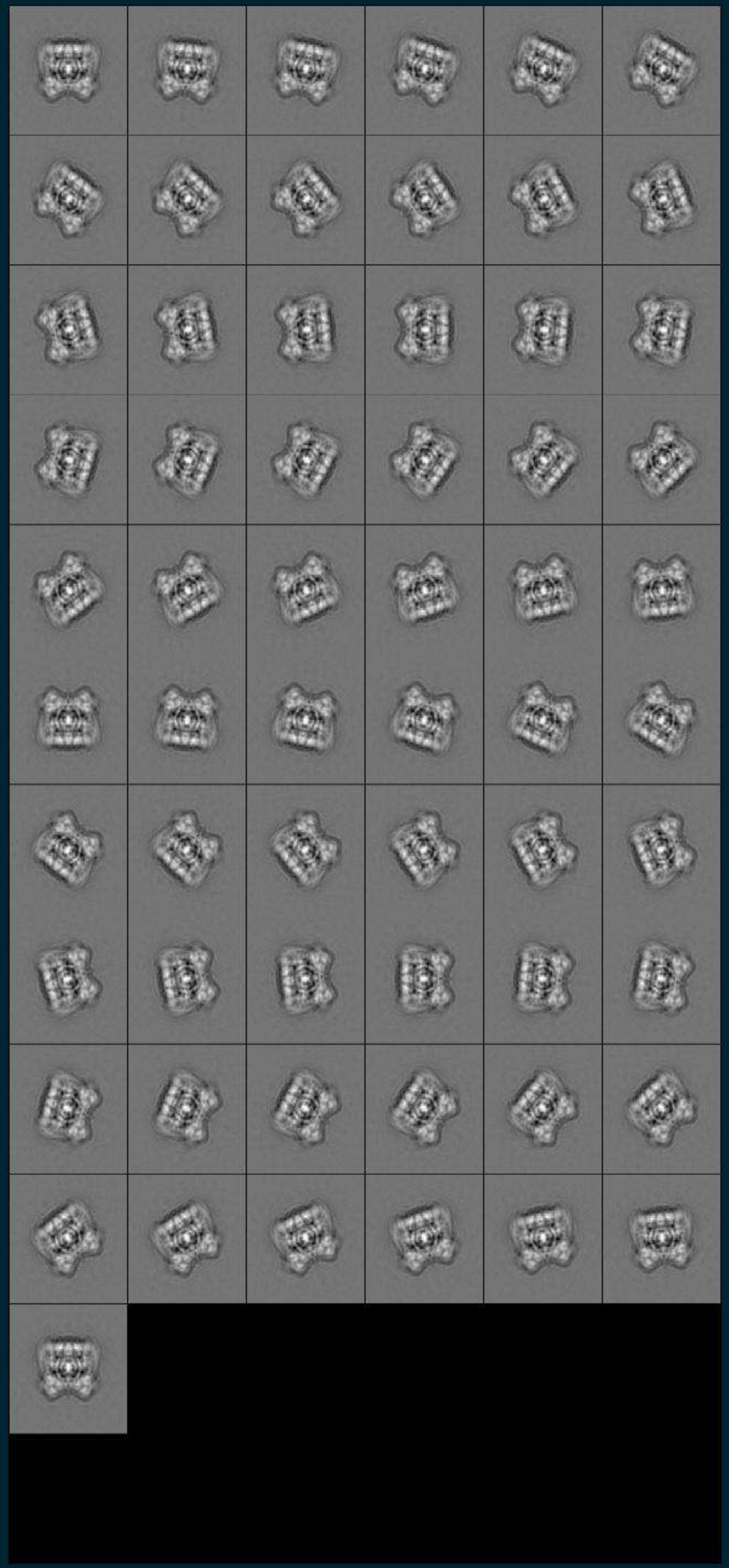


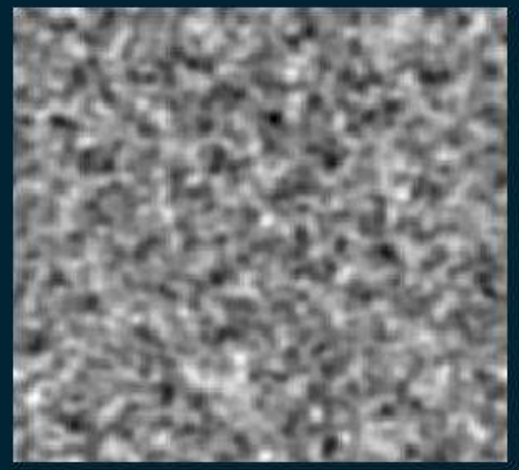


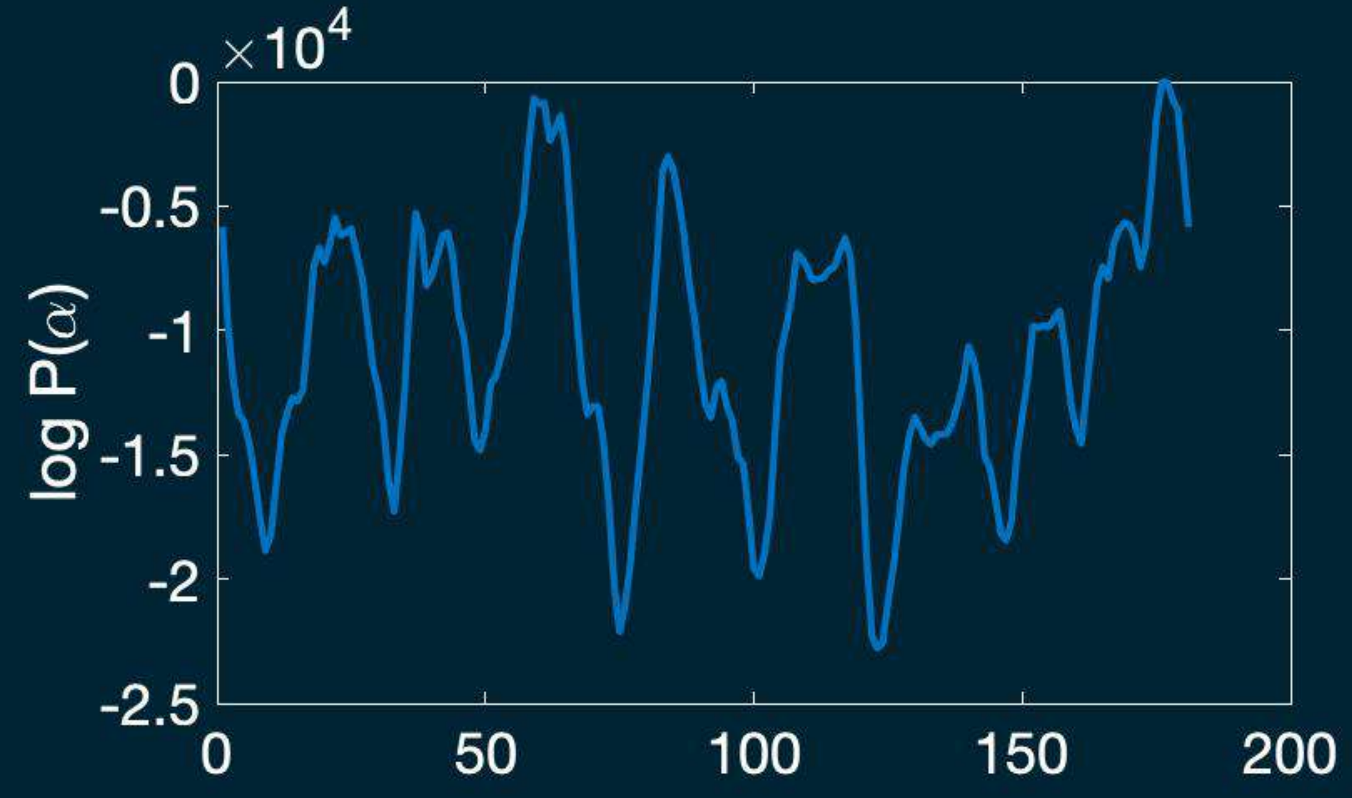


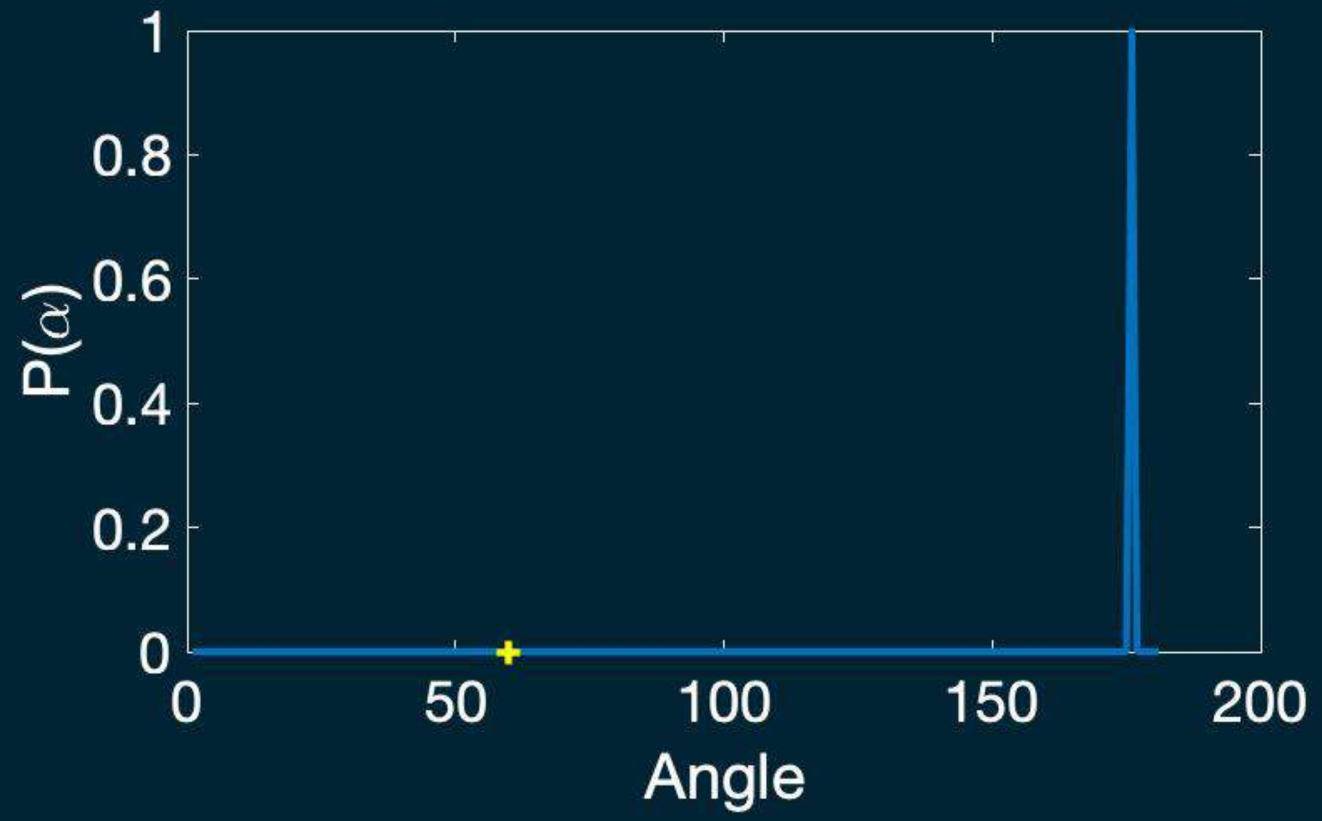


Evaluating Γ_ϕ is expensive: one of 5 parameters



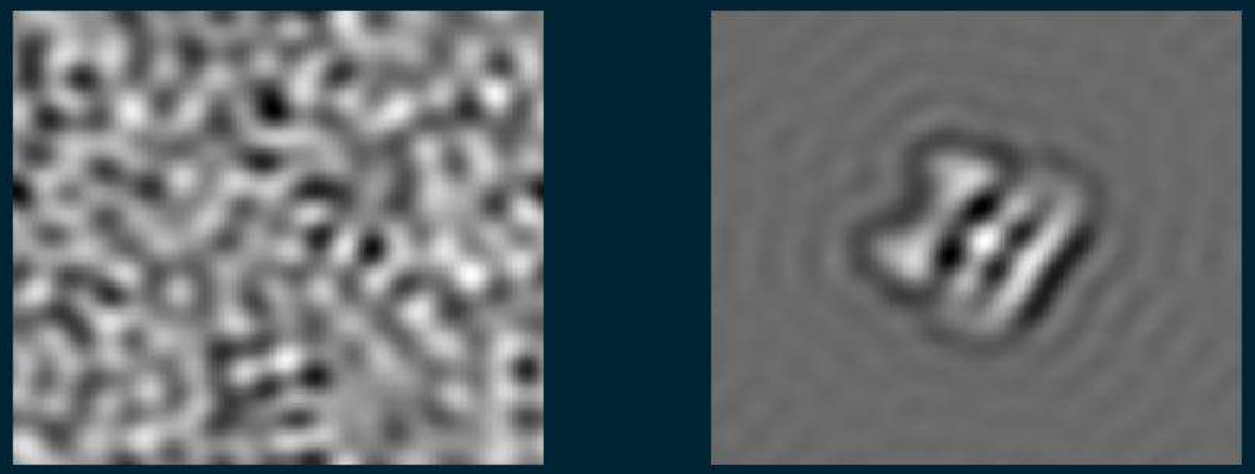




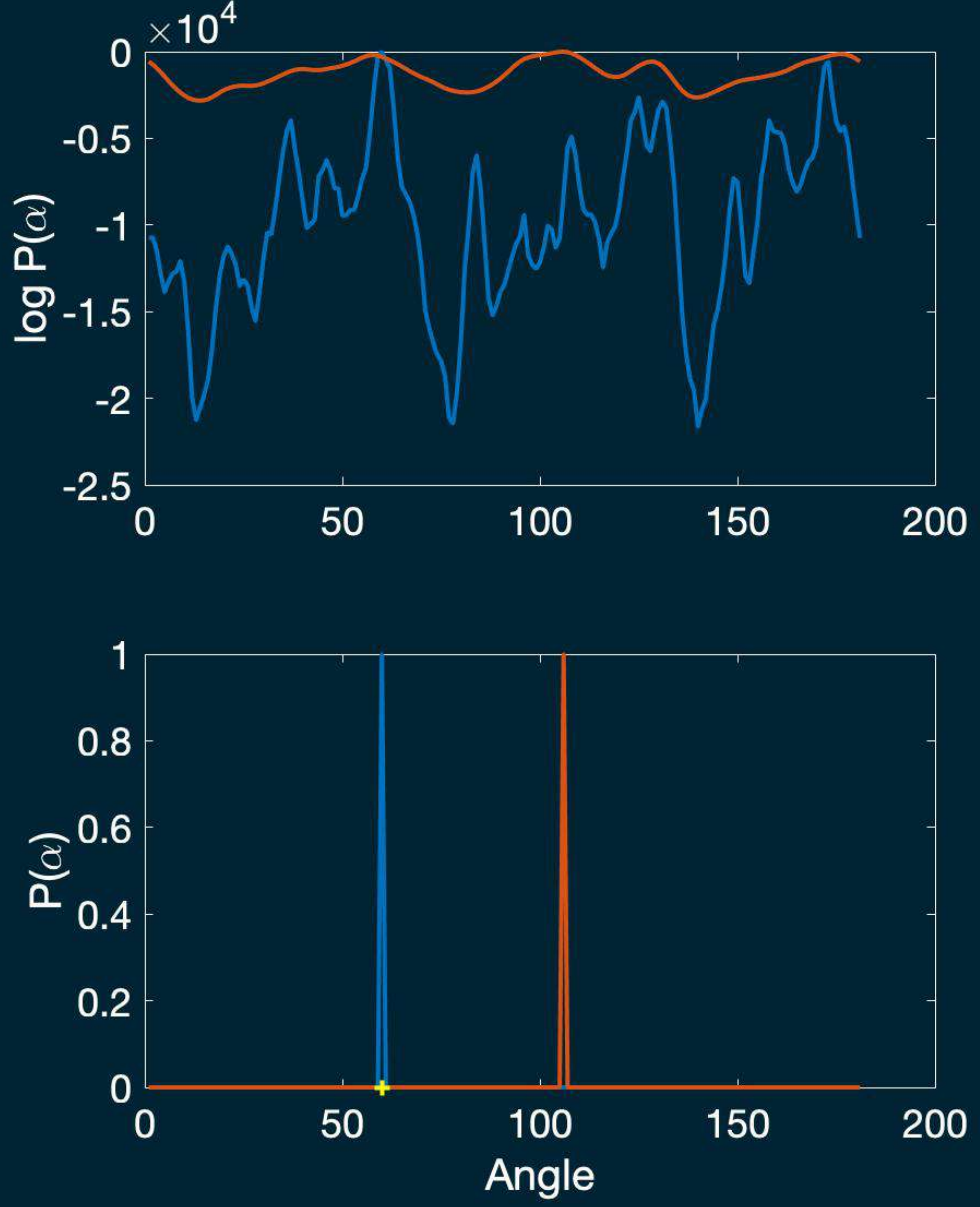


How to decrease the effort?

Domain reduction: branch and bound, illustrated for 1D

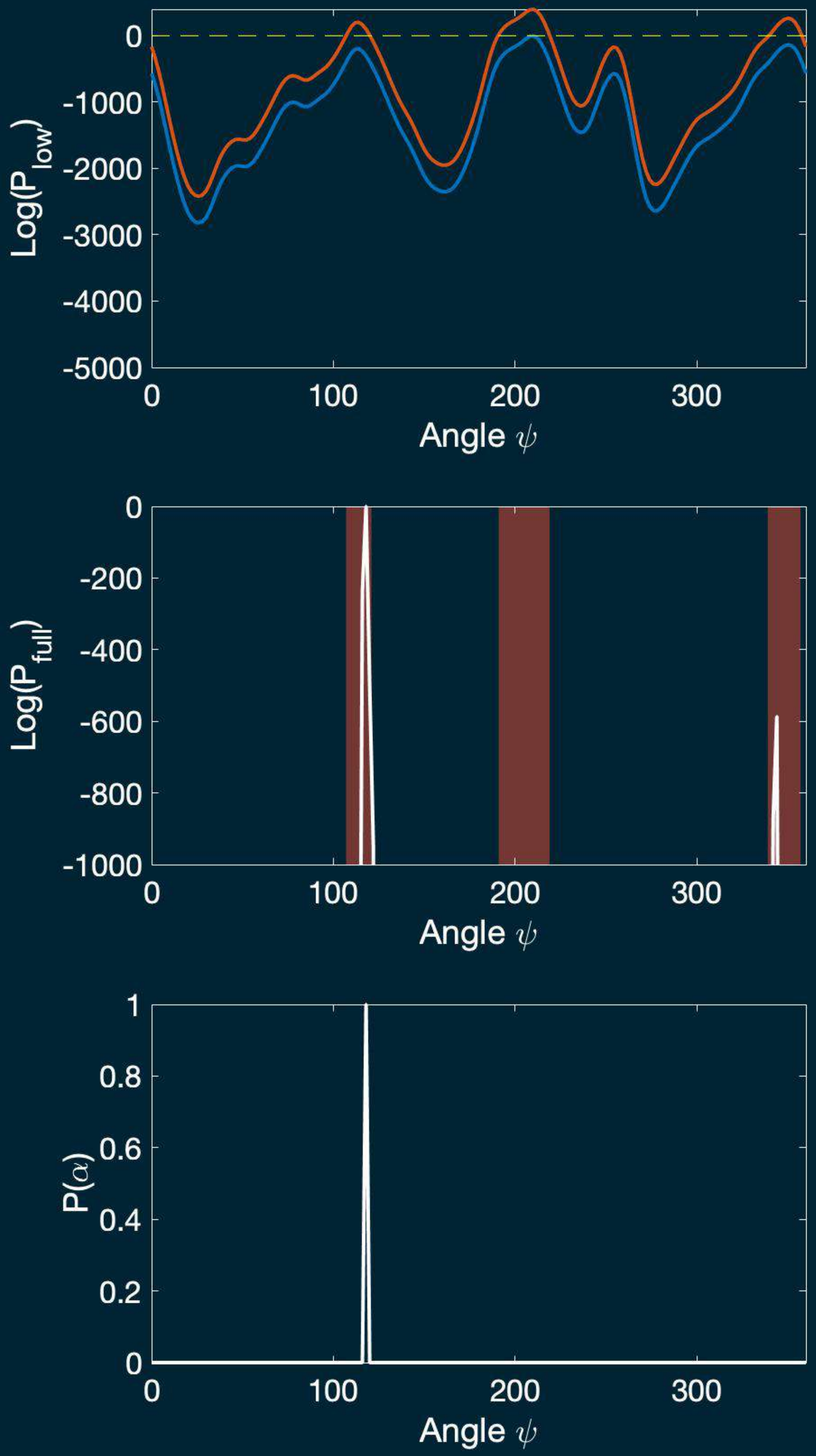


1. To save time, we compute probabilities of orientations at low resolution.

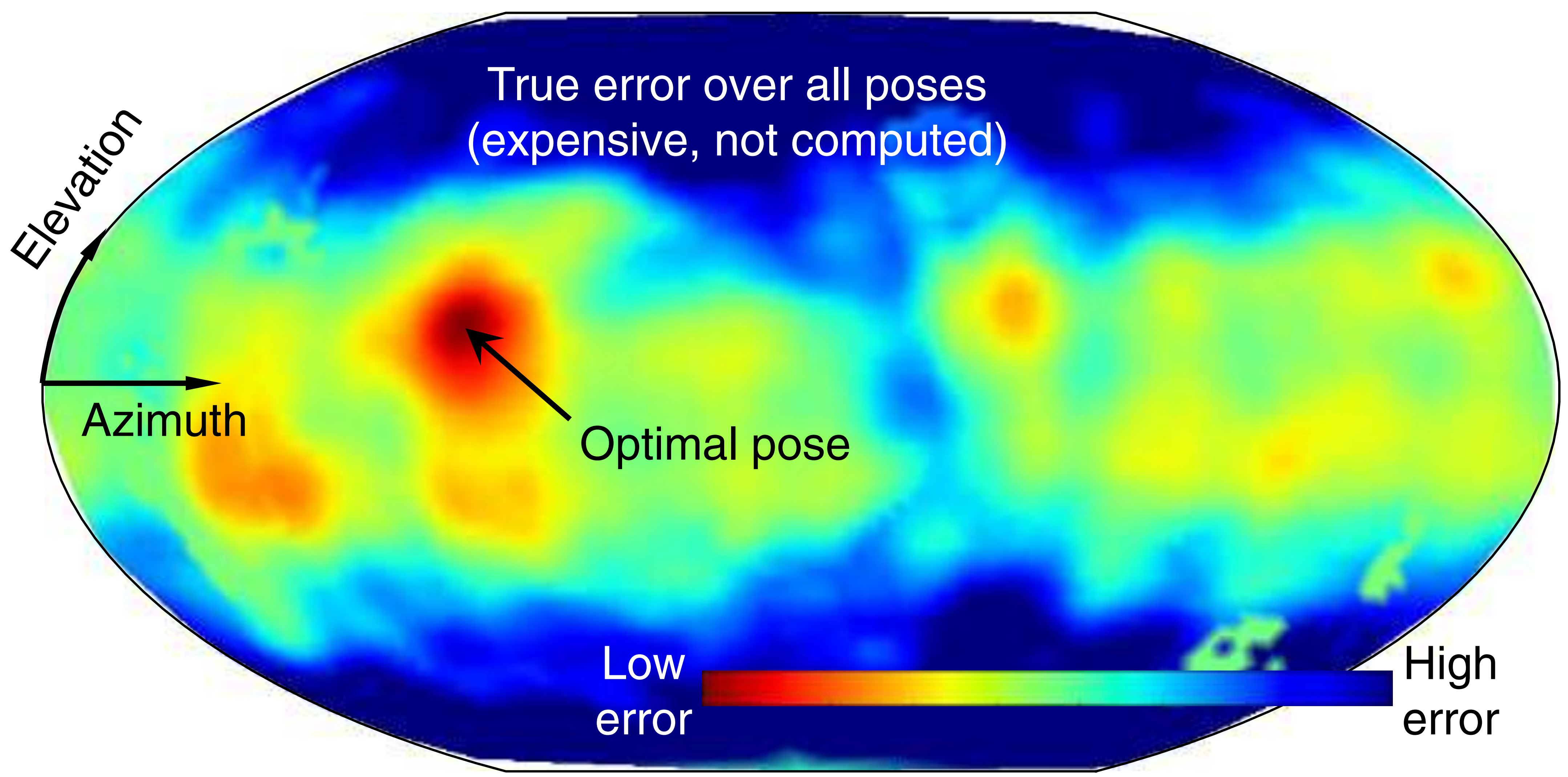


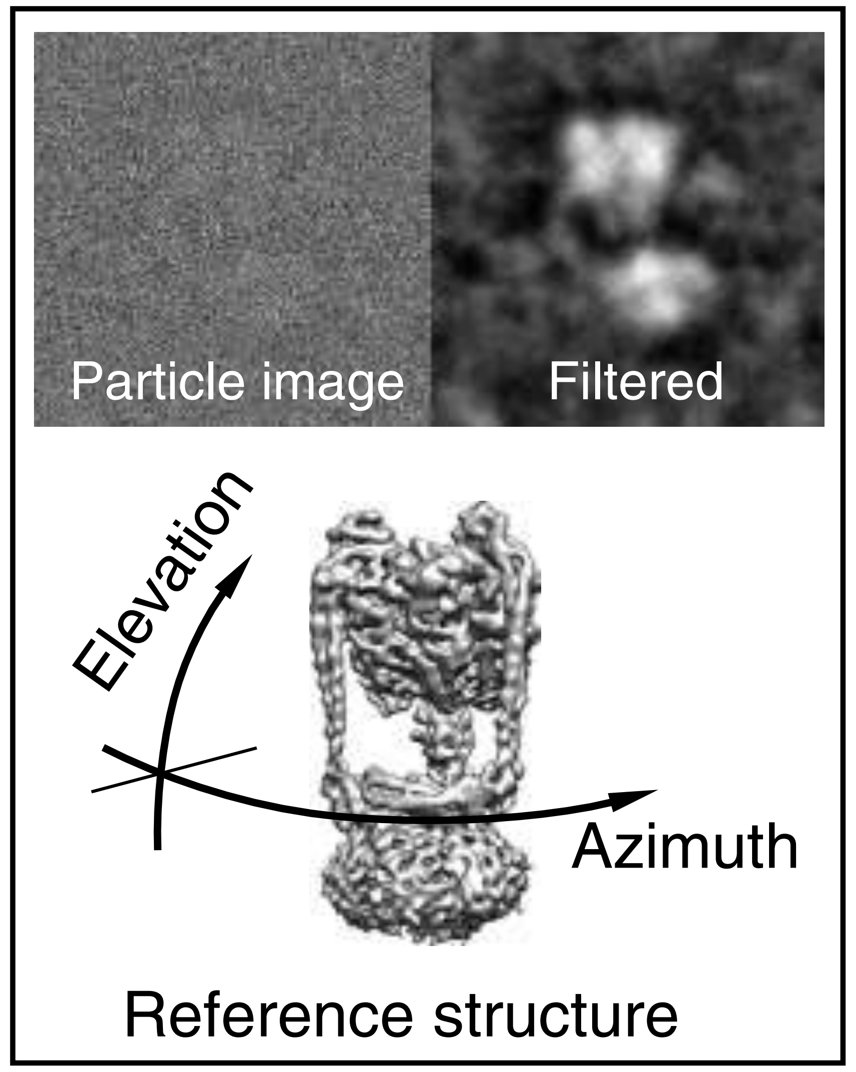
2. We place bounds on how much higher the probabilities could be at full resolution.

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

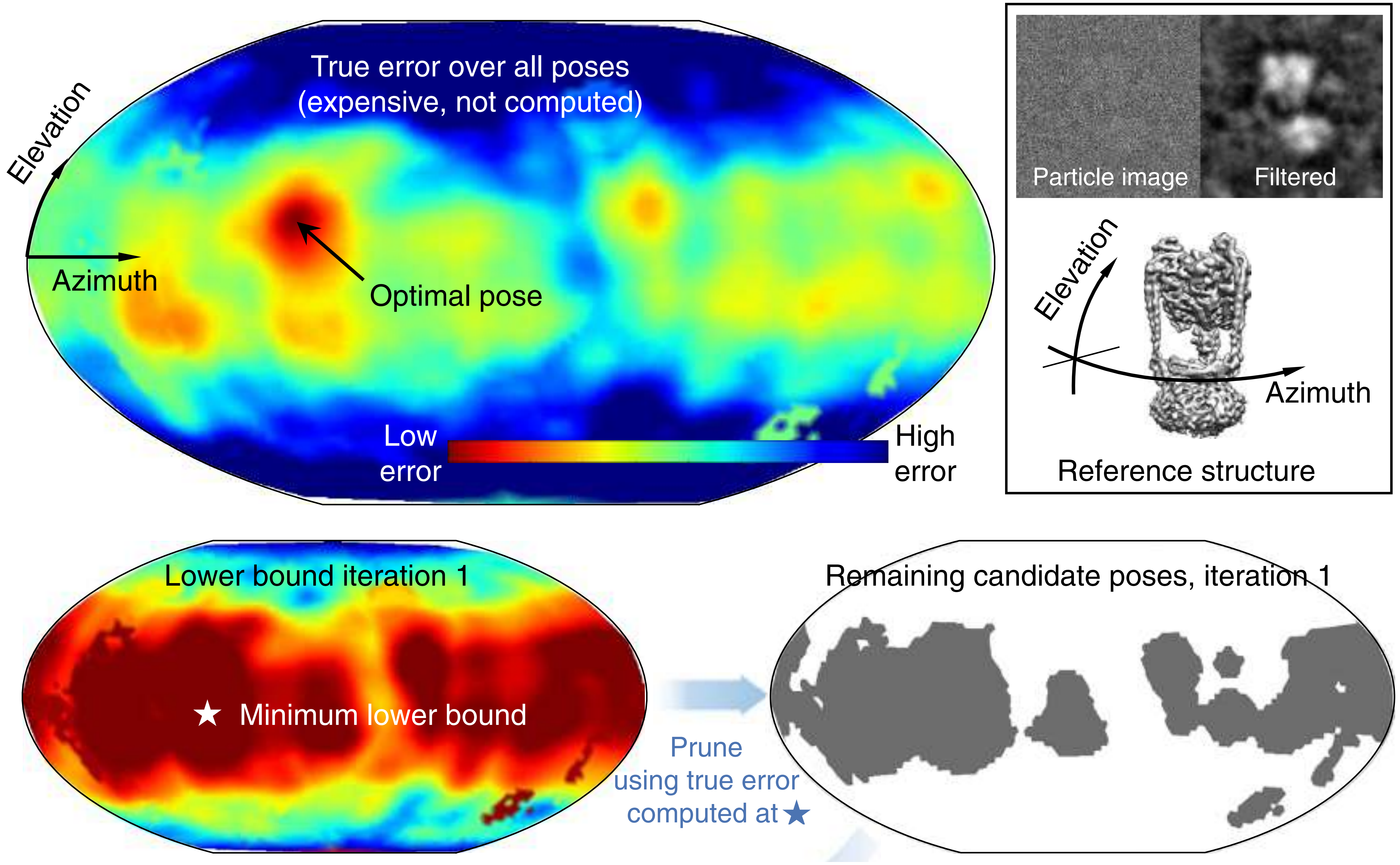


Branch-and-bound in cryoSPARC for integrating over orientations

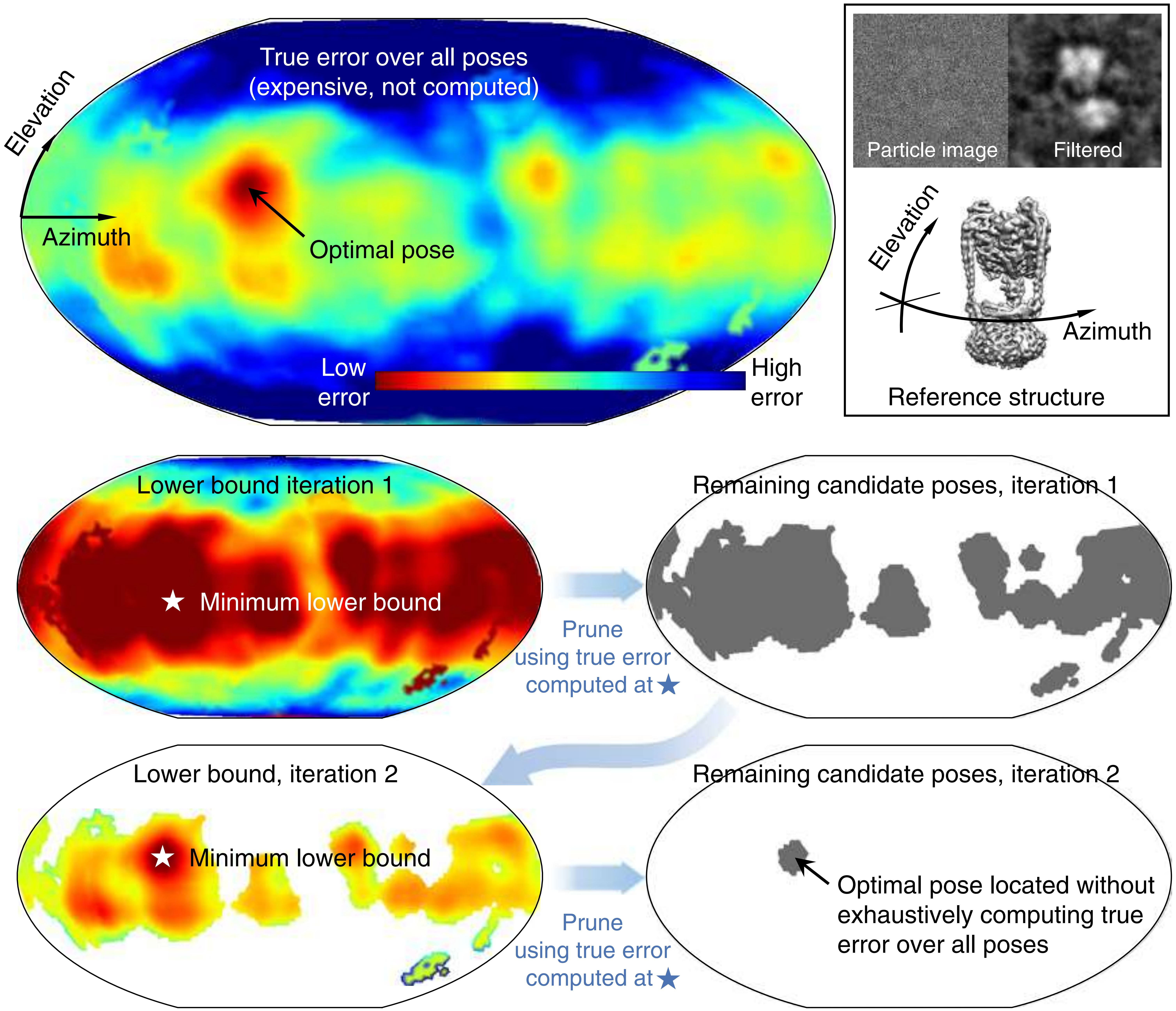




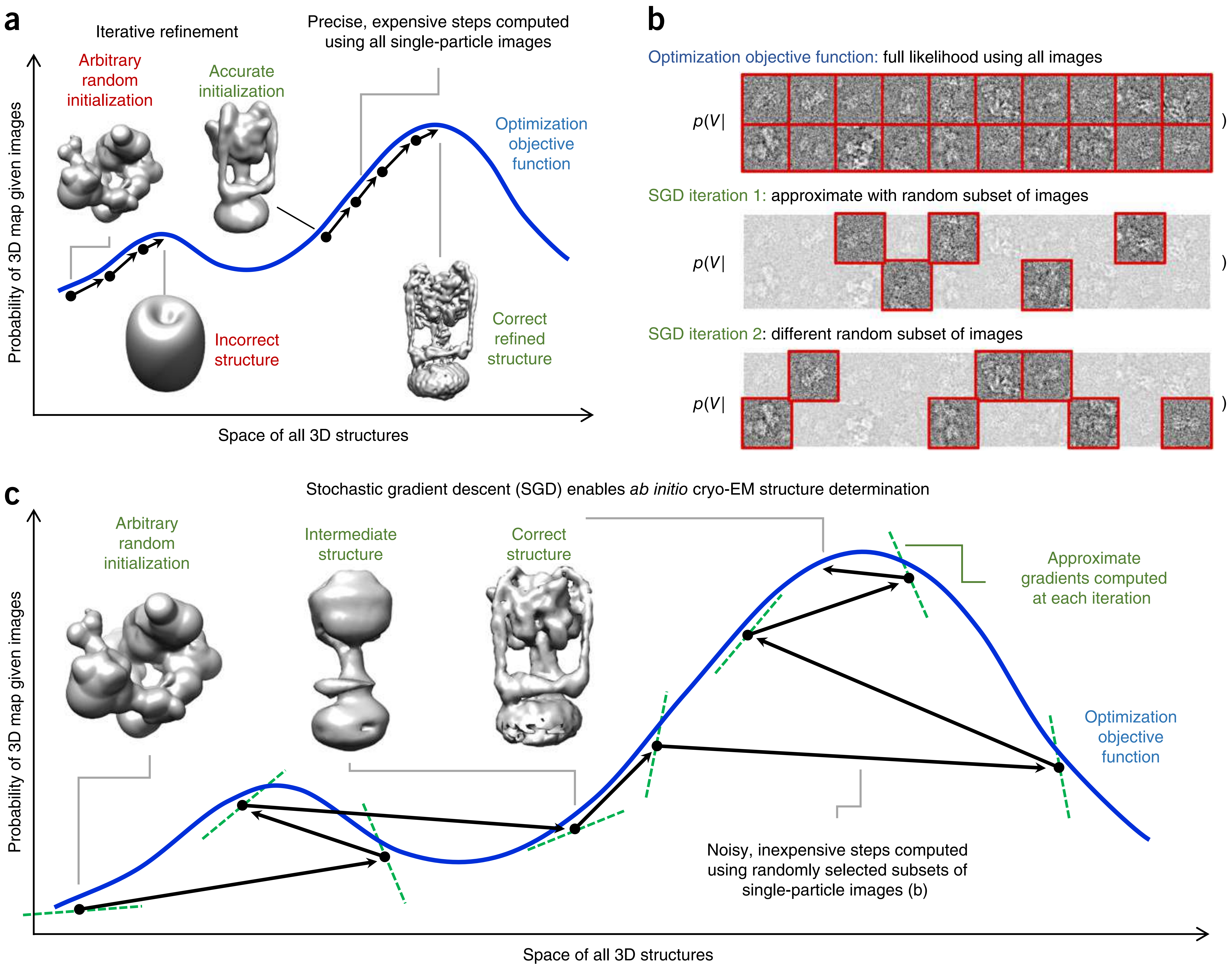
Branch-and-bound in cryoSPARC for integrating over orientations



Branch-and-bound in cryoSPARC for integrating over orientations



Stochastic gradient descent to avoid model bias



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

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