EM Image Formation and Single-Particle Reconstruction

Fred Sigworth
Yale University

Part 1. Phase contrast and the contrast transfer function

Phase contrast and the contrast transfer function

1. Complex numbers: review
2. Defocus contrast and the CTF (simple version)
3. Image delocalization
4. Objective lens effects on the CTF

Why complex numbers?

- They make the equations simpler
- Natural for Fourier transforms
- Give us the magnitude and phase of structure factors

i, the imaginary unit

The unit imaginary number

$$i = \sqrt{-1}$$

A complex number

Real part

$$z = a + ib$$

Imaginary part

You can do arithmetic with complex numbers

$$z = a + ib$$
$$w = c + id$$

Add $z + w = (a + c) + i(b + d)$

Multiply $zw = (ab - bd) + i(ad + bc)$

Real part $\mathrm{Re}(z) = a$

Imaginary part $\mathrm{Im}(z) = b$

Absolute value $|z| = \sqrt{a^2 + b^2}$

Conjugate $z^* = a - ib$

The exponential function e^x

$$e = 2.718...$$

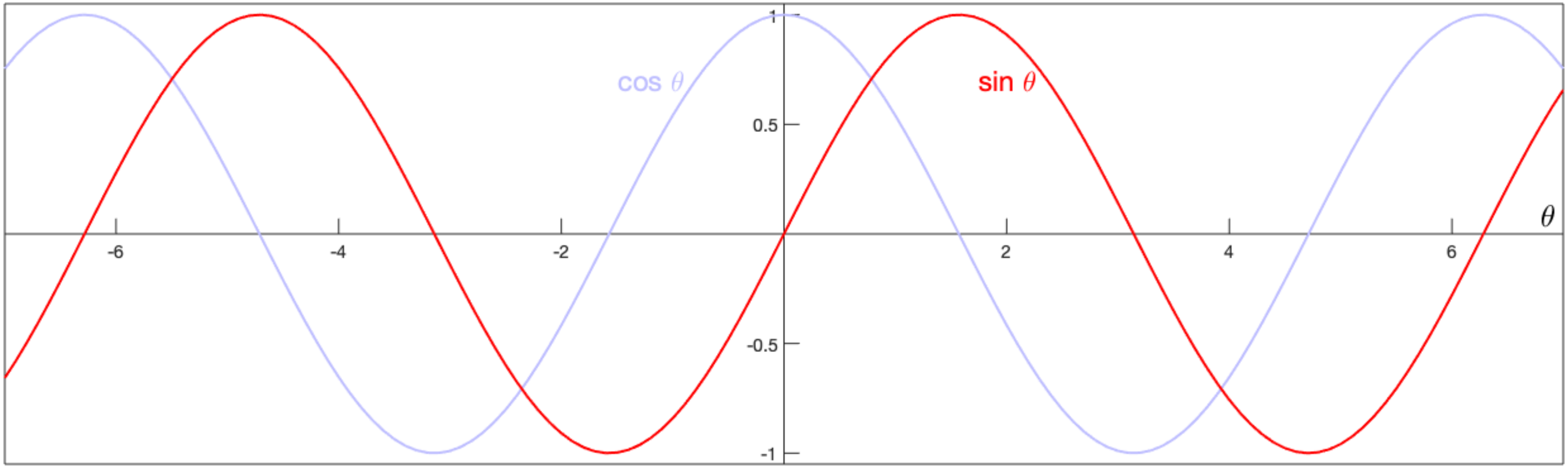
$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{2 \times 3} + \ldots$$

A very important approximation

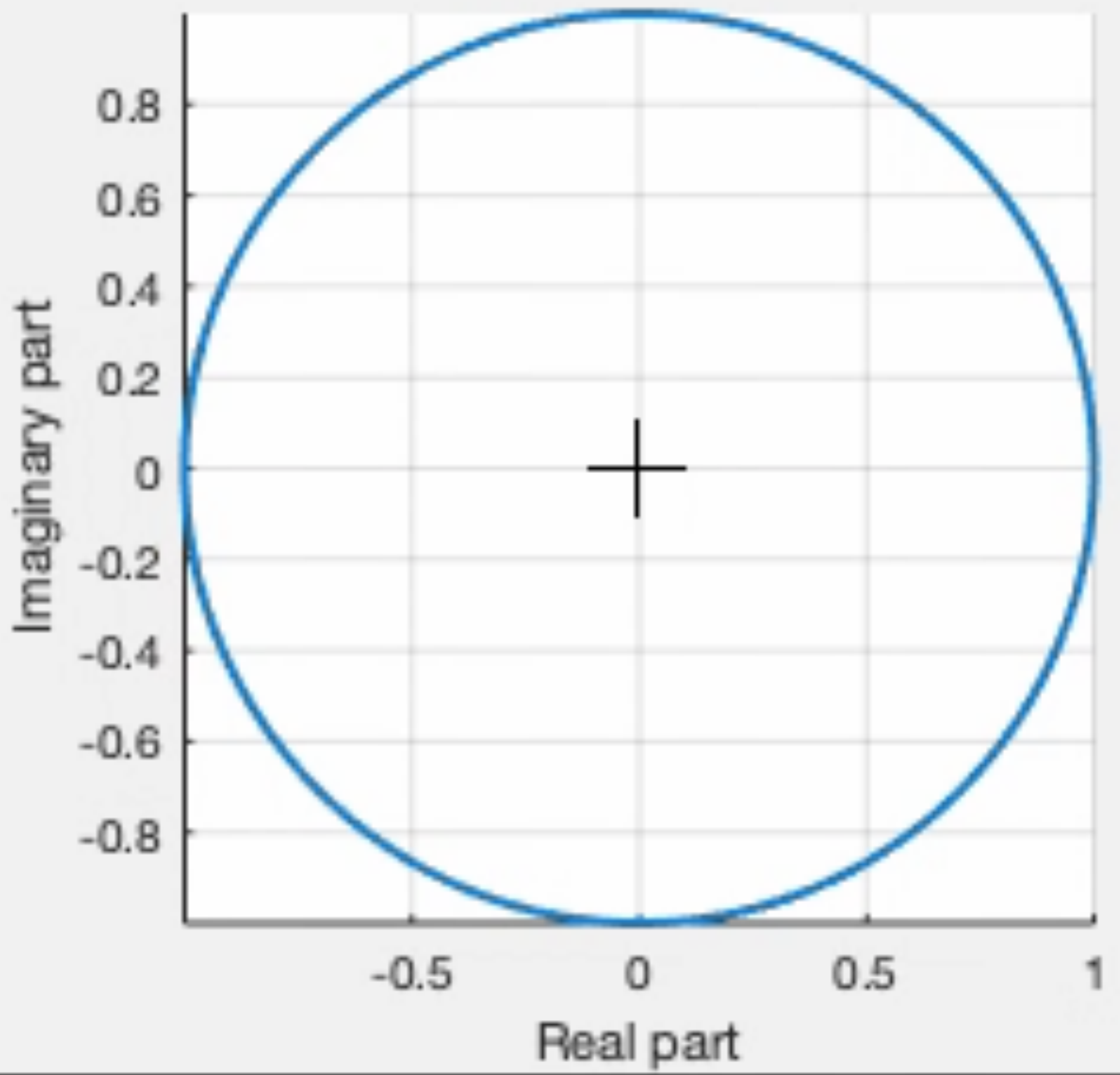
$$e^x \approx 1 + x, \quad x \ll 1$$

The complex exponential

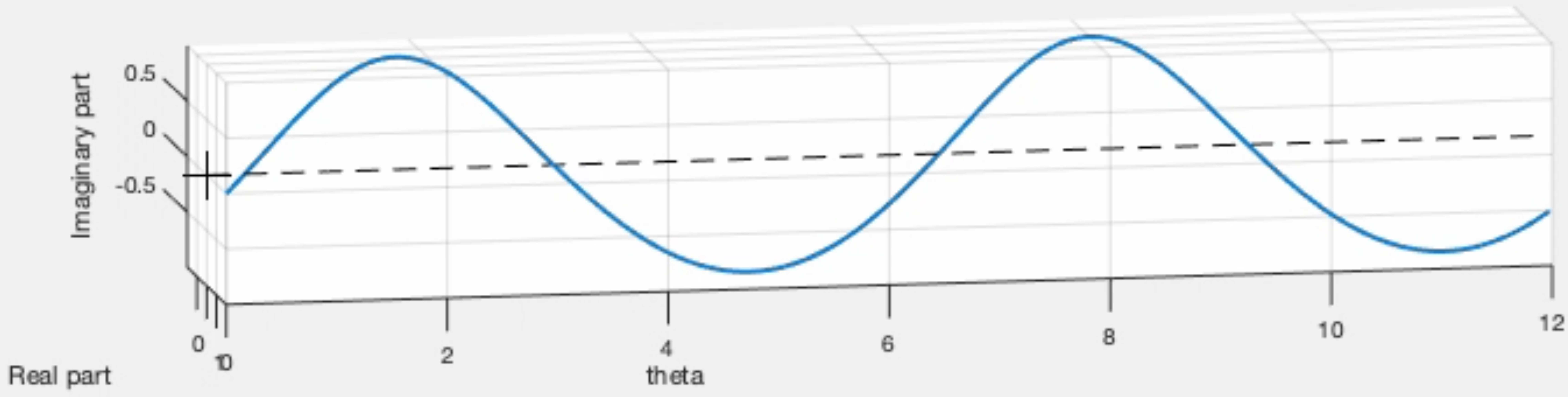
$$e^{i\theta} = \cos\theta + i\sin\theta$$



A plot of $e^{i\theta}$



A plot of $e^{i\theta}$



Any z can be represented as (a, b) or as (r, θ)

$$z = a + ib$$

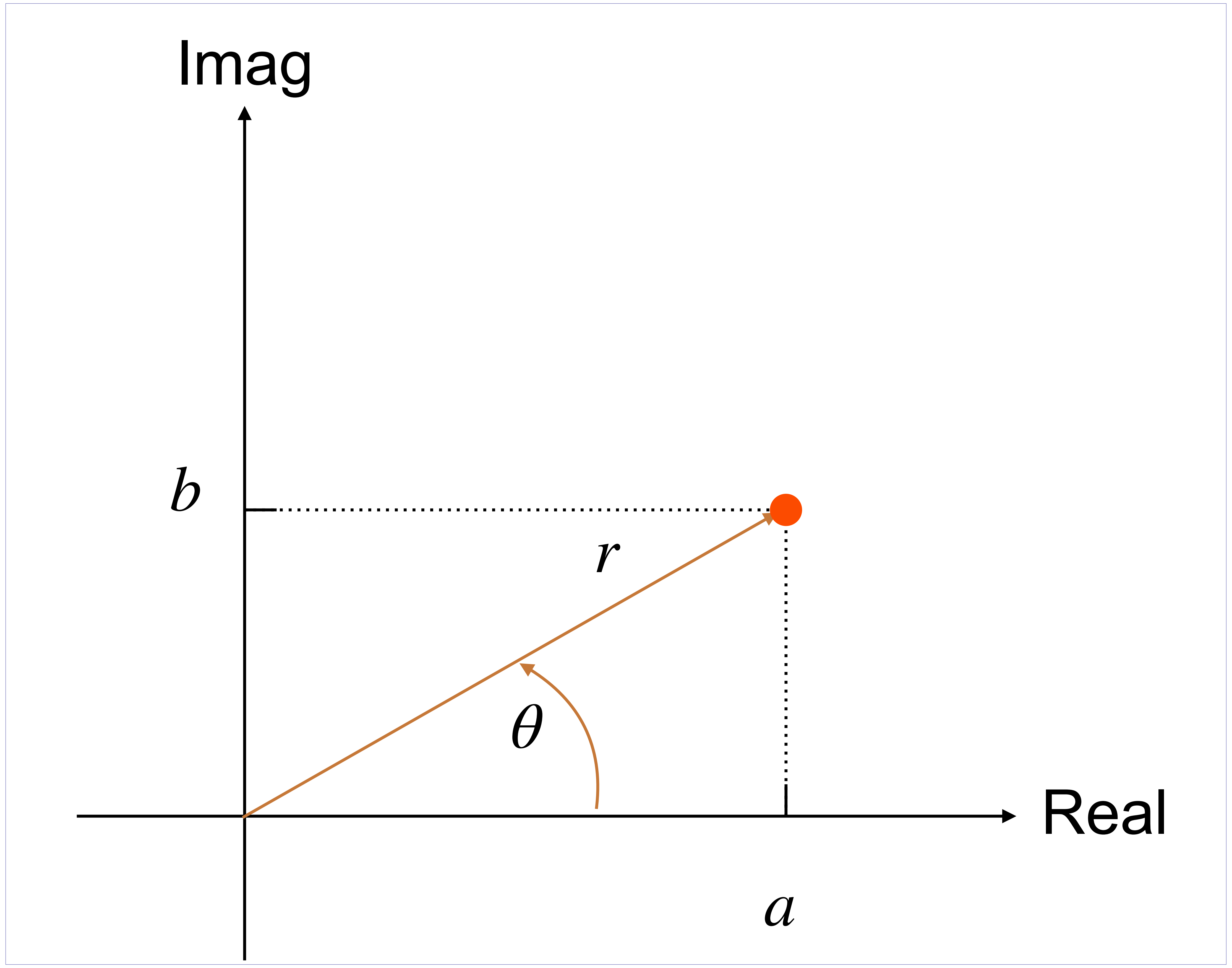
a is the real part

b is the imaginary part

$$z = re^{i\theta}$$

r is the magnitude

θ is the phase



Recall that

$$e^x e^y = e^{x+y}$$

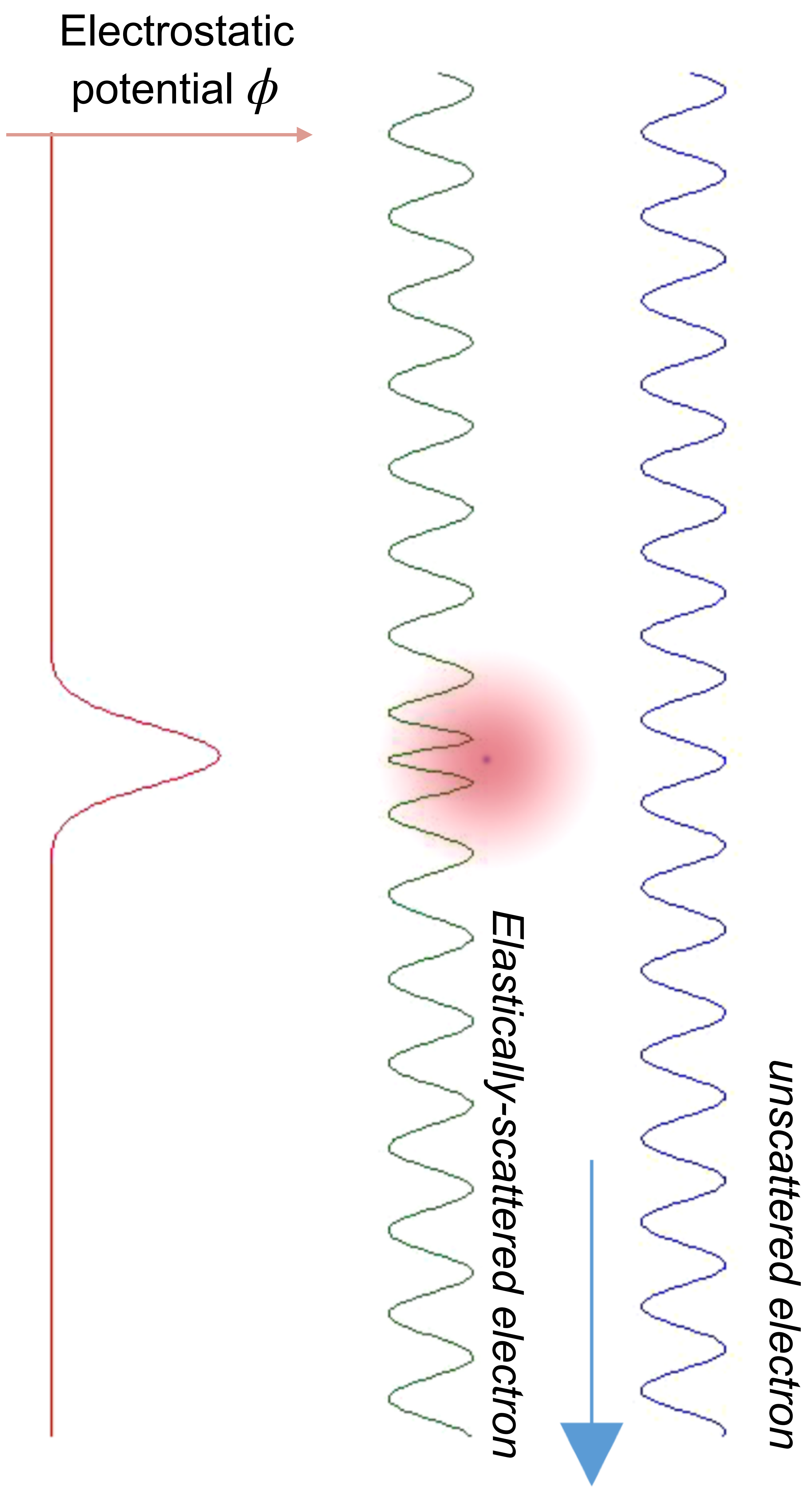
so, when you multiply two complex numbers, the phases add:

$$e^{i\theta_1} e^{i\theta_2} = e^{i(\theta_1+\theta_2)}.$$

Phase contrast and the contrast transfer function

1. Complex numbers: review
2. Defocus contrast and the CTF (simple version)
3. Image delocalization
4. Objective lens effects on the CTF

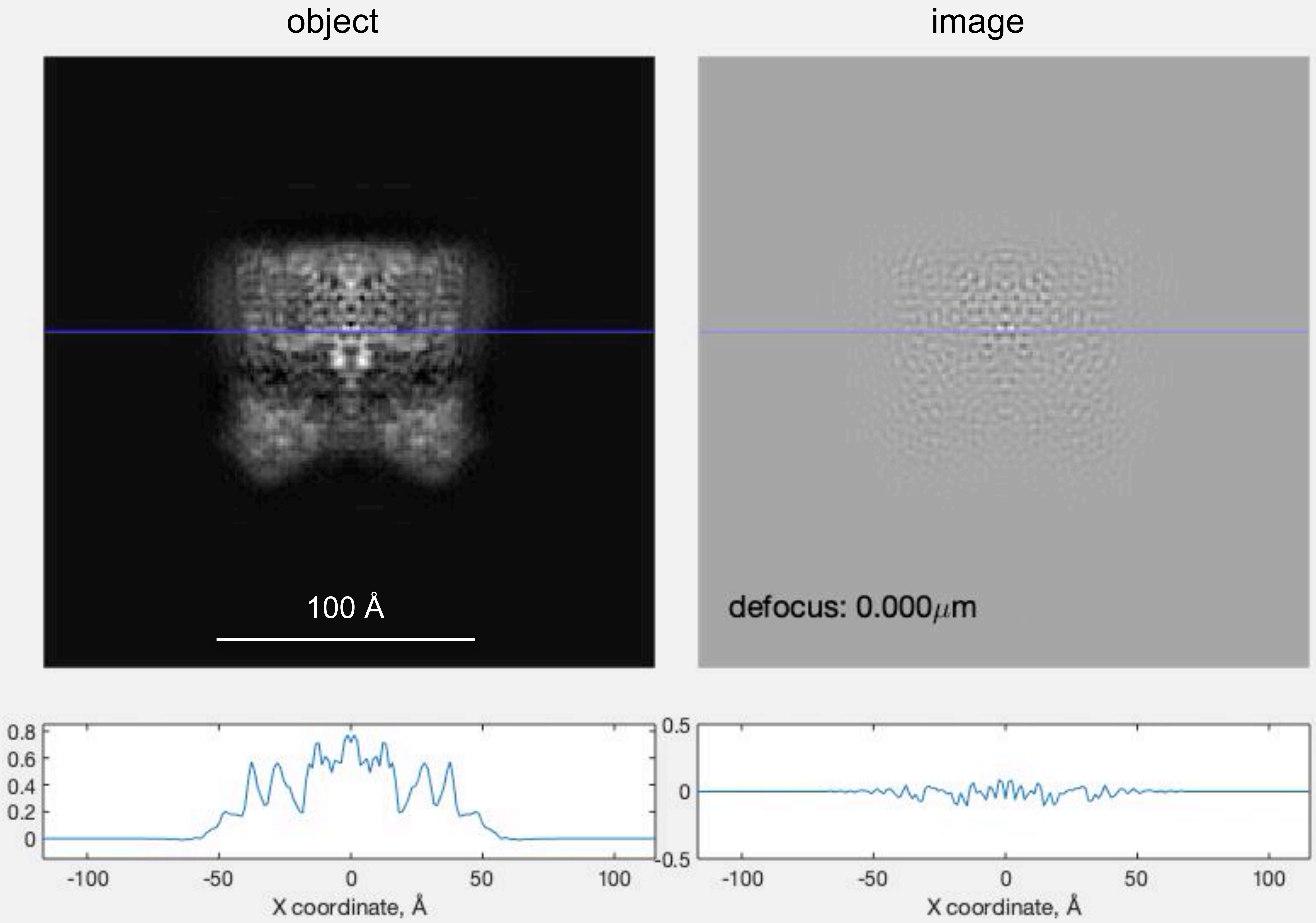
Cryo-EM specimens are imaged by phase contrast



The imaging electrons are phase-shifted when passing near atomic nuclei or fixed charges.

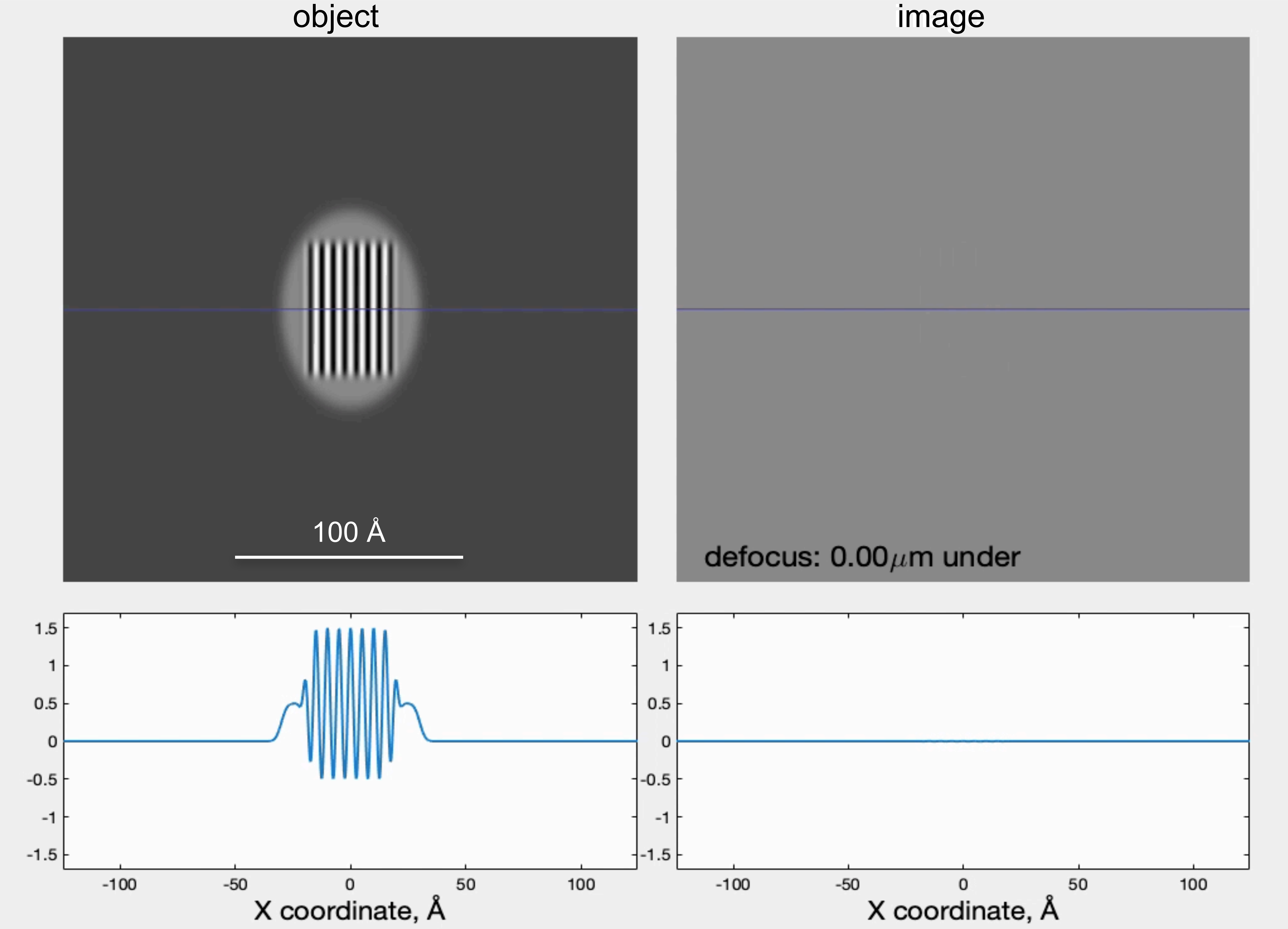
The phase shift coefficient σ is about 0.5 milliradian per volt-angstrom of integrated potential.
The phase shift near a single atom is ~1 milliradian.

Most cryo-EM data are acquired using defocus contrast



- At high defocus, high-resolution information in the image is strongly delocalized.
- Image processing can re-localize the signals, but at most only about half of the theoretical contrast is preserved by defocusing.

Image of an object with 5Å periodicity

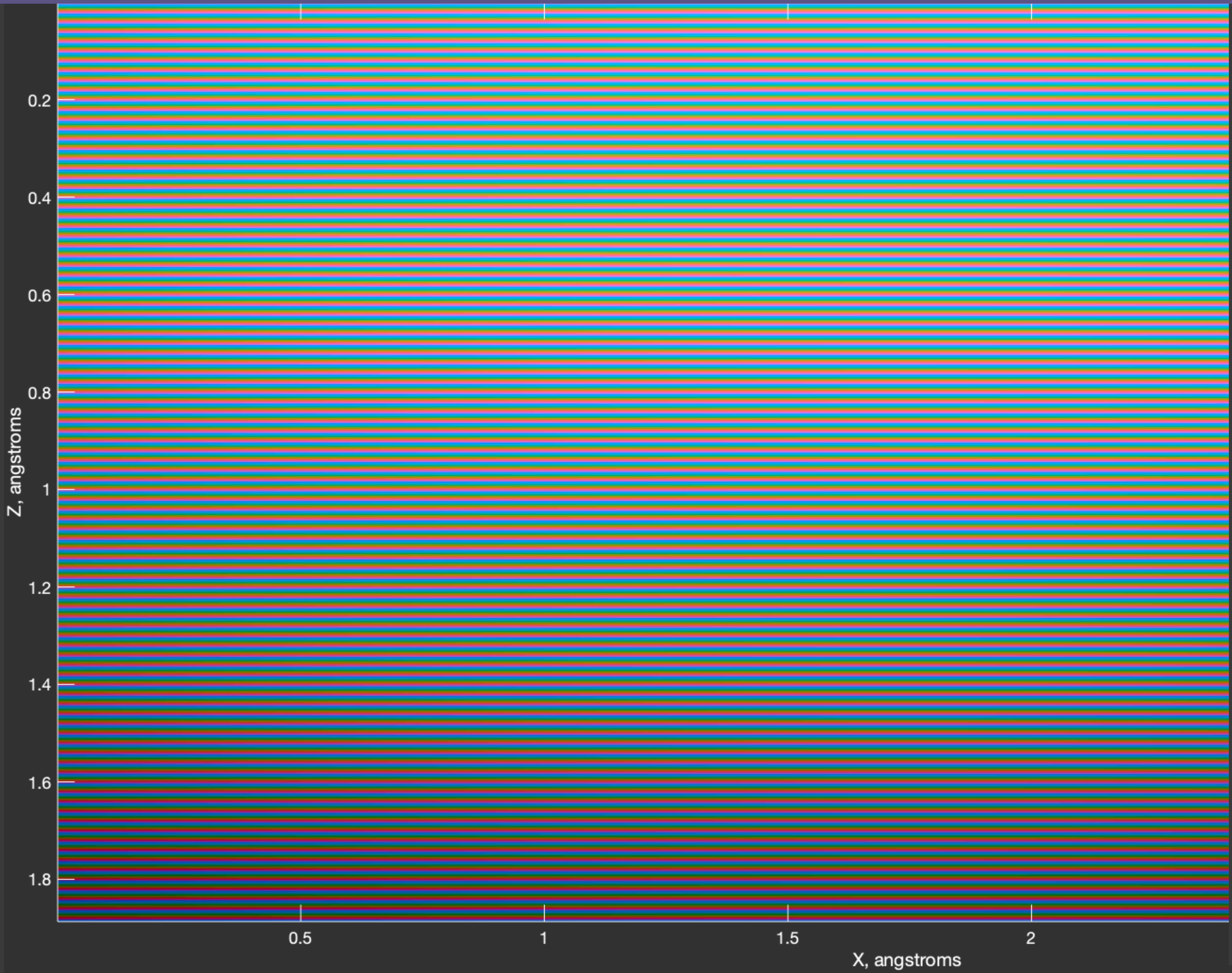


- At high defocus, high-resolution information in the image is strongly delocalized.
- Image processing can re-localize the signals, but at most only about half of the theoretical contrast is preserved by defocusing.

Defocus contrast in a nutshell

1. The contrast in the image of a grating object varies with the amount of defocus.
2. The grating object produces diffracted waves with shifting phase.
3. When the diffracted waves interfere with the undiffracted waves, we have contrast.

A snapshot of an electron wave



Energy (keV)	Wavelength (Å)	Velocity (fraction of c)
120	0.033	0.59
200	0.025	0.70
300	0.020	0.78

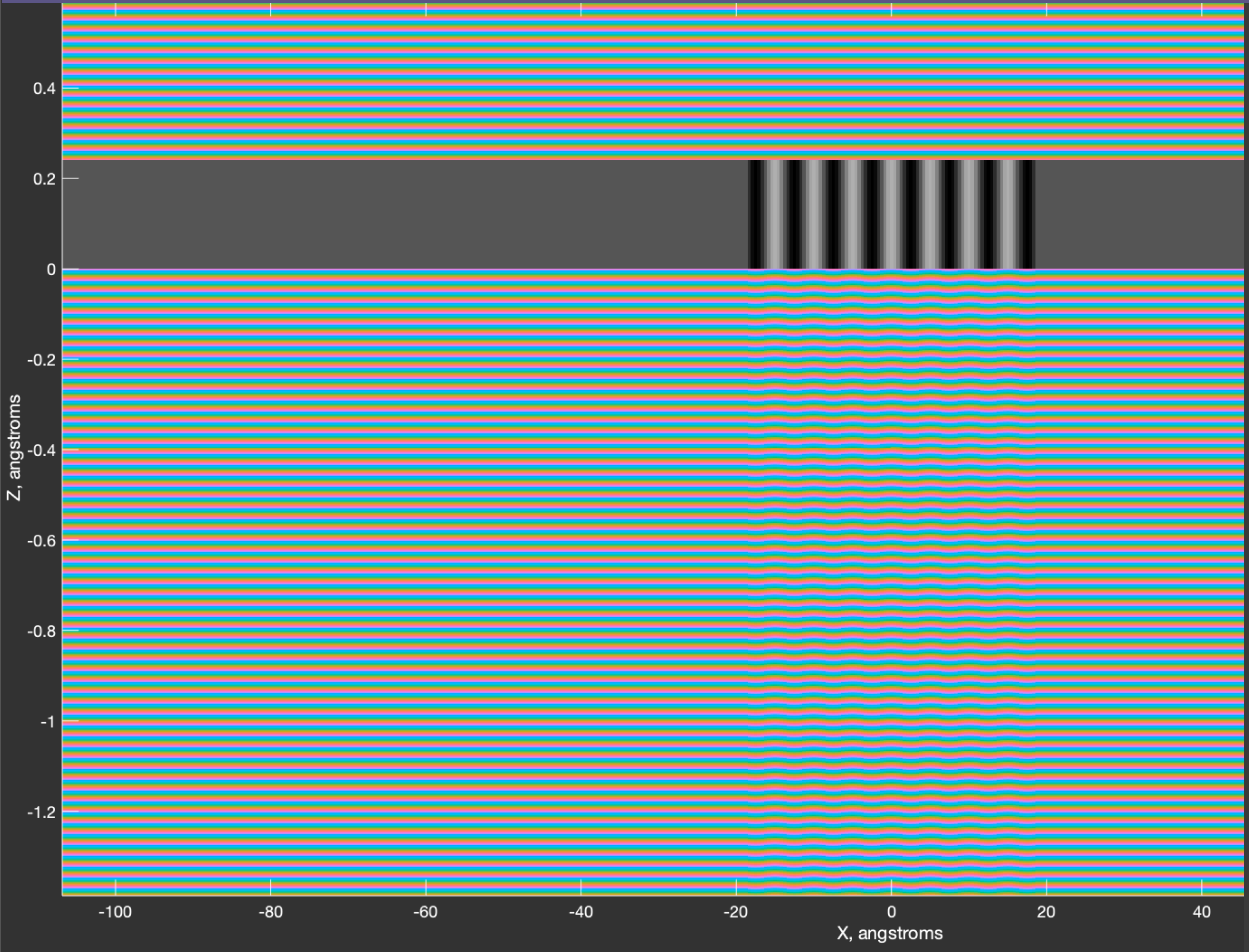
For an electron propagating in the z direction, the time-independent wave function is

$$\Psi_0 = e^{ikz}$$

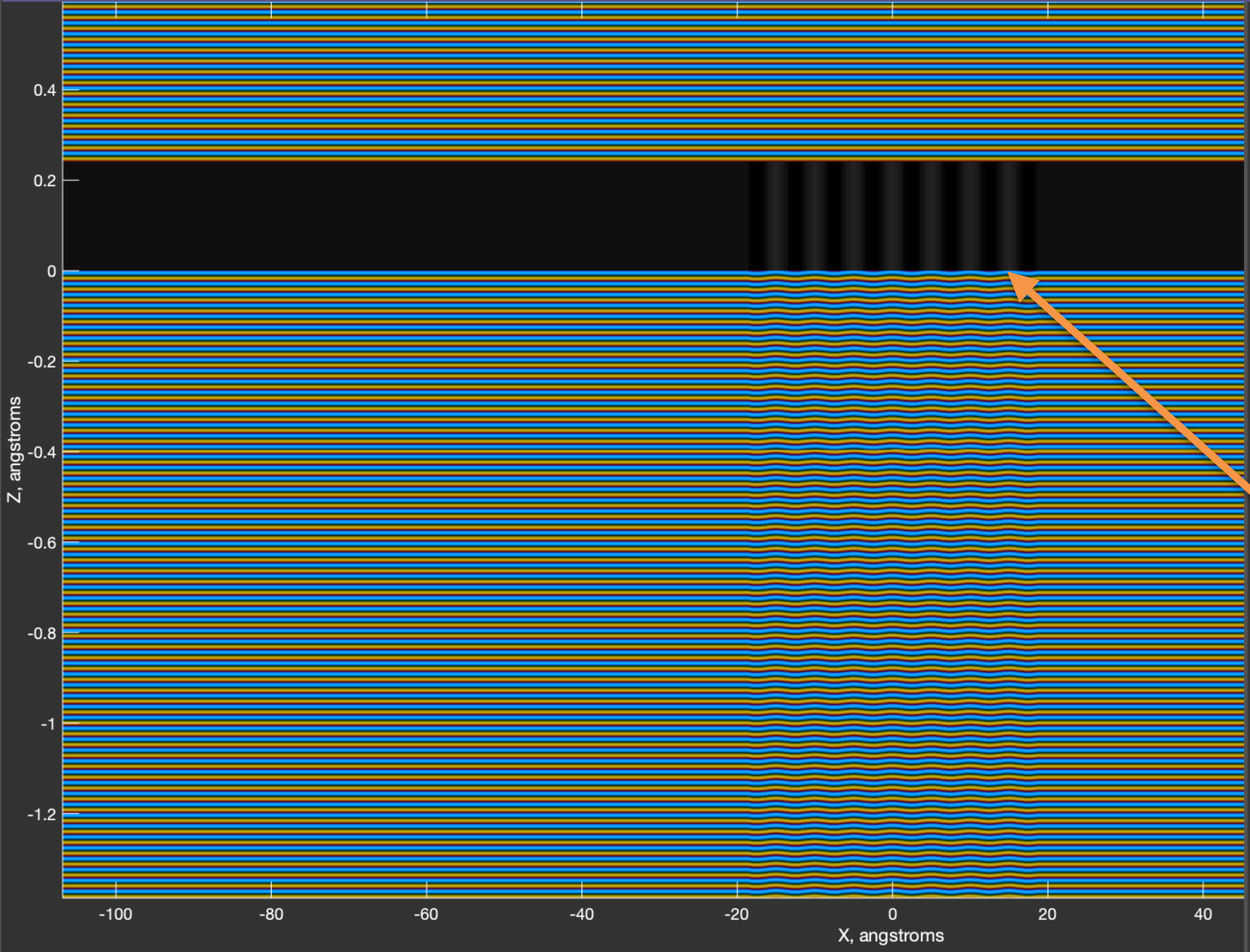
with

$$k = 2\pi/\lambda$$

Insert a phase-shifting object that perturbs the electron wave function



Insert a phase-shifting object that perturbs the electron wave function



The object is a grating,

$\epsilon\phi(x) = \epsilon\cos(2\pi x/d)$.

Example:

$d = 5$Å and $\epsilon \ll 1$.

At $z = 0$,

$\Psi = e^{i\epsilon\phi(x)}$

The weak-phase approximation

- Just below the specimen, at $z = 0$, the electron wave function is $\Psi = e^{i\epsilon\phi(x)}$.
- Then, by the approximation $e^x \approx 1 + x$ we have just after the specimen

$$\boxed{\Psi \approx 1 + i\epsilon\phi(x)}$$

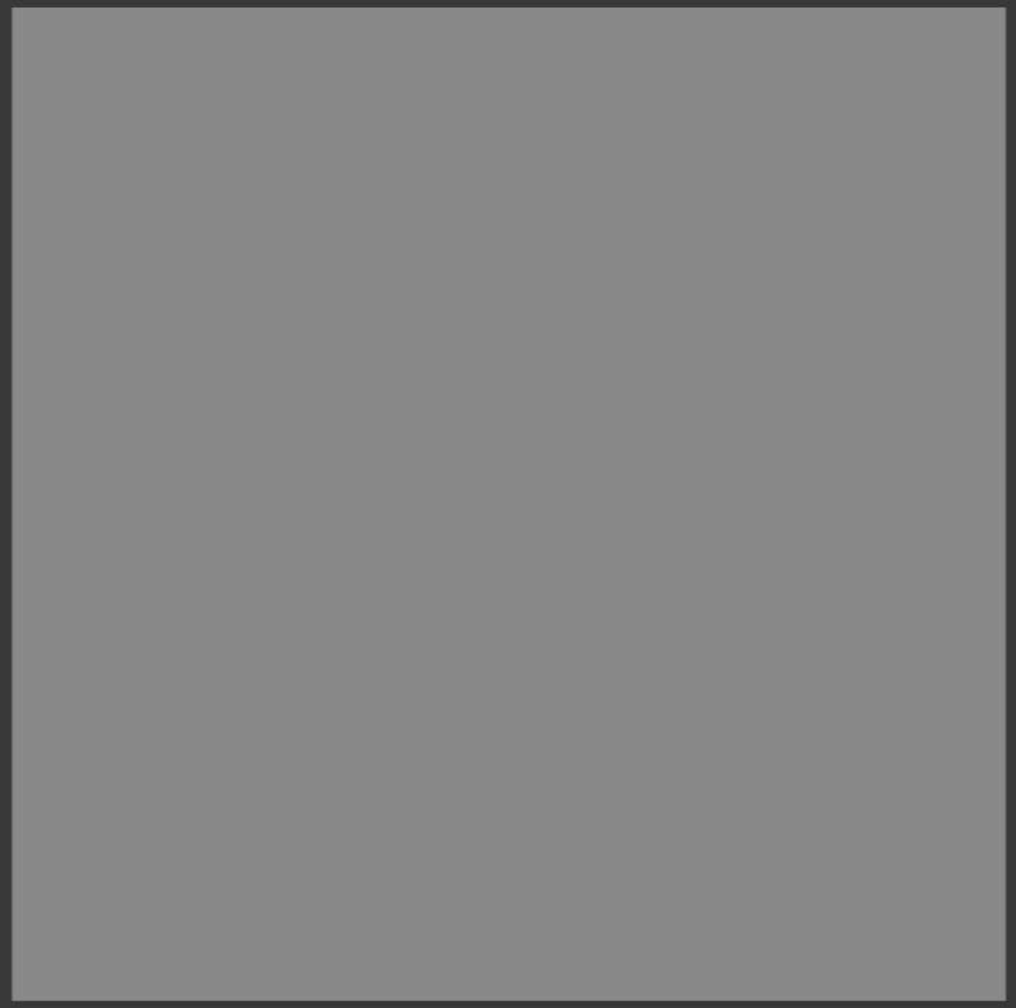
This is the weak phase approximation.

What are the two terms in the approximation?

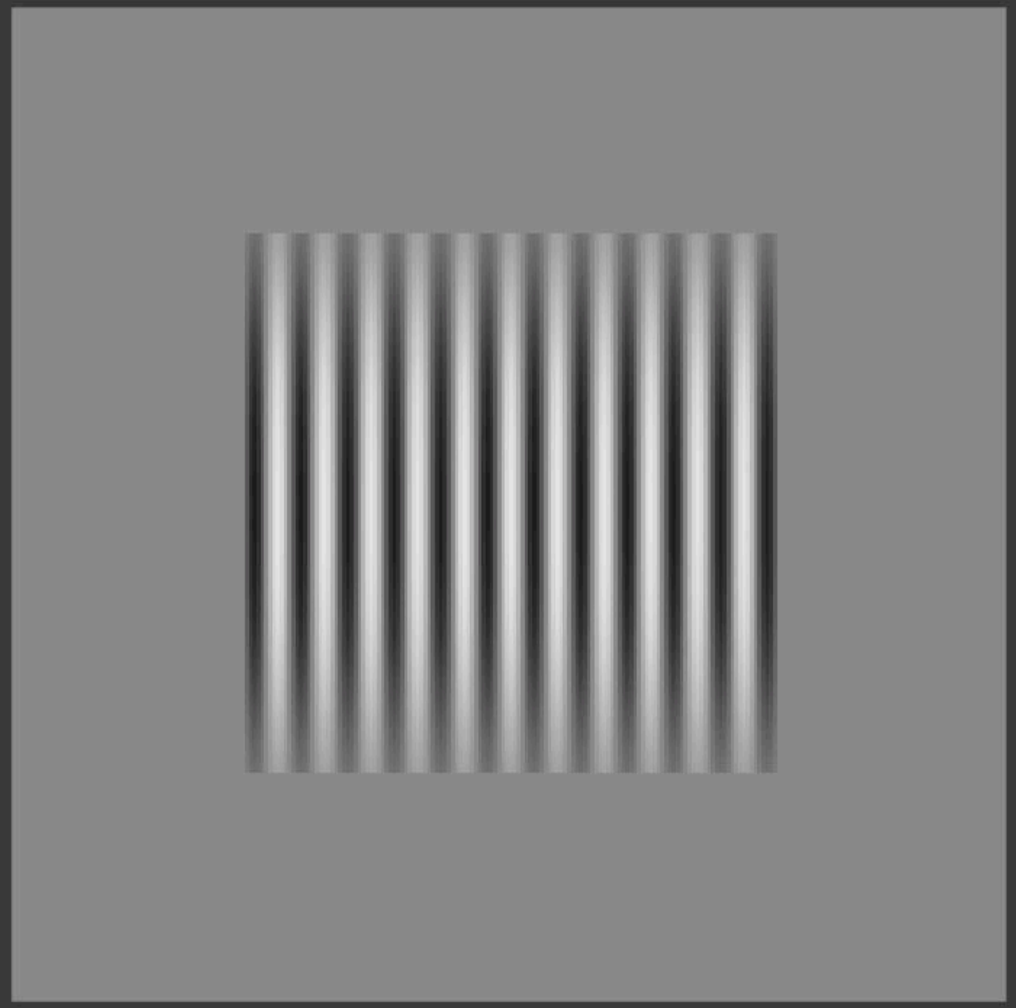
- There is an undiffracted wave —essentially the same as the incident wave—of amplitude 1. We'll call this Ψ_0
- And there is a new wave combination of amplitude ϵ. In this example of a grating there are actually two diffracted waves, Ψ_+ and Ψ_-
  - The full wavefunction is

$$\Psi = \Psi_0 + \Psi_+ + \Psi_-$$

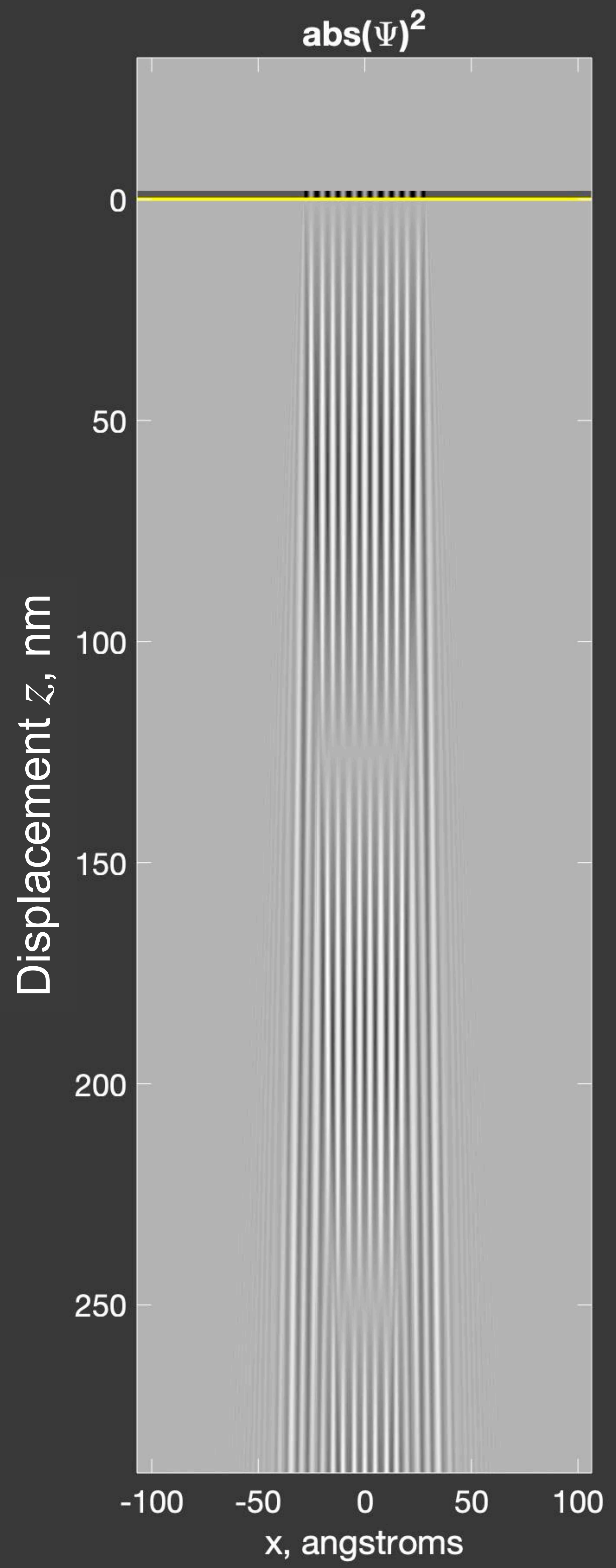
The contrast of a grating object varies with the distance below the object

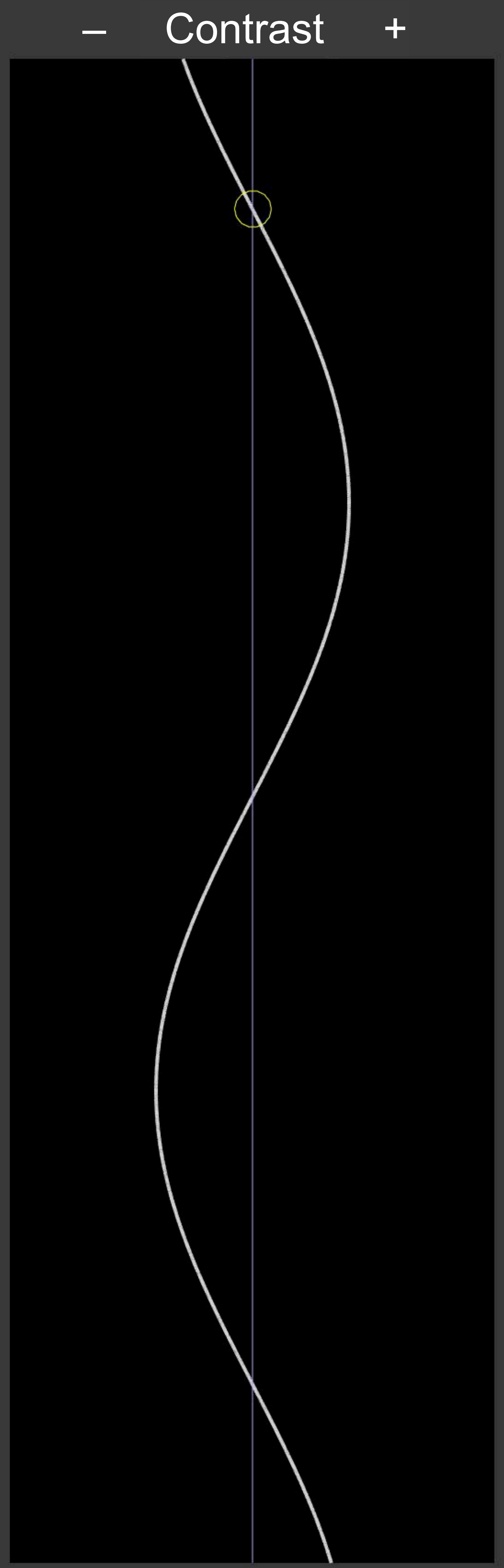


Intensity at z



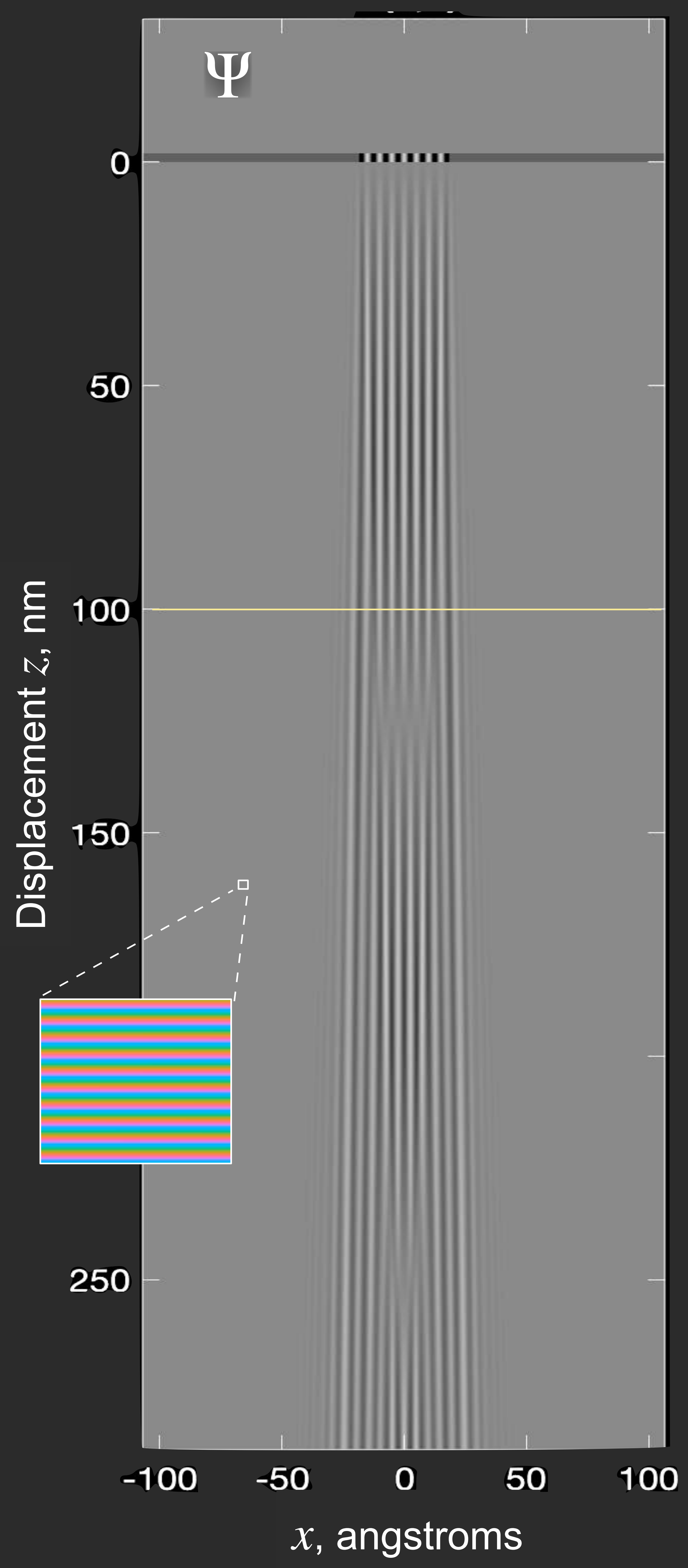
The grating $\phi(x)$

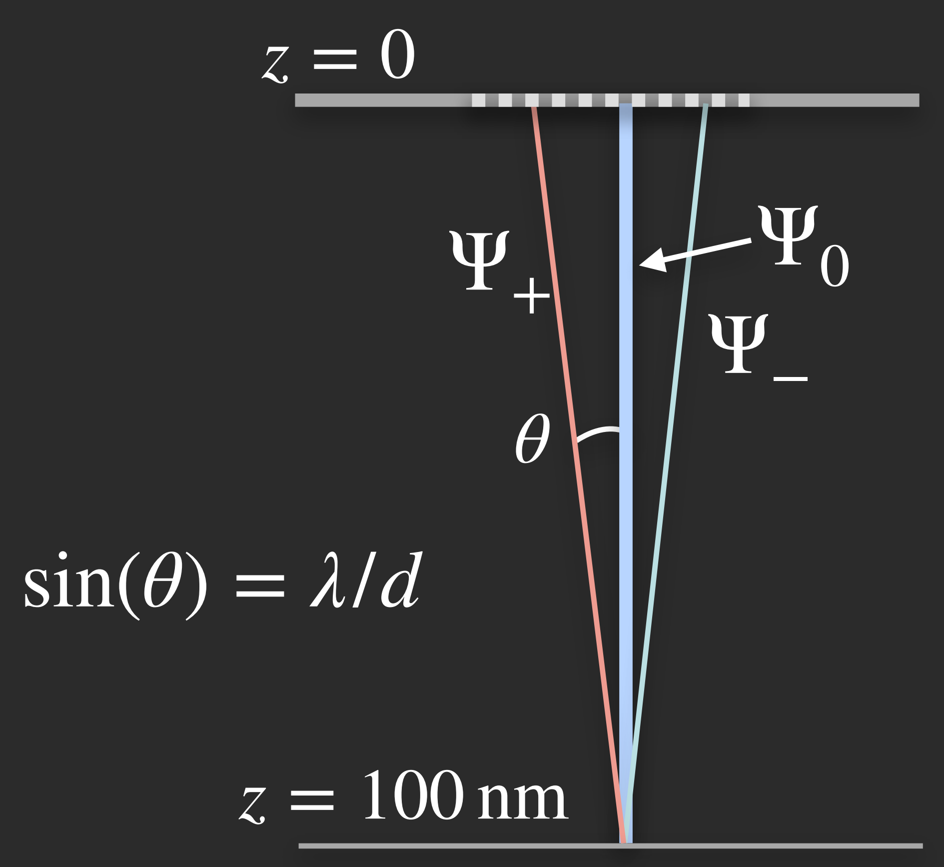




Interference between the undiffracted wave and diffracted waves produces contrast.

Waves interfere to make contrast





- The two diffracted waves Ψ_+ and Ψ_- travel at very small angles $+\theta$ and $-\theta$ to the undiffracted wave.
- To reach a distance z below the specimen, they take a path longer than Ψ_0 does. Let ζ = the path length difference.

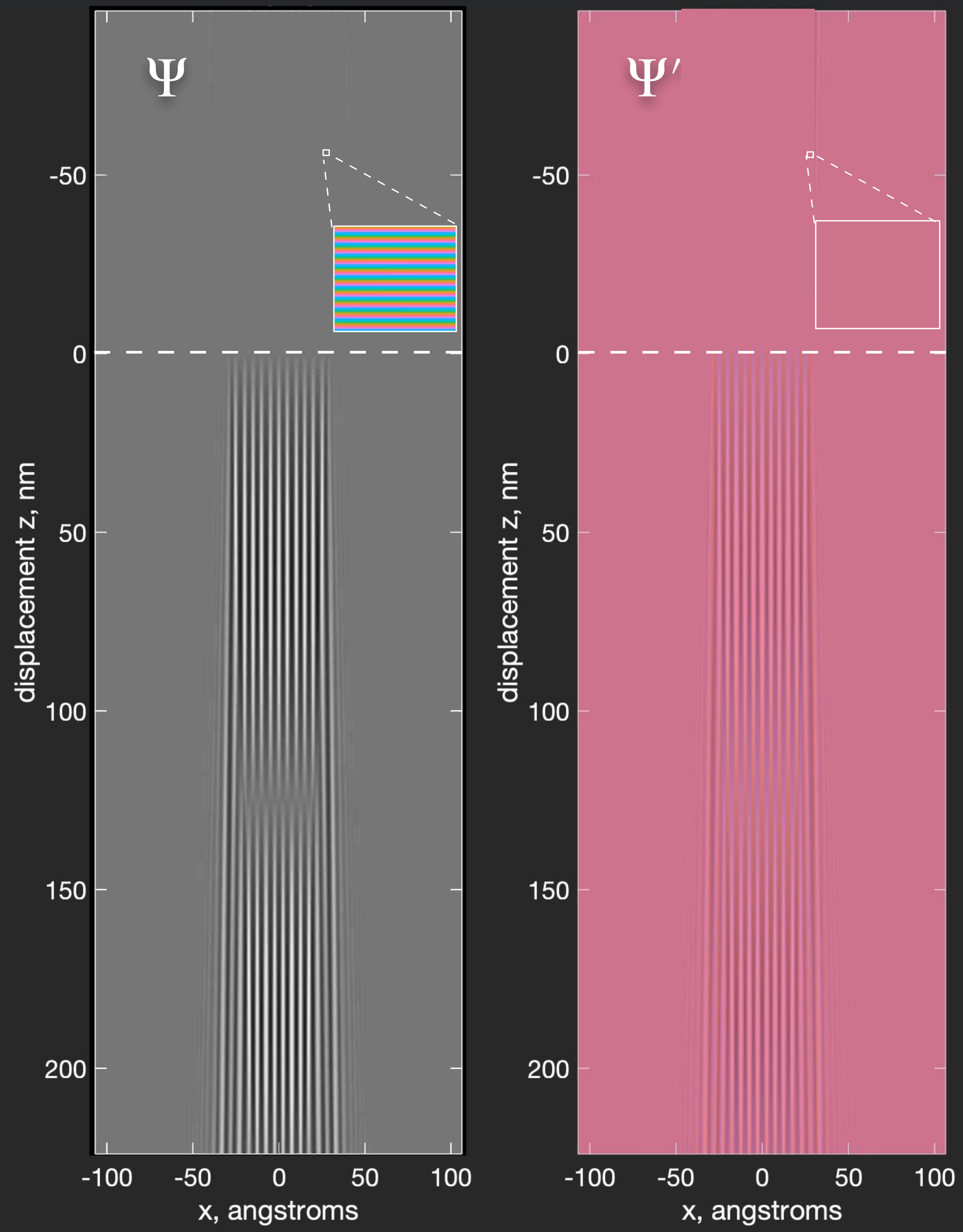
$$\zeta = \frac{z}{\cos\theta} - z \;\approx\; z\lambda^2/2d^2.$$

- In our example $\lambda = .02\text{Å}$ and the grating $d = 5\text{Å}$. At the level $z = 100\,\text{nm}$, $\zeta = .008\text{Å}$, about half a wavelength.
- Define χ = the phase difference between the undiffracted and diffracted waves.

$$\begin{aligned}\chi &= 2\pi\zeta/\lambda \\ &= \pi\lambda z/d^2\end{aligned}$$

- In this example $\chi = 0.8\pi$

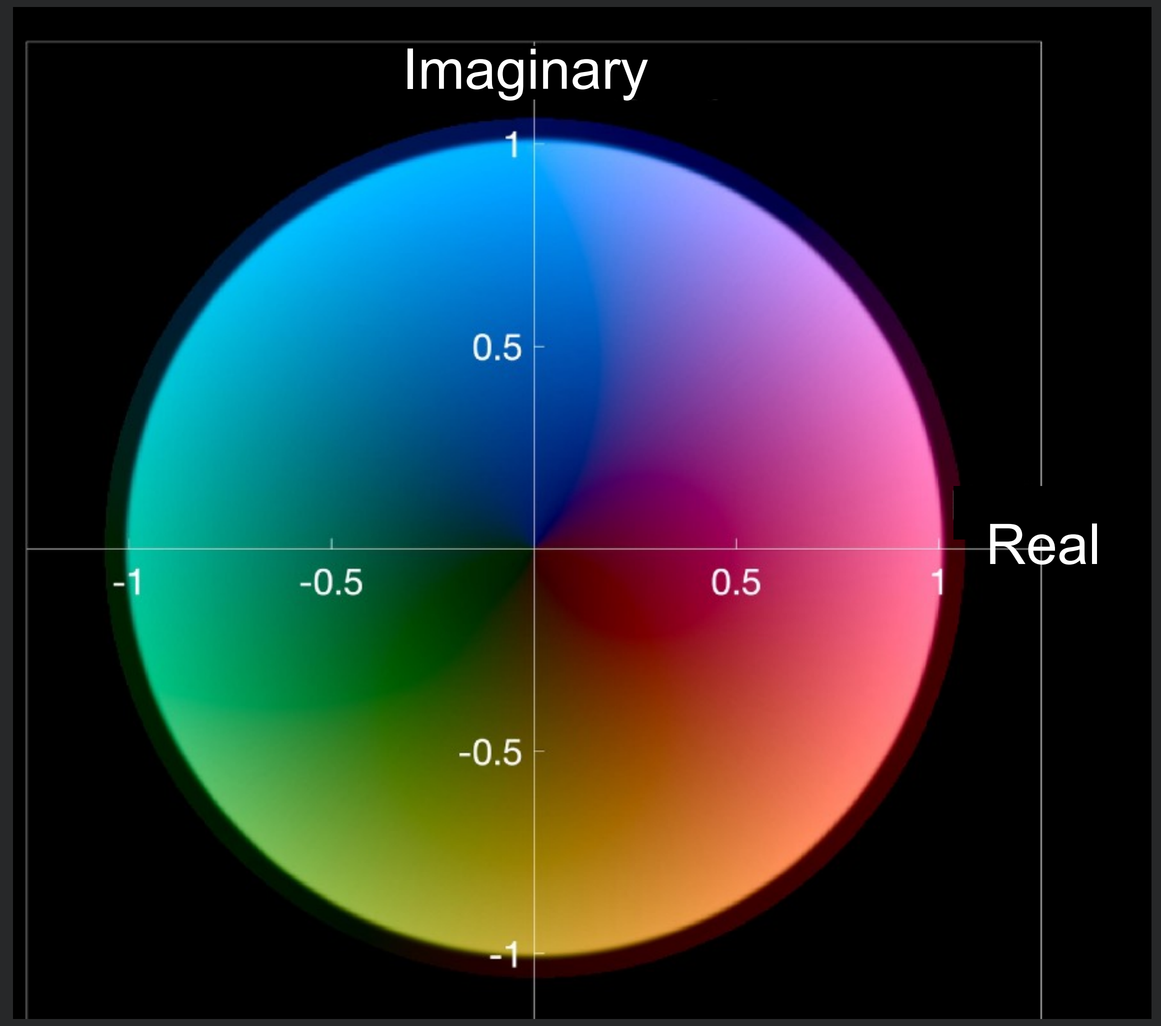
Where the phase of the diffracted waves is right, we have contrast.



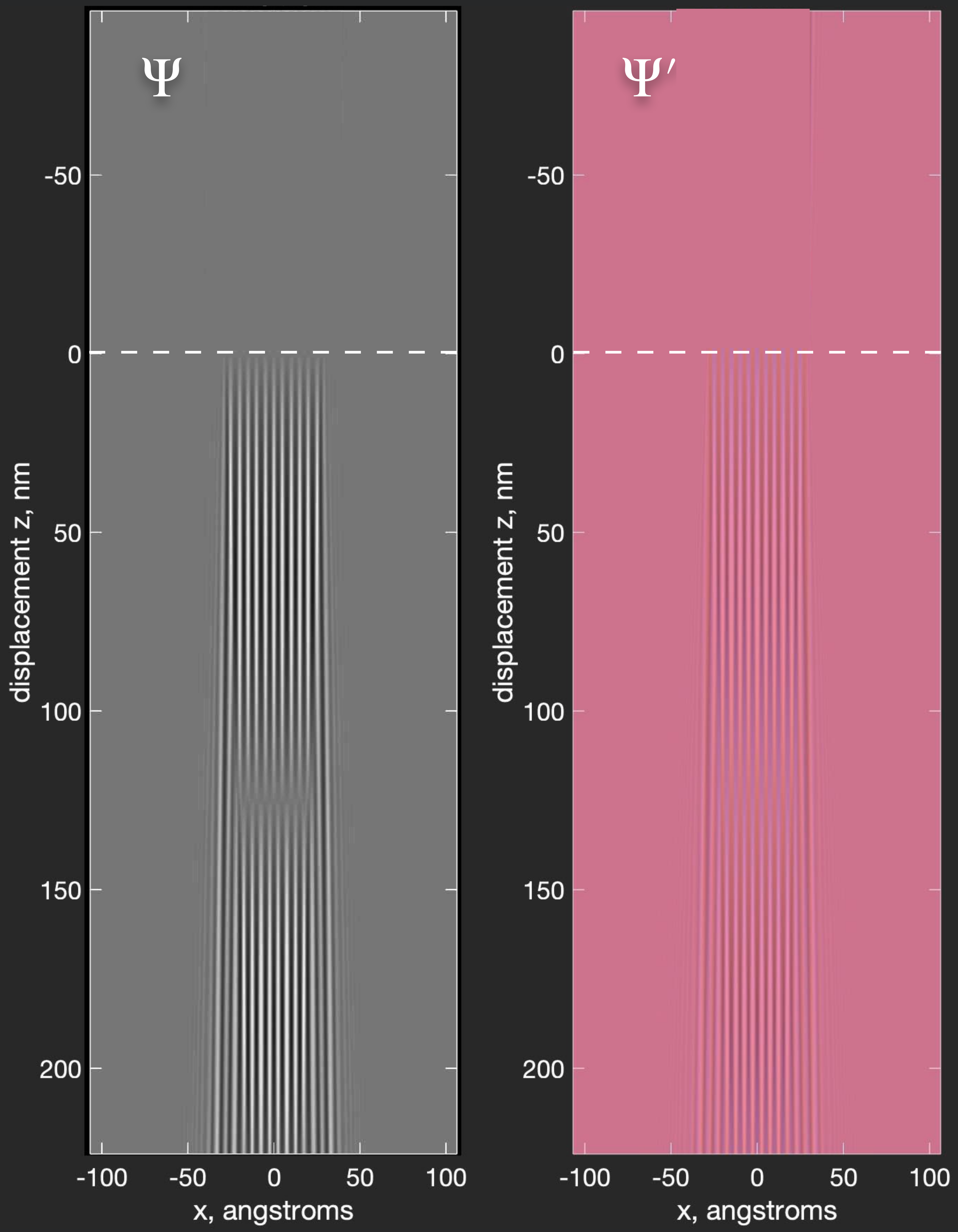
Let's unwrap the oscillations in Ψ:

We'll define $\Psi' = \Psi/\Psi_0$

Complex number color scheme



Where the phase of the diffracted waves is right, we have contrast.



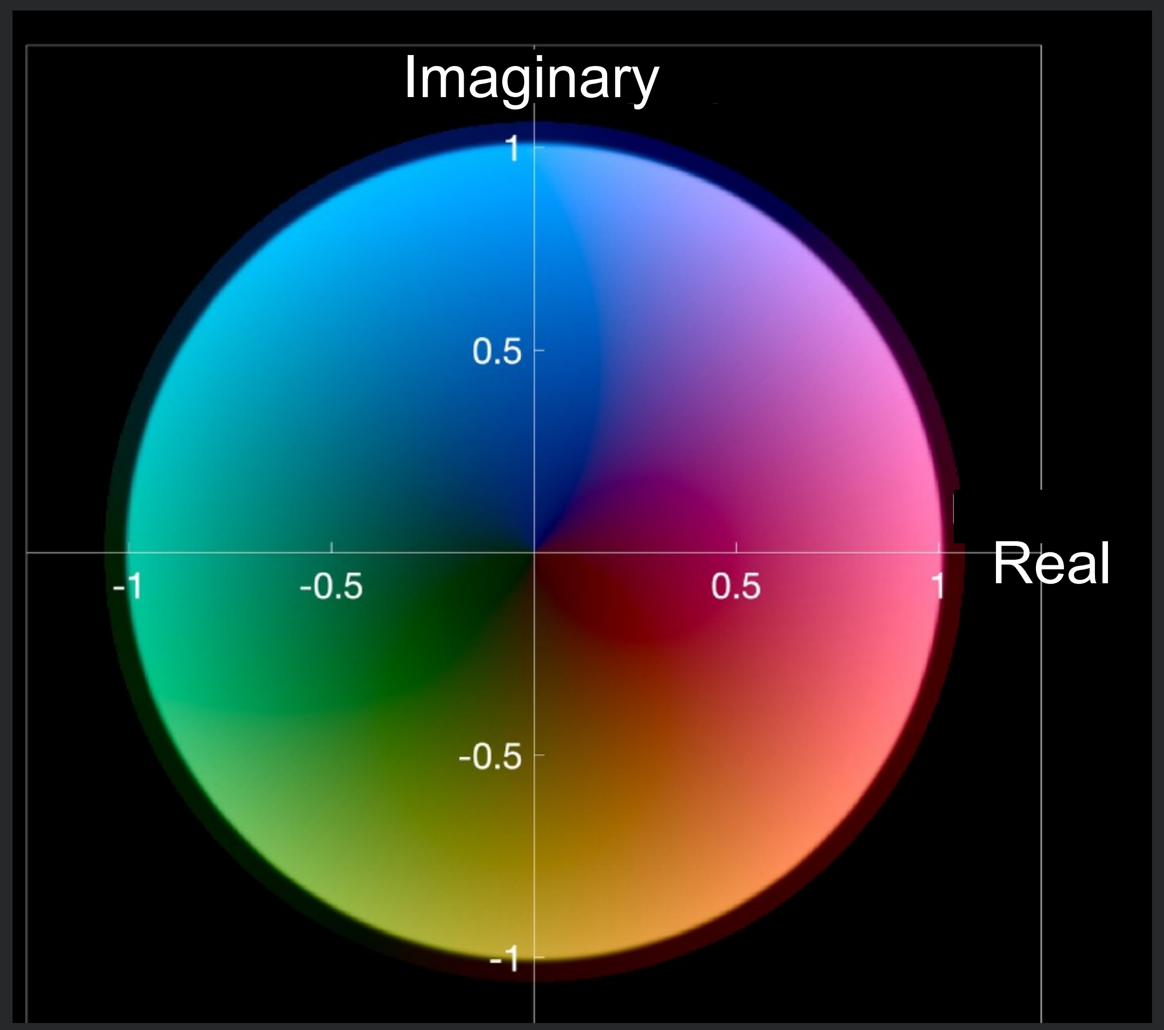
Let's unwrap the oscillations in Ψ:

We'll define $\Psi' = \Psi/\Psi_0$

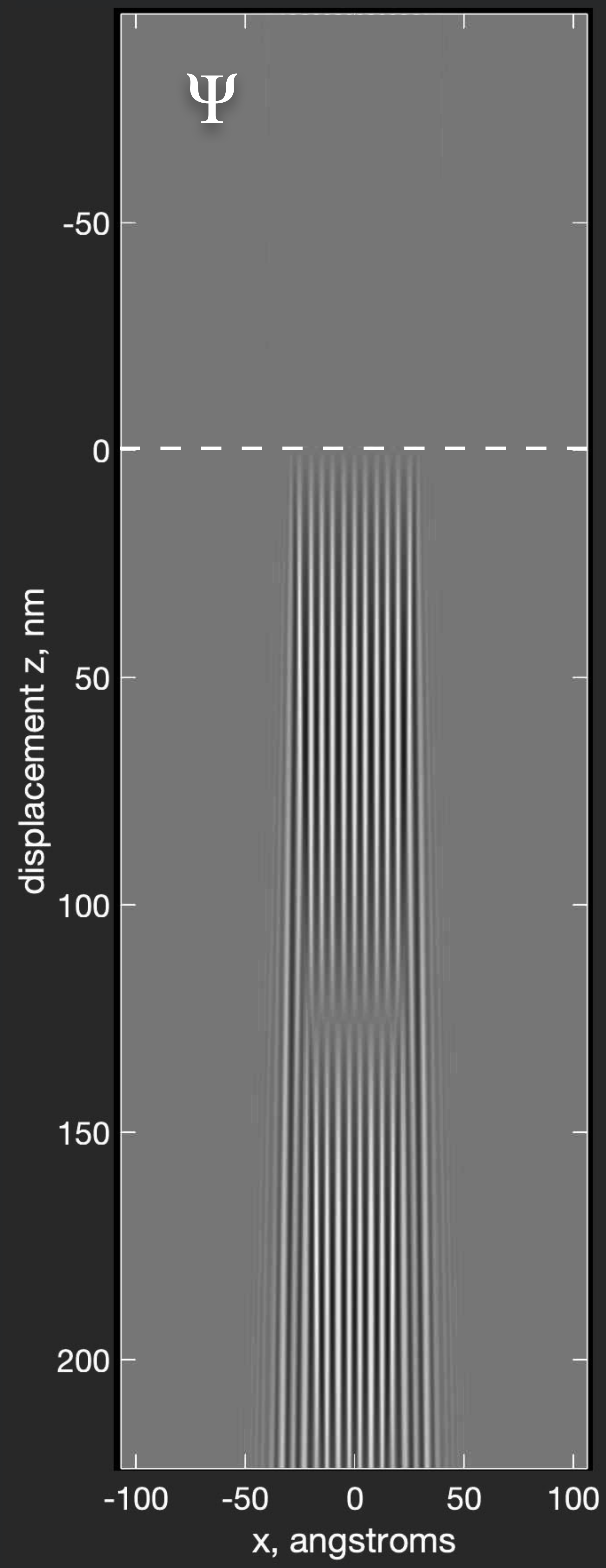
Let's remove the undiffracted wave, so we have just the diffracted waves,

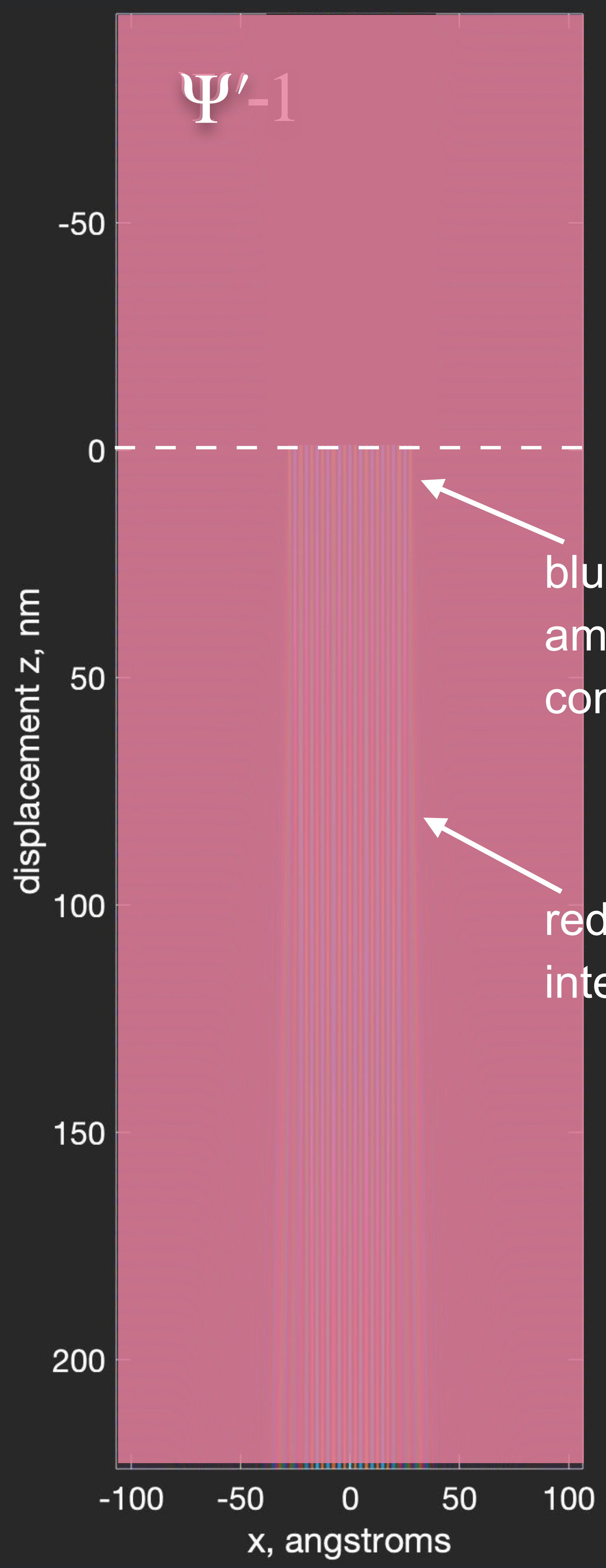
$(\Psi' - 1) = \Psi_+ + \Psi_-$

Complex number color scheme



When the phase of the diffracted waves is right, we have contrast.





blue / yellow: imaginary amplitude, no interference, no contrast

red / green: real amplitude; interference

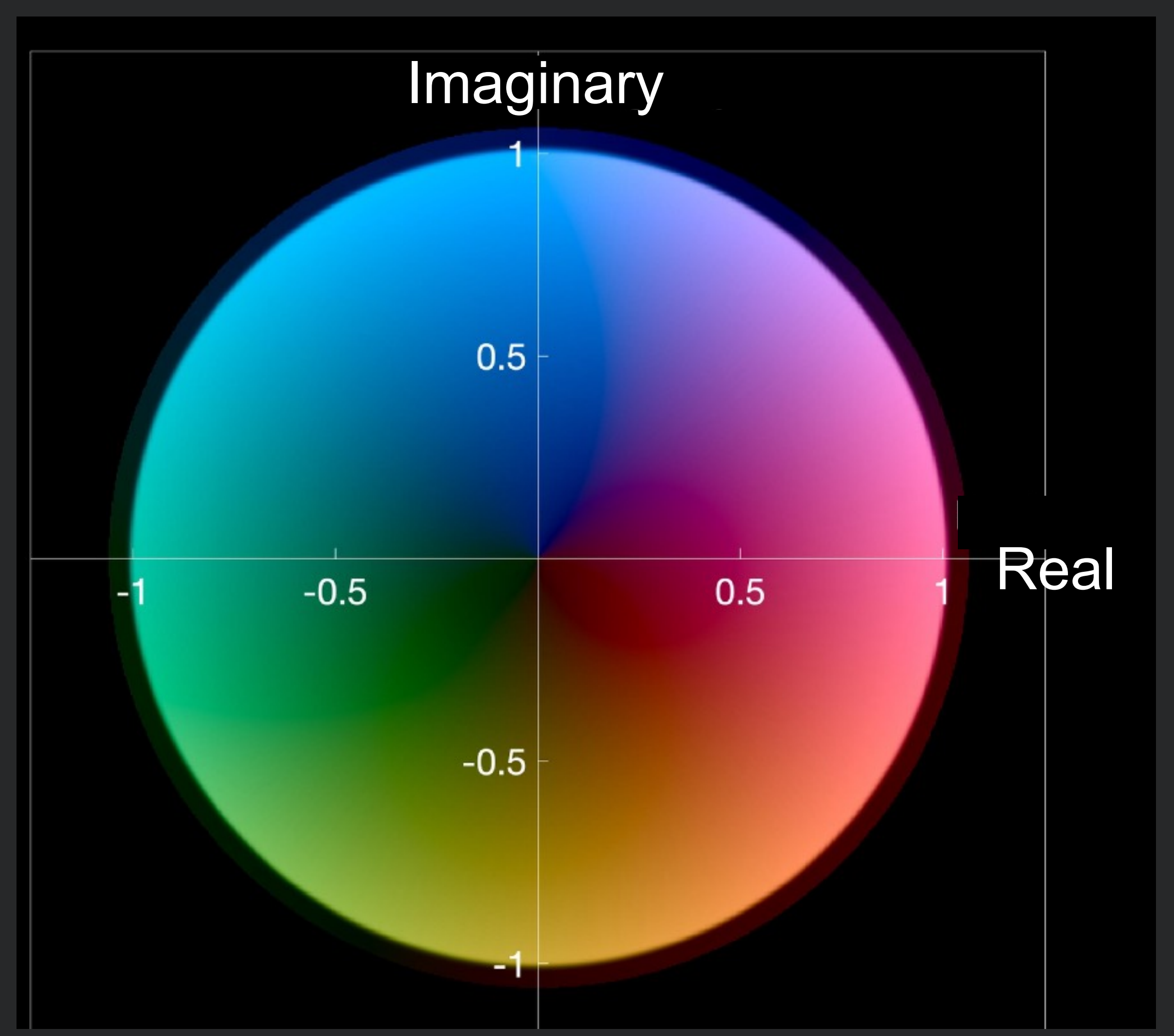
Let's unwrap the oscillations in Ψ:

We'll define $\Psi' = \Psi/\Psi_0$

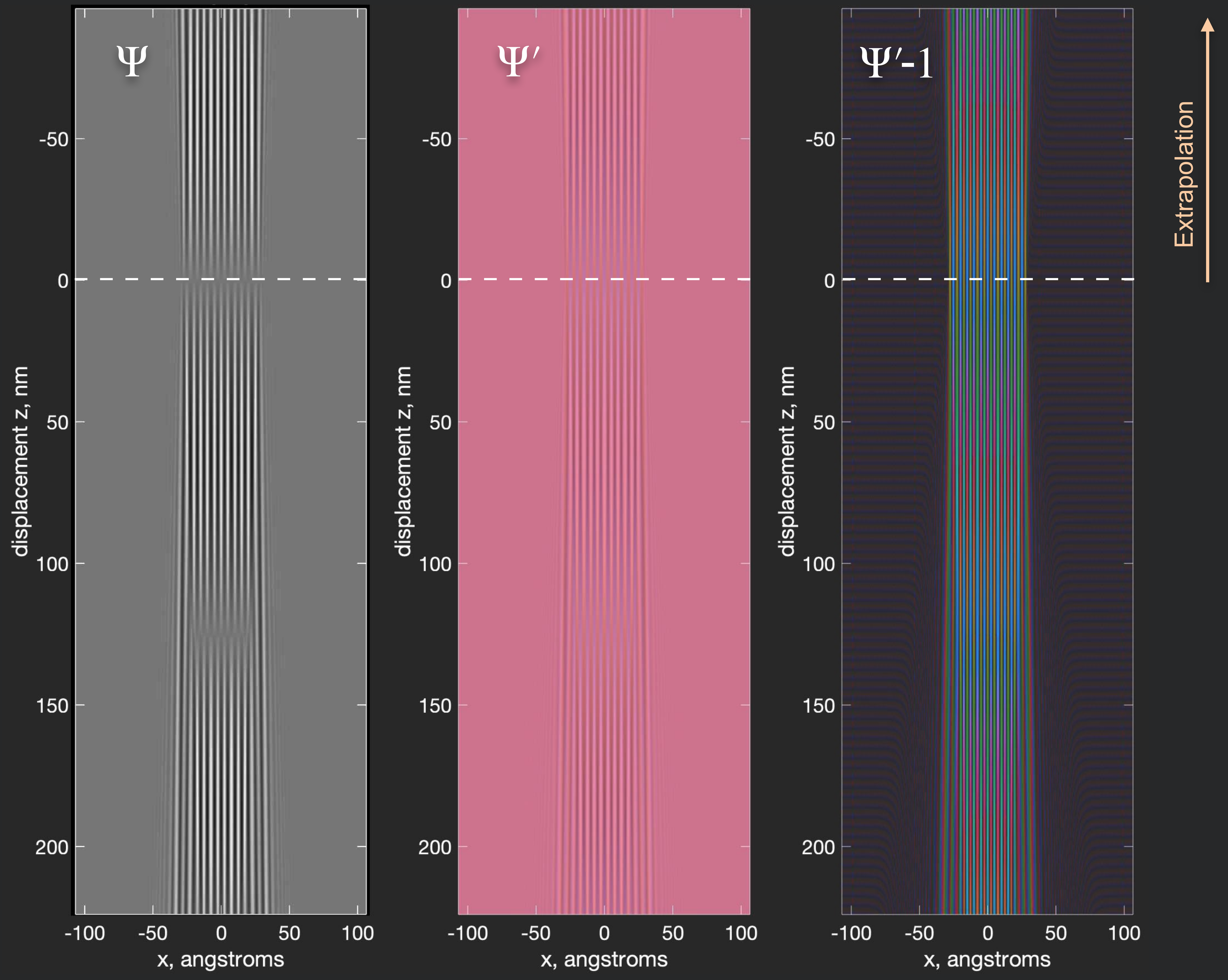
Let's remove the undiffracted wave, so we have just the diffracted waves,

$(\Psi' - 1) = \Psi_+ + \Psi_-$

Complex number color scheme



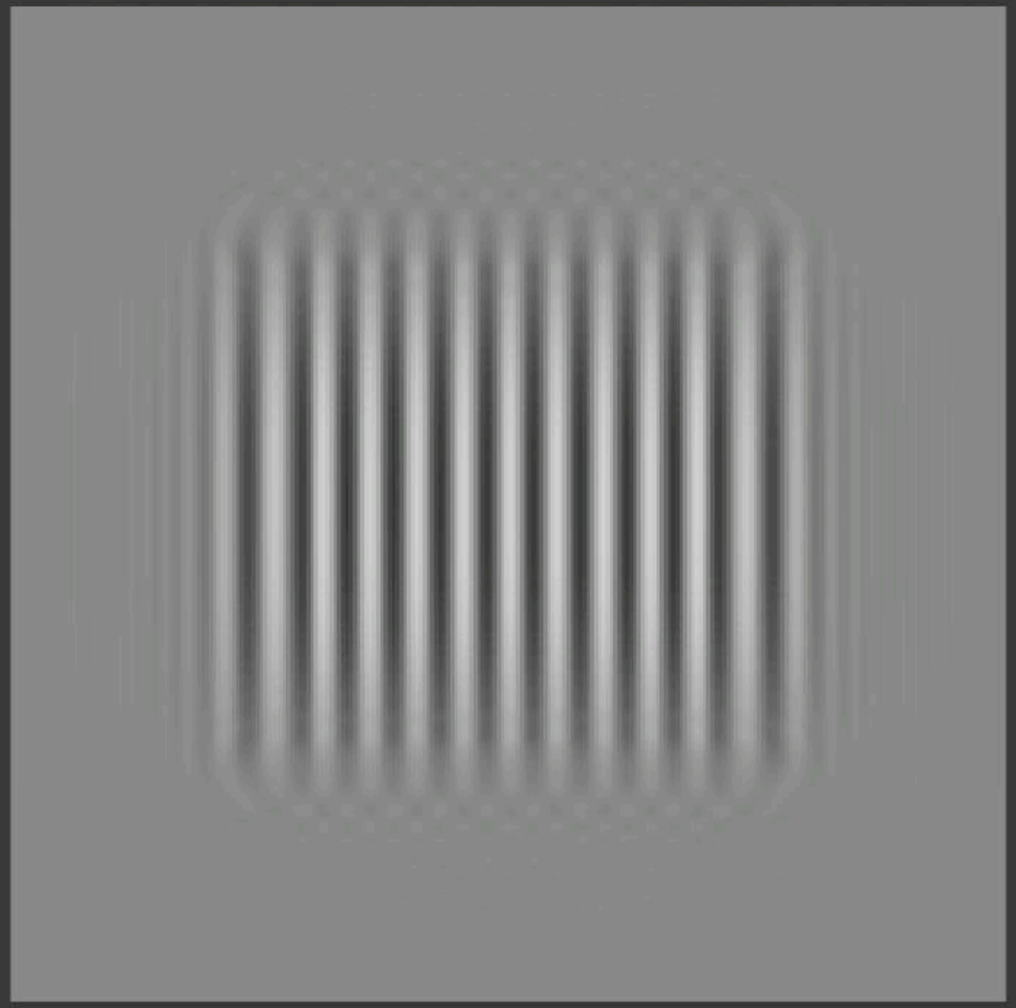
What happens when the objective lens is focused *above* the specimen?



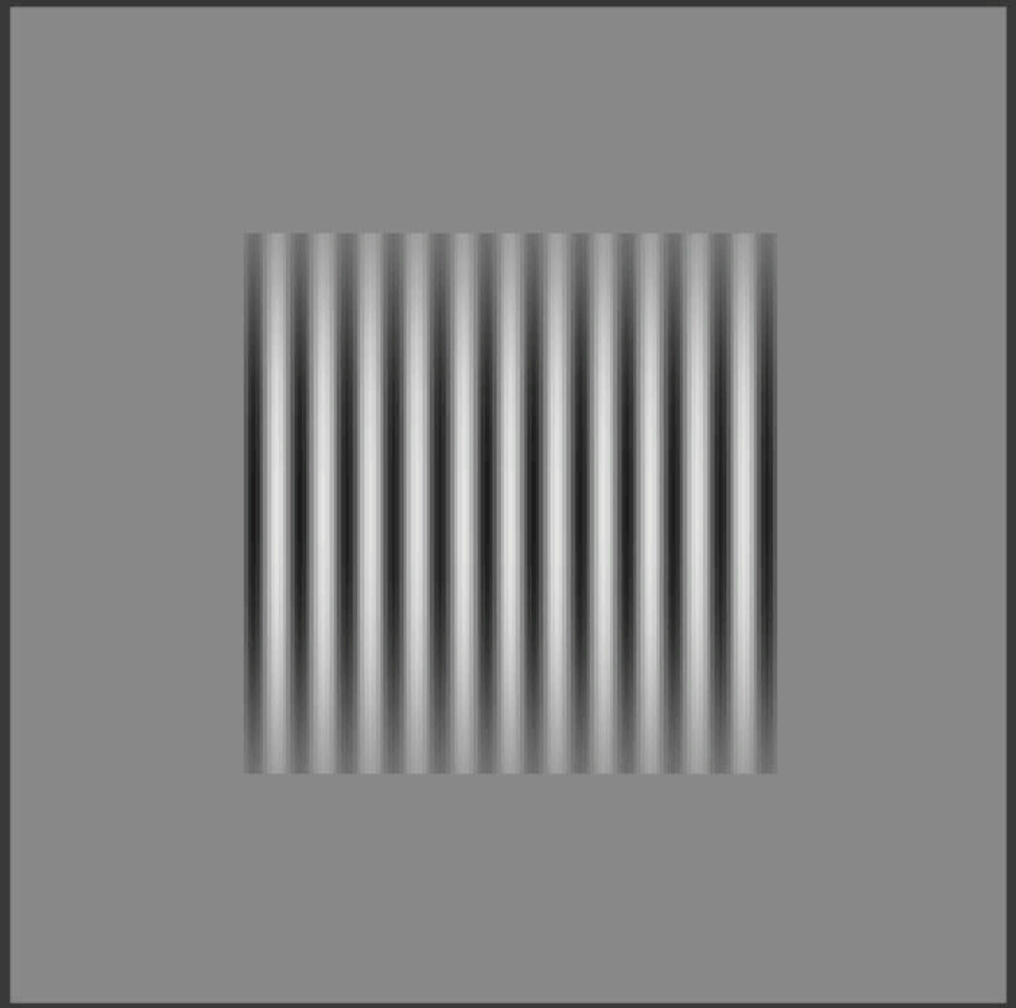
What wavefunction above the specimen would give rise to what we see below it?

We can back-propagate Ψ:
this is what the objective lens "sees"

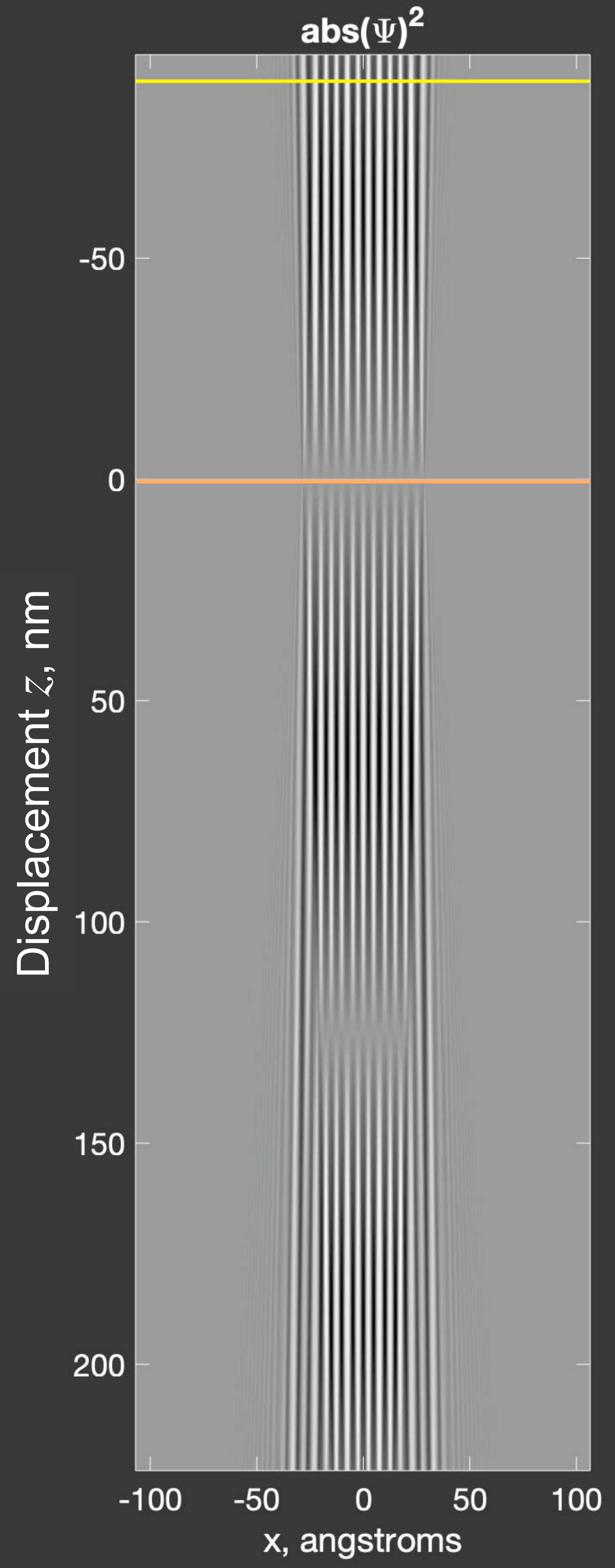
What happens when the objective lens is focused *above* the specimen?

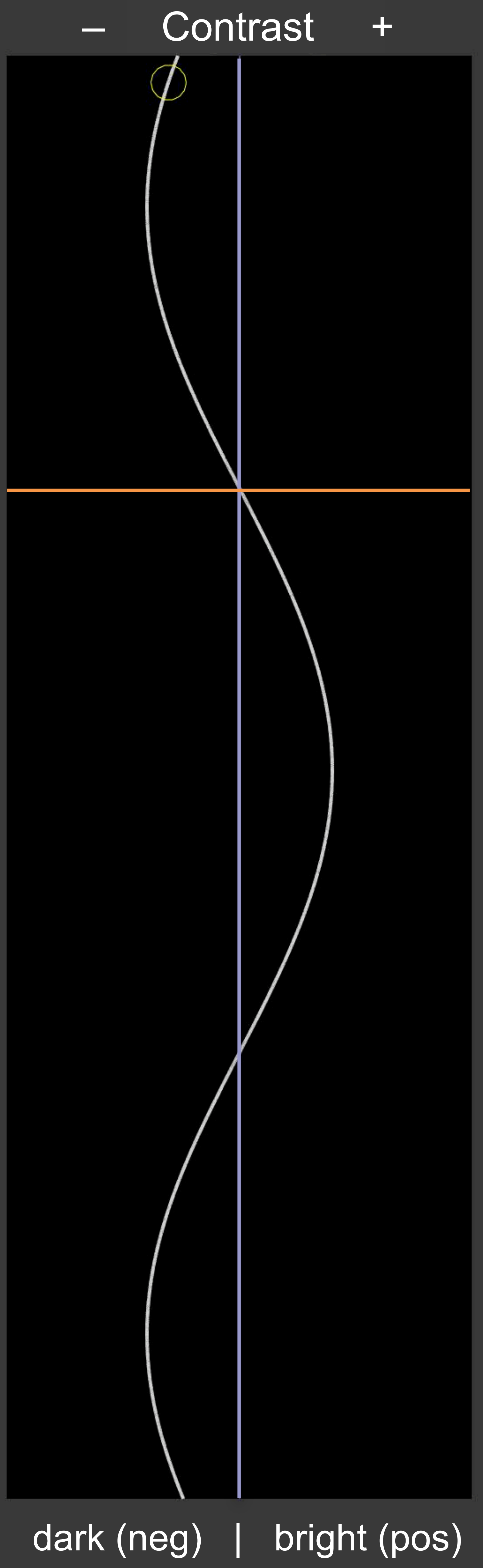


Intensity at z



The grating $\phi(x)$



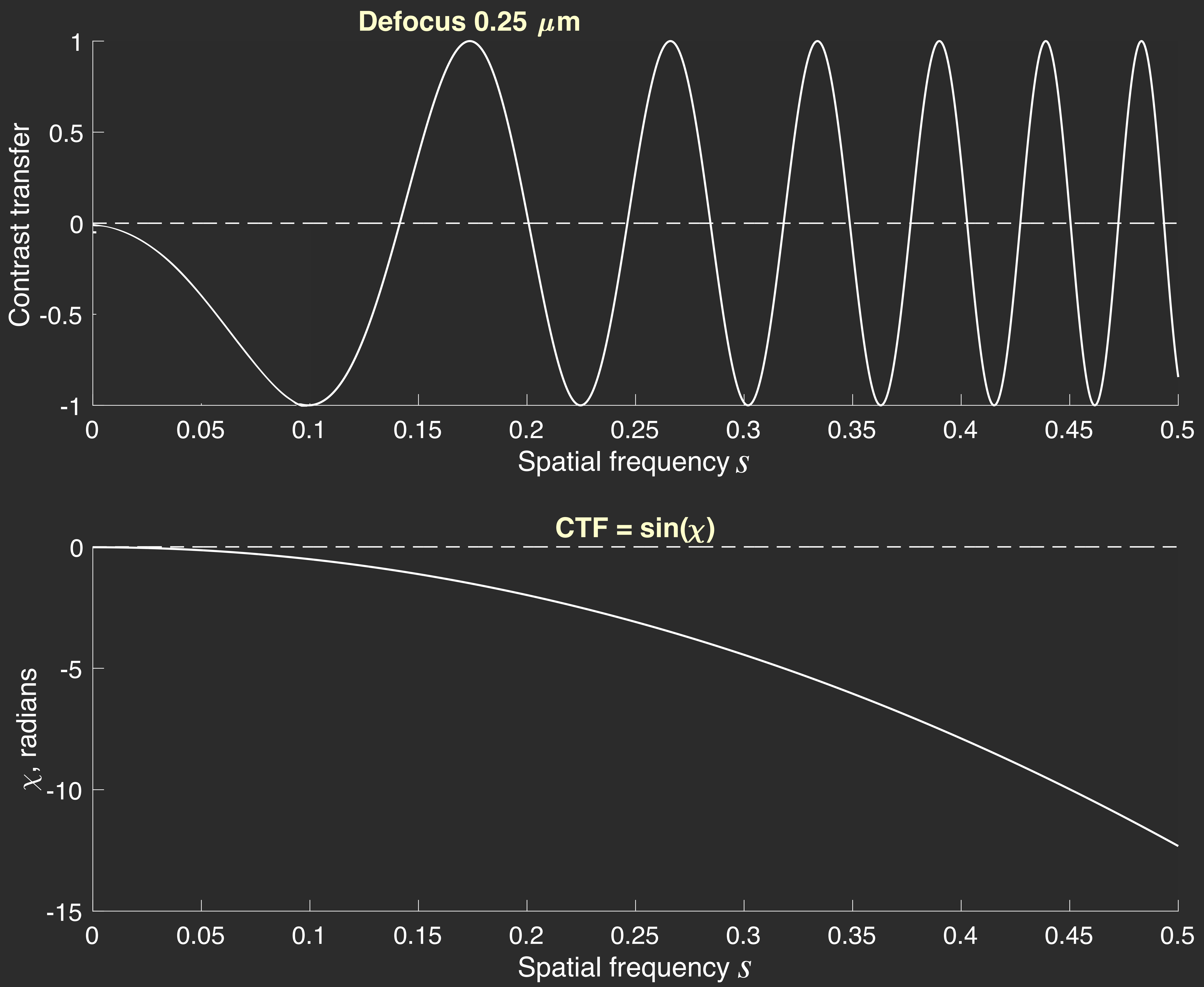


Terminology

- "Underfocus" is focusing the objective lens above the specimen.
- By convention, defocus values δ are positive for underfocus: $\delta = -z$
- Spatial frequency is $s = 1/d$
- The phase shift χ is proportional to δ.
- The contrast transfer function is given by

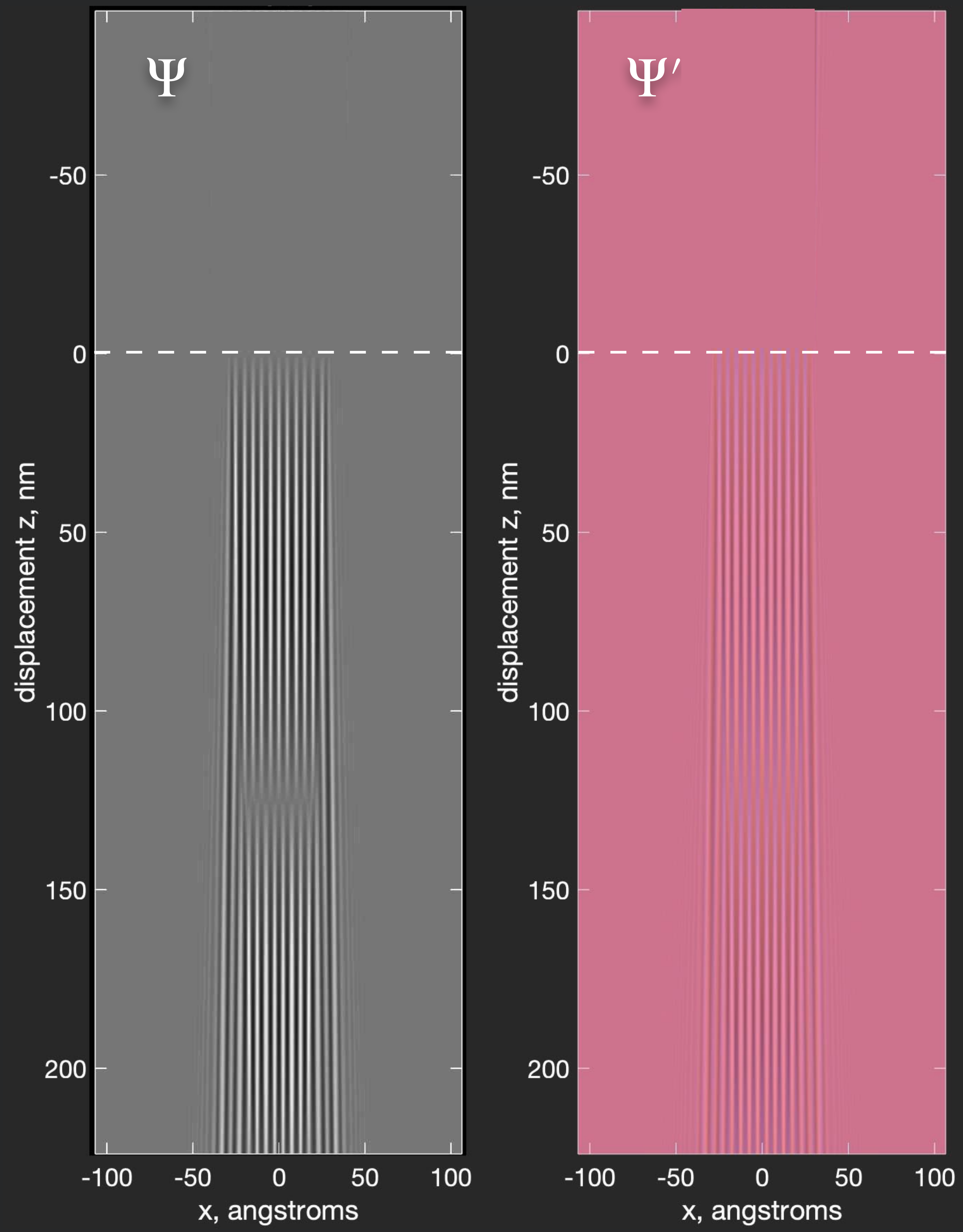
$$\boxed{\begin{aligned} \mathrm{CTF} &= \sin(\chi) \\ &= \sin(-\pi\lambda\delta\, s^2) \end{aligned}}$$

The basic contrast-transfer function as a function of s



$$\boxed{\mathrm{CTF} = \sin(-\pi\lambda\delta s^2)}$$

Formal derivation of the defocus-contrast CTF



We define

$$\Psi' = \Psi / \Psi_0$$

$$\Psi' = 1 + ie^{-ik\zeta} \cdot \epsilon \cos(2\pi x/d)$$

and can be written as

$$\Psi' = 1 + ie^{-i\chi}\,\epsilon\phi(x).$$

Grating object:

$$\epsilon\phi(x) = \epsilon\cos(2\pi x/d)$$

Electron propagation:

$$k = 2\pi/\lambda$$

Diffracted wave path difference:

$$\zeta \approx z\lambda^2/2d^2$$

Wave aberration function:

$$\chi = k\zeta \approx \pi\lambda z/d^2$$

Formal derivation of the CTF for a grating of spacing d

$$\Psi' = 1 + ie^{-i\chi}\,\epsilon\phi(x).$$

The measured intensity is

$$\begin{aligned}
|\Psi|^2 = |\Psi'|^2 &= (\text{real part})^2 + (\text{imag part})^2 \\
&= \big[1 + \sin(\chi)\,\epsilon\phi(x)\big]^2 + \big[\cos(-\chi)\,\epsilon\phi(x)\big]^2 \\
&= \big[\boxed{1 + 2\sin(\chi)\,\epsilon\phi(x)} + \mathcal{O}\epsilon^2\big] + \big[\mathcal{O}\epsilon^2\big].
\end{aligned}$$

Grating object:

$$\epsilon\phi(x) = \epsilon\cos(2\pi x/d)$$

Wave aberration function:

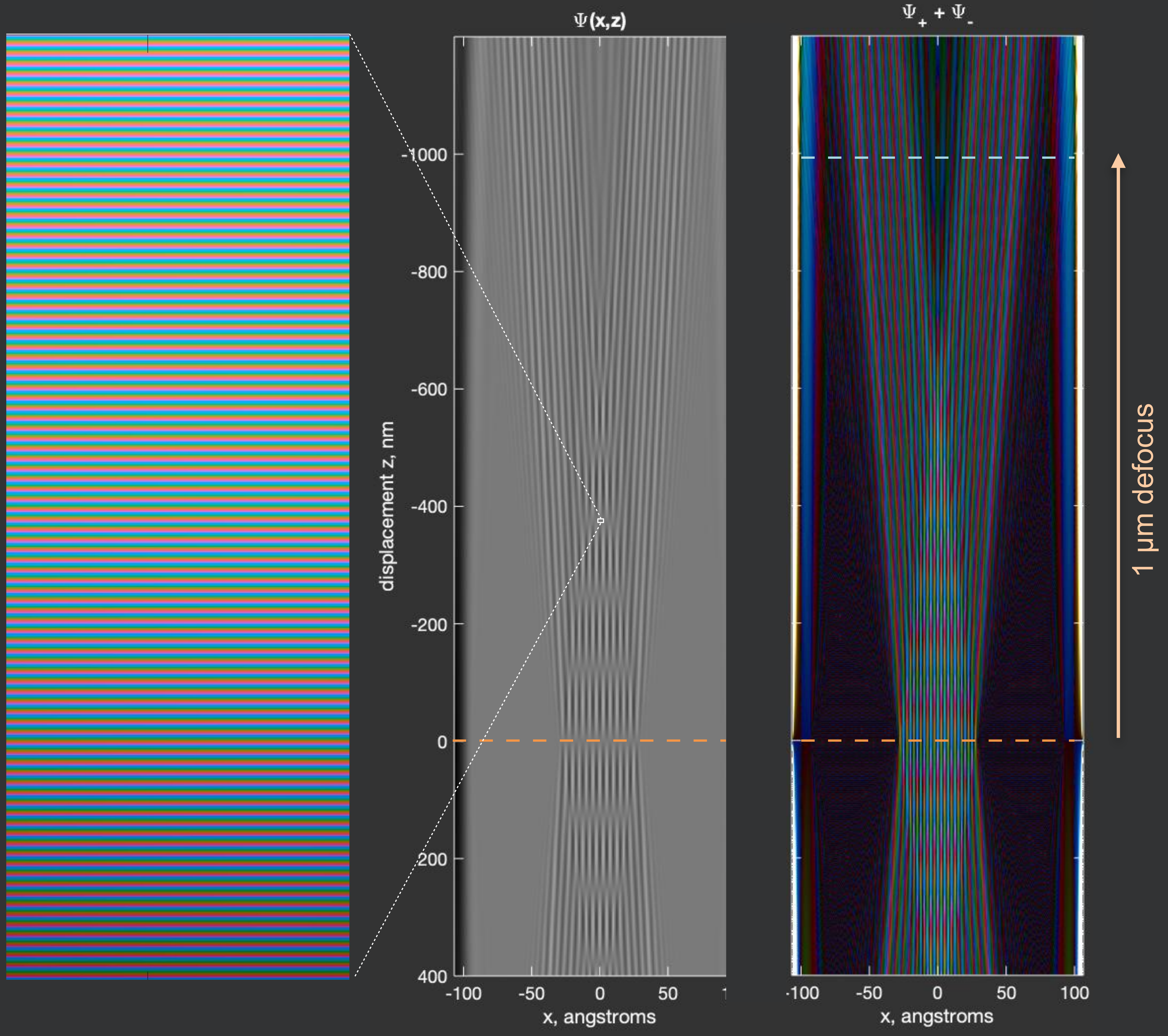
$$\chi \approx \pi\lambda z/d^2$$

<u>In practice</u>

- We ignore the constant background intensity.
- Everyone ignores the factor of 2 also.
- So we say the transfer from phase shift to intensity change is

$$\boxed{\text{CTF} = \frac{\Delta\text{Intensity}}{\Delta\text{Electron phase}} = \sin(\chi)}$$

A little defocus is actually a long distance

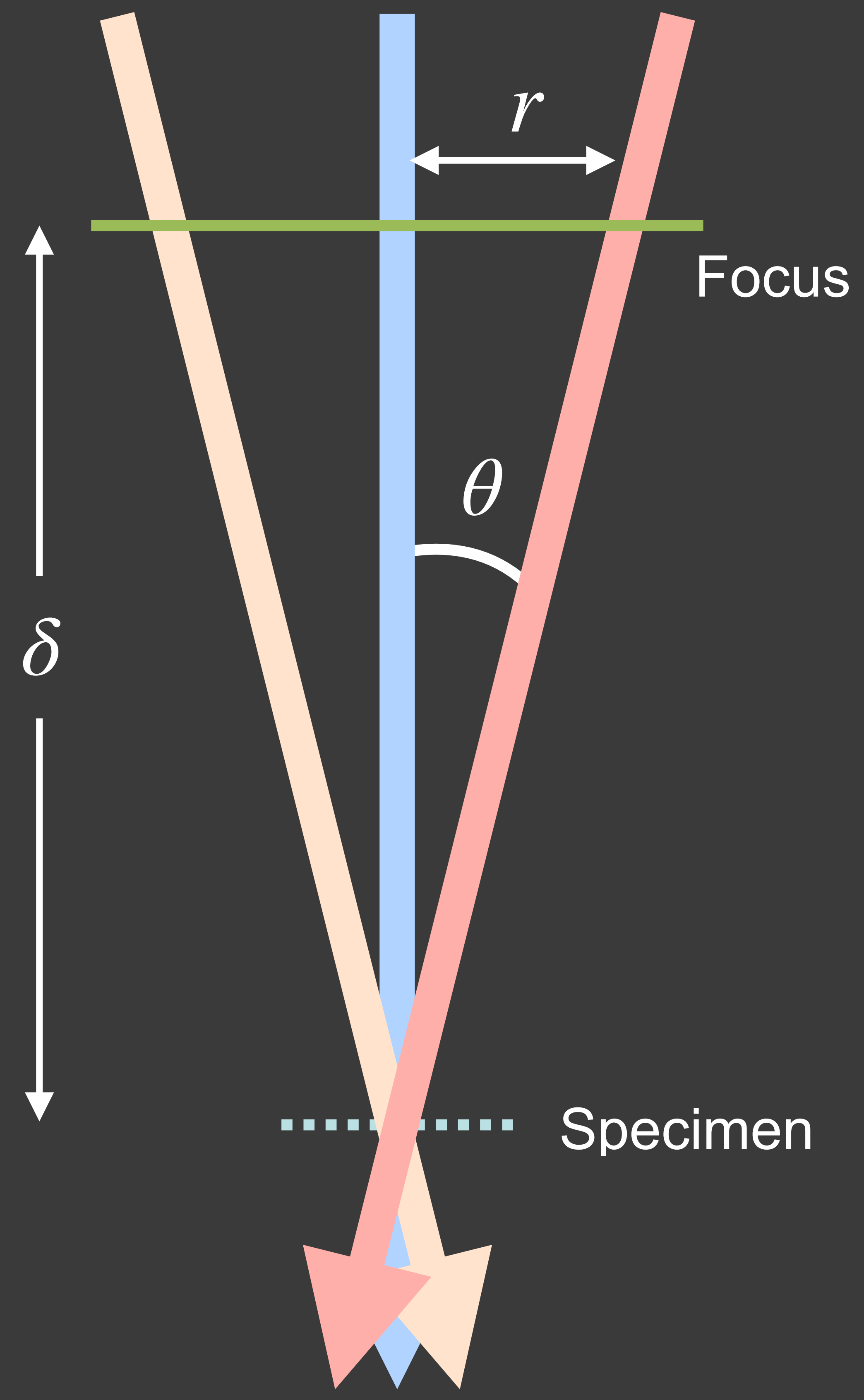


1 µm—a moderate defocus for cryo-EM imaging–is 500,000 wavelengths!

This has ramifications regarding

- beam coherence
- specimen charging
- delocalization

With large defocus, how bad is the image delocalization?

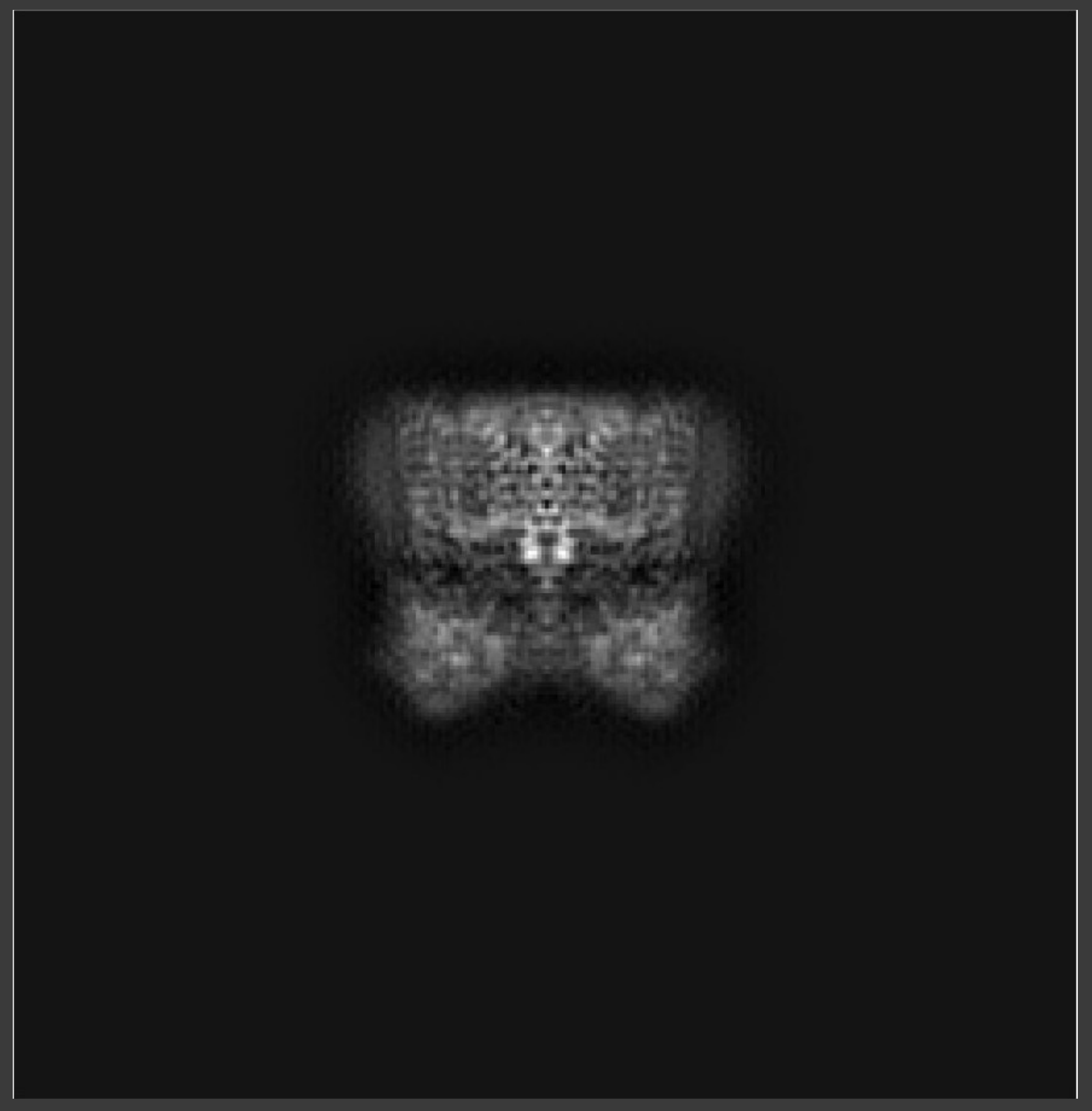


The dispersion radius is given by

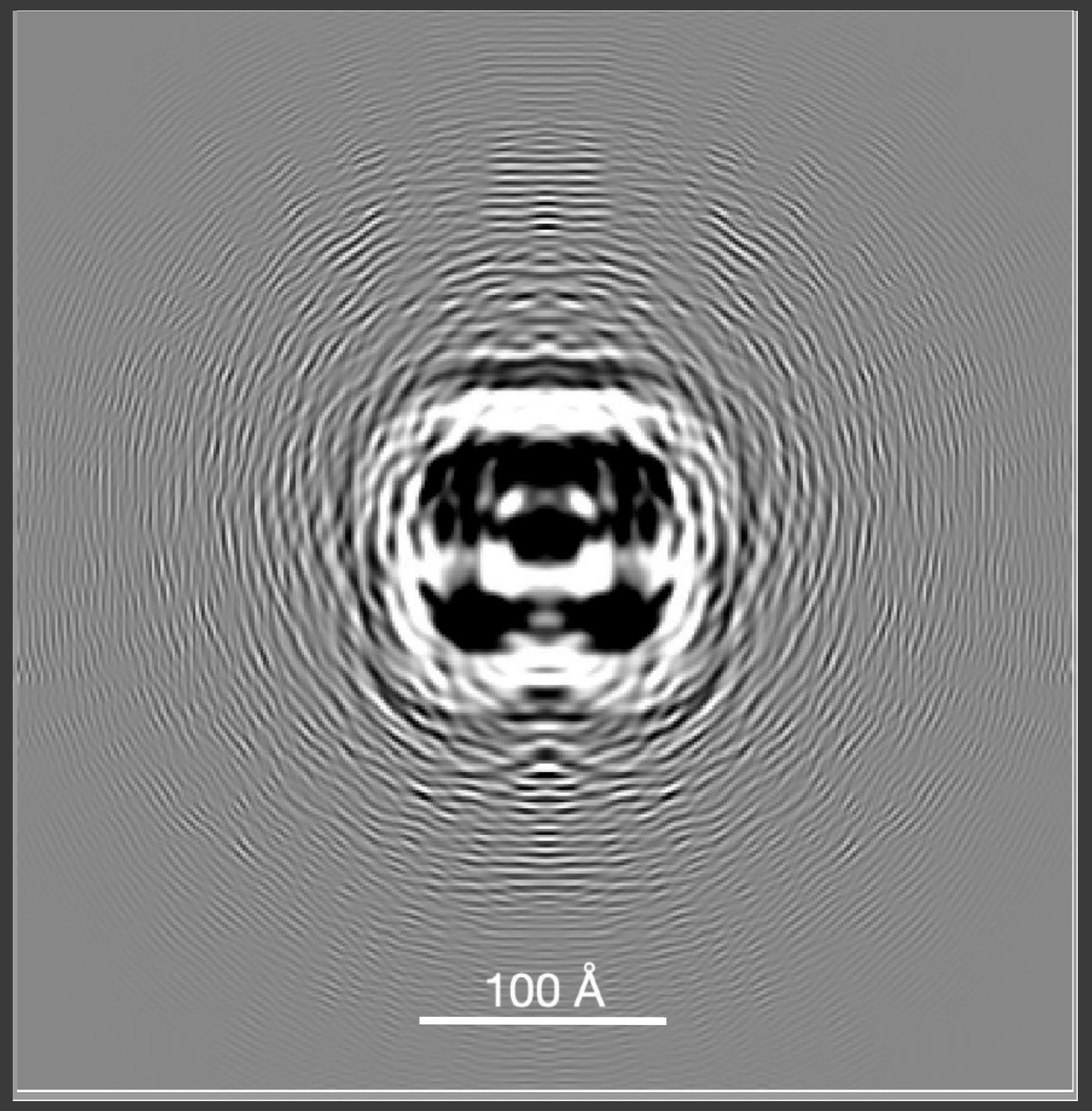
$$r = \delta \tan \theta$$
$$= \delta \lambda s \quad \text{(small angle approx.)}$$

Homework problem:

- How how much space do I need around my particle to include all the information up to 3Å, if I use 3μm of defocus?
- How much space would I need for 1.5μm of defocus?

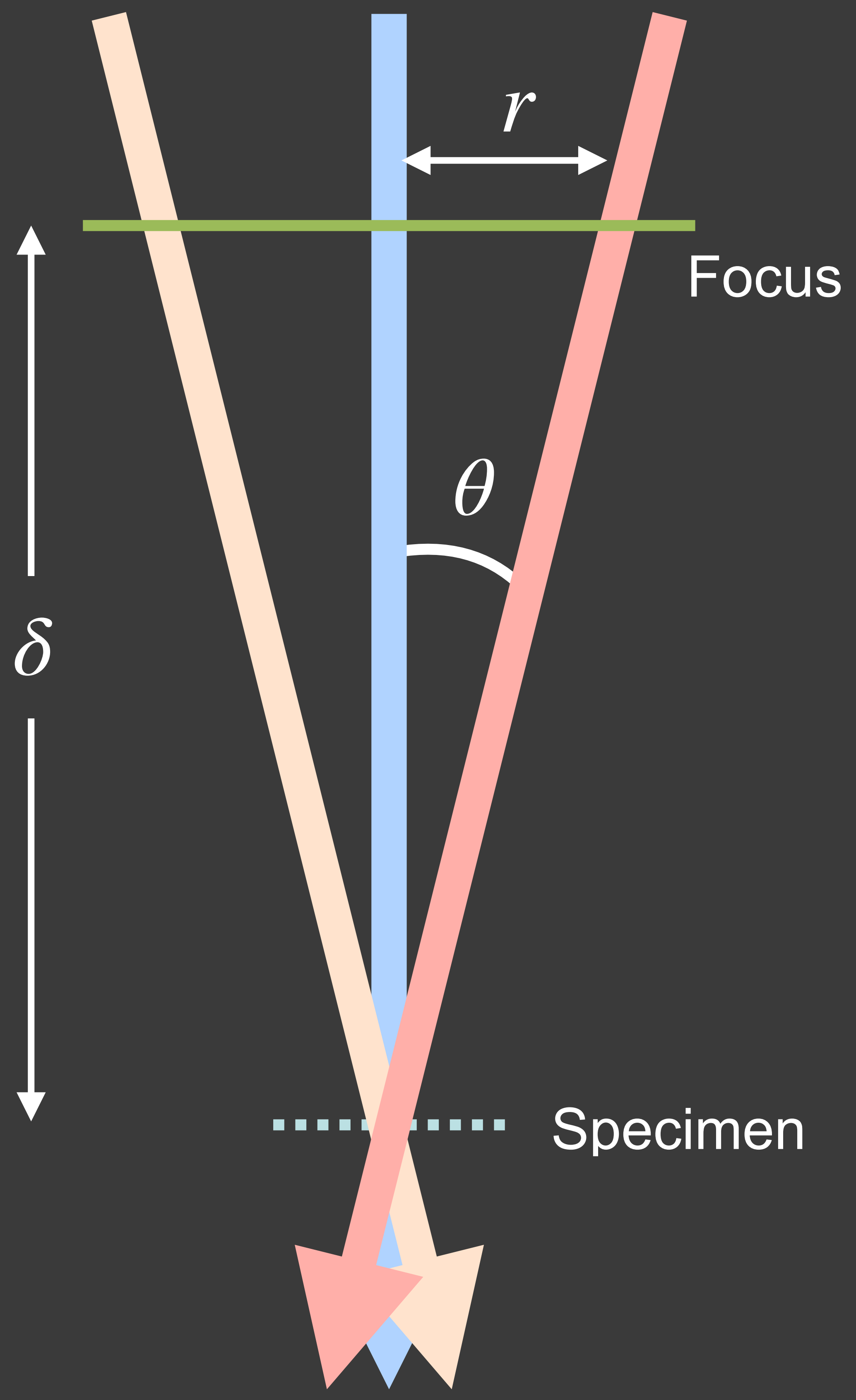


Object



3 μm defocus

With large defocus, how bad is the image delocalization?



The dispersion radius is given by

$$r = \delta \tan\theta$$
$$= \delta\lambda s \text{ (small angle approx*)}$$

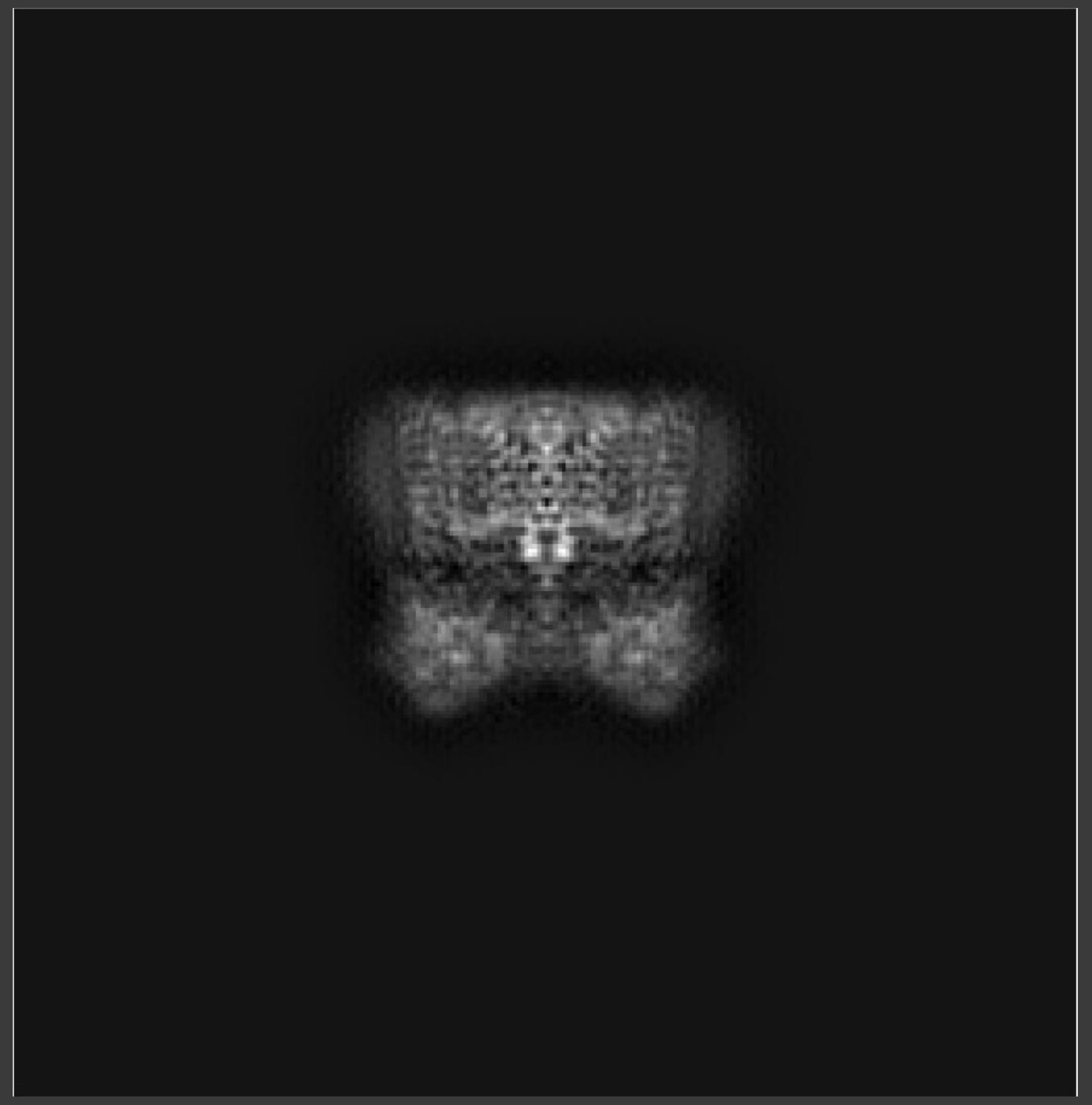
For example at 3μm defocus and 3Å resolution

$$\delta = 3 \times 10^4 \text{Å}$$
$$\lambda = .02\text{Å}$$
$$s = 0.33\text{Å}^{-1}$$

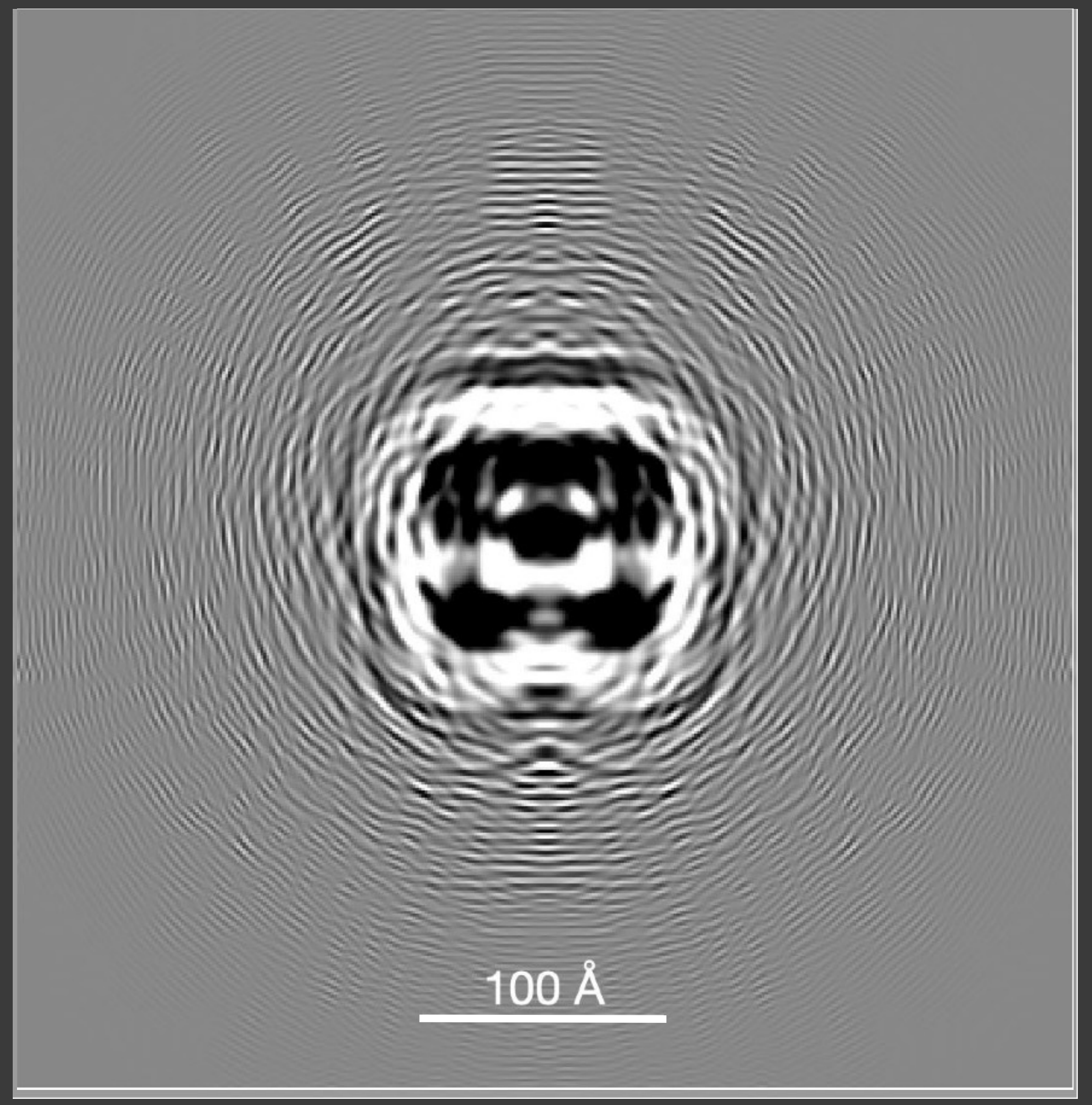
then

$$r = 200\text{Å}$$

In this case one would want 200Å of space in the box around each particle image.



Object



3 μm defocus

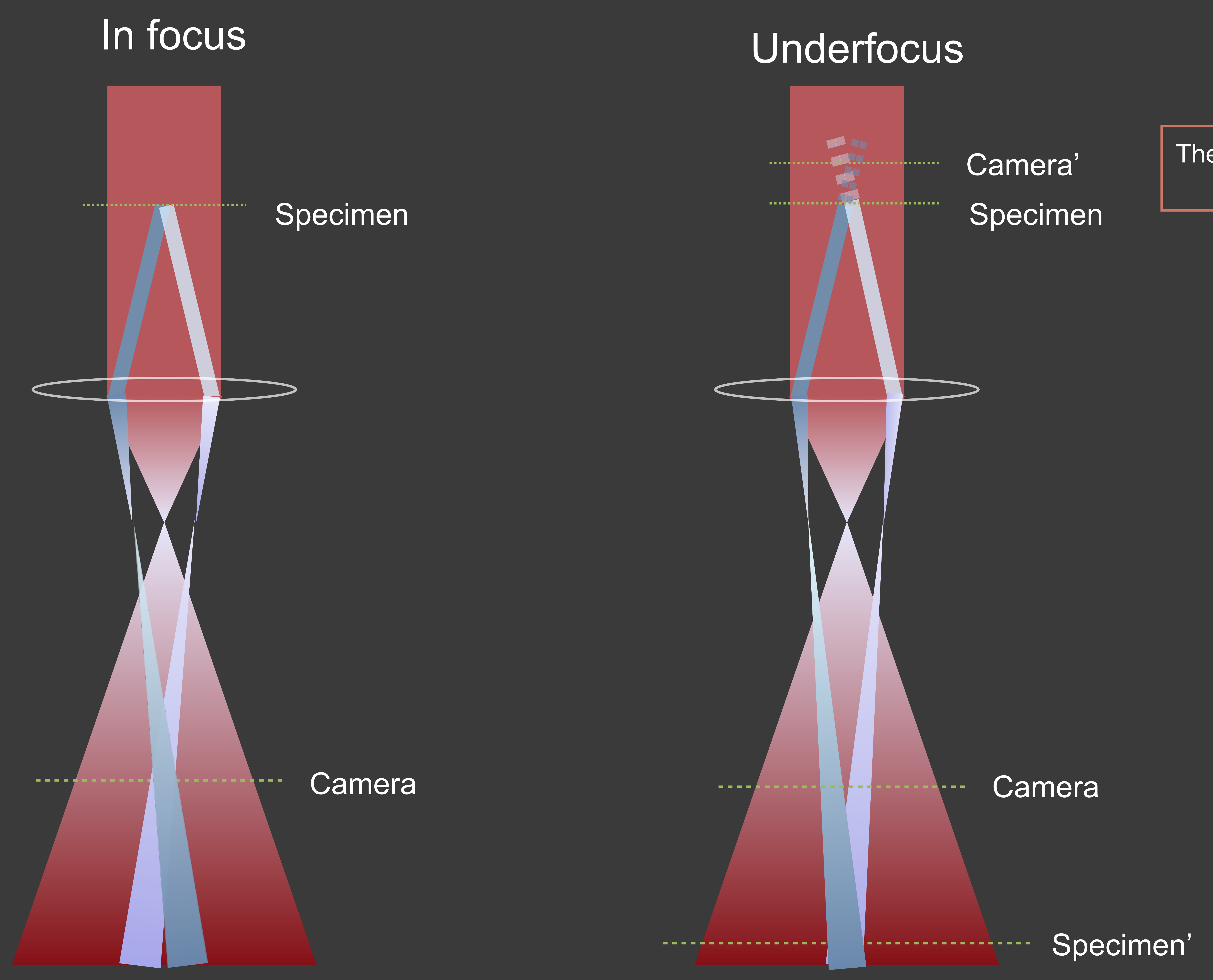
---

*Note: beyond about 3Å, spherical aberration needs to be taken into account too.

Phase contrast and the contrast transfer function

1. Complex numbers: review
2. Defocus contrast and the CTF (simple version)
3. Image delocalization
4. Objective lens effects on the CTF

An objective lens reproduces interference patterns at the camera



The camera “looks” above the specimen

With spherical aberration a lens bends high-angle rays more strongly

Spherical aberration changes the defocus by

$$\delta' = -C_s\lambda^2 s^2/2.$$

The contrast transfer function now includes δ',

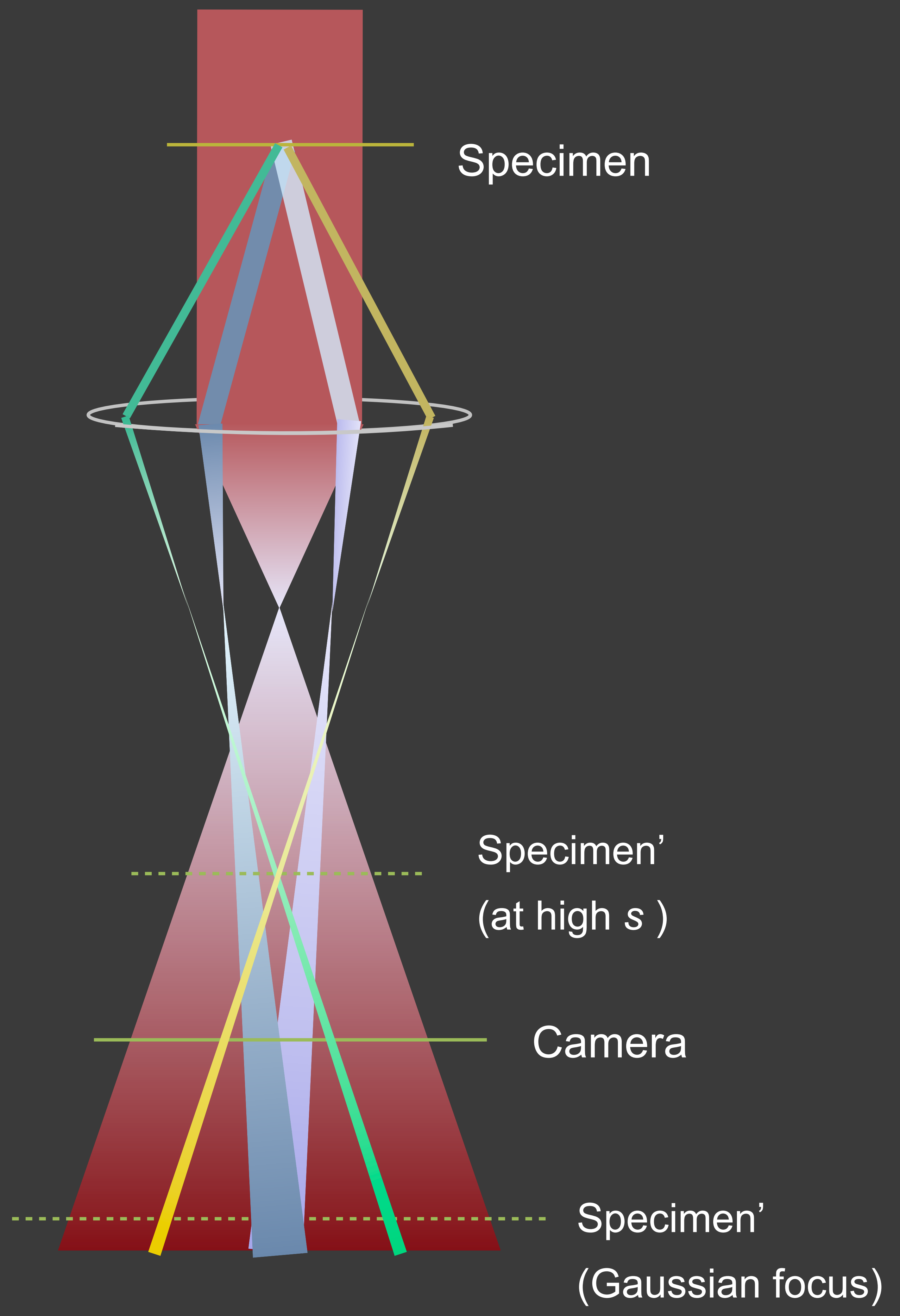
$$\mathrm{CTF} = \sin(-\pi\lambda\,(\delta + \delta')\,s^2)$$

or, expanded,

$$\boxed{\mathrm{CTF} = \sin(-\pi\lambda\delta s^2 + \frac{\pi}{2}C_s\lambda^3 s^4)}$$

The coefficient C_s is typically ~2mm. Spherical aberration typically becomes important for $s \gtrsim 0.25\text{Å}^{-1}$, or about 4Å resolution.

Underfocus



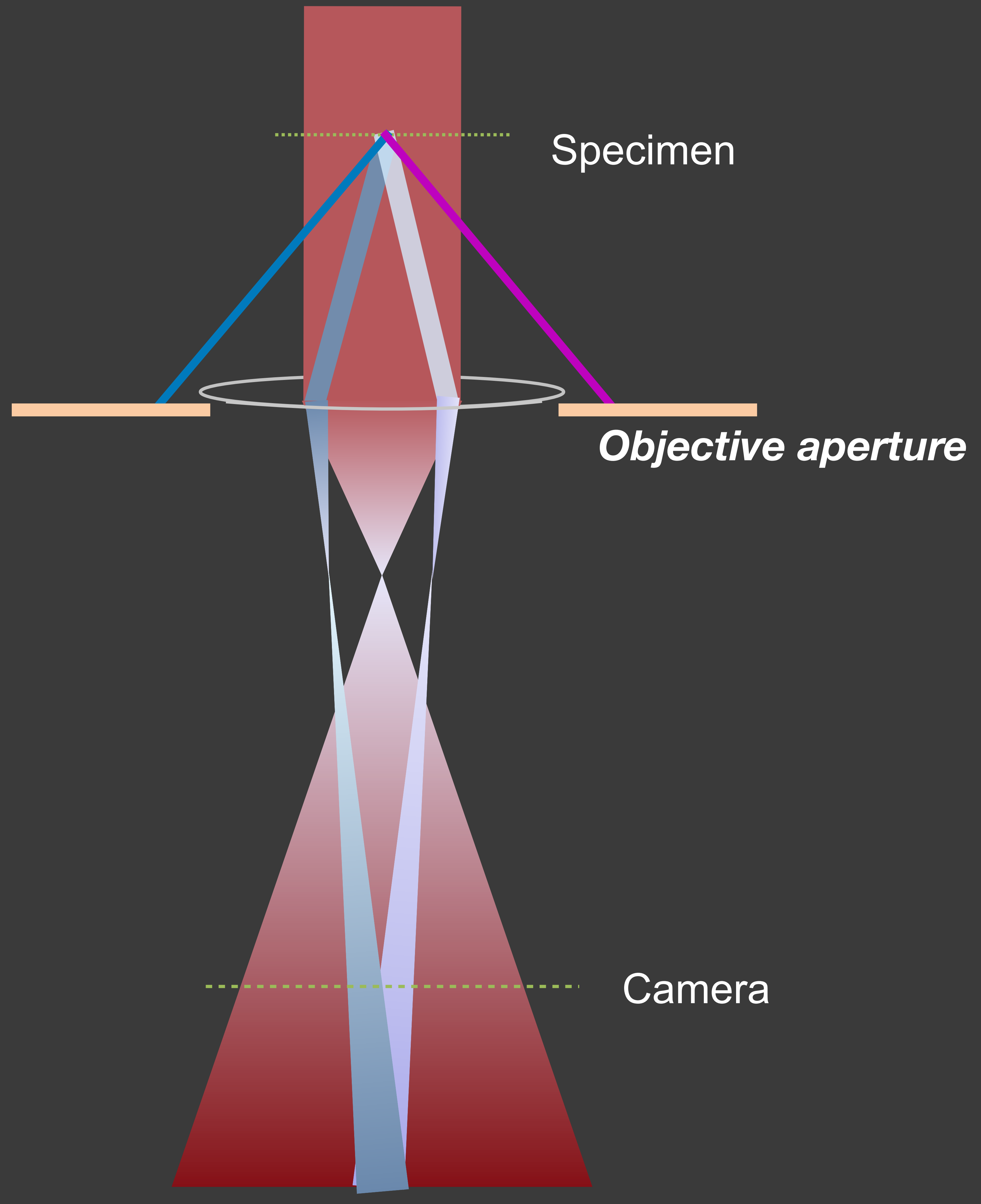
Very high-angle scattering yields amplitude contrast

Electrons that pass very close to an atomic nucleus are scattered at high angles, and are caught by the objective aperture.

- The loss of these electrons results in a small amount of negative amplitude contrast.
- For proteins α is typically around 0.05.
- The amplitude contrast term allows the CTF to have a small negative value even at zero spatial frequency.

Combining all these terms, the contrast transfer function is given by

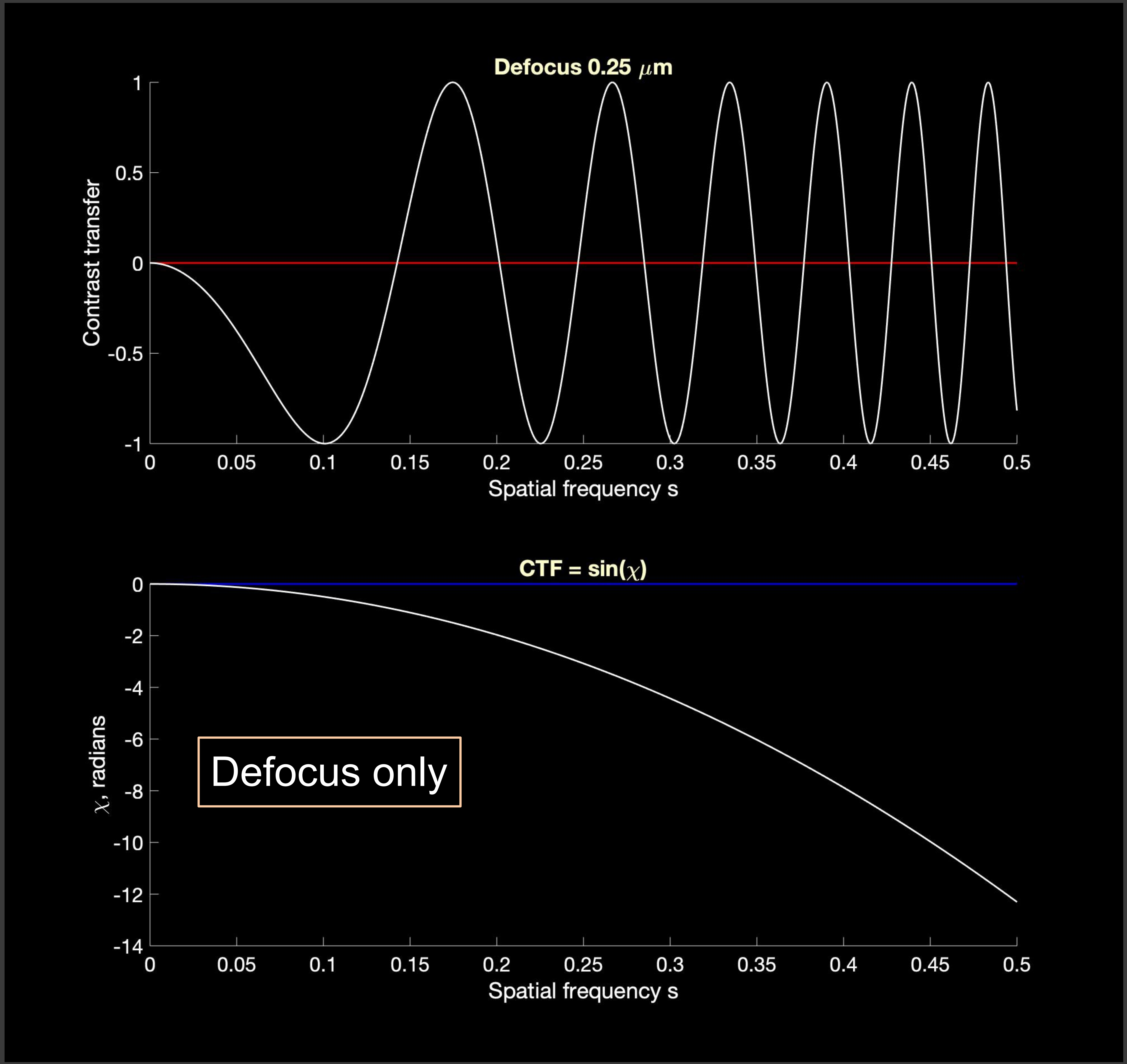
$$\mathrm{CTF} = \sin(\underbrace{-\pi\lambda\delta s^2}_{\textit{defocus}} + \underbrace{\frac{\pi}{2}C_s\lambda^3 s^4}_{\textit{sphere abb.}} \underbrace{- \alpha}_{\textit{amplitude}})$$



The simple defocus contrast is what we've seen before

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

defocus sphere abb. amplitude



Now adding in spherical aberration and amplitude contrast

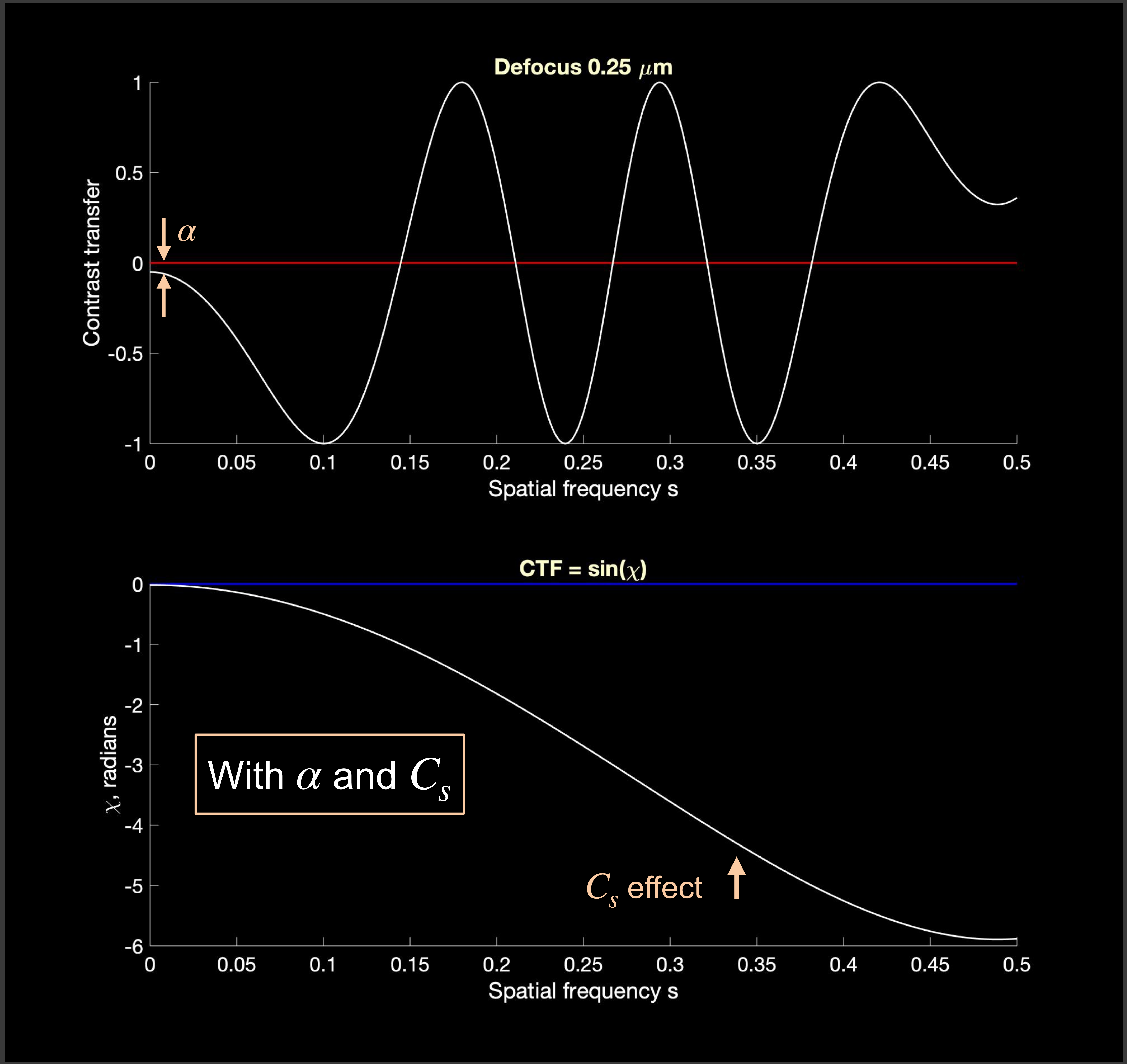
Here you can see why everyone uses underfocus: the amplitude contrast and defocus contrast are additive in this case.

Also, Cs has the effect of reversing some of the oscillations in the CTF.

Combining all these terms, the contrast transfer function is given by

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

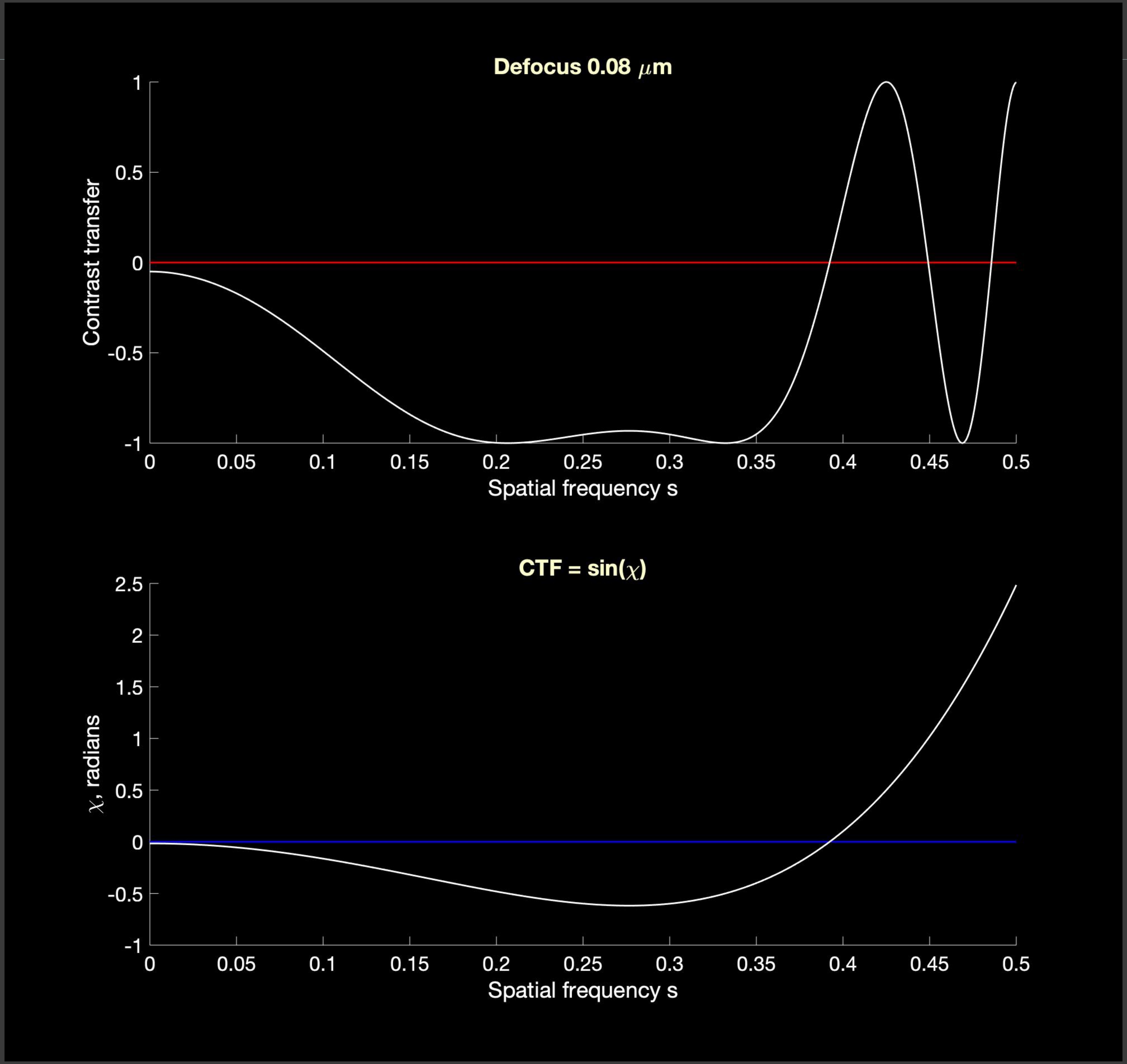
defocus sphere abb. amplitude



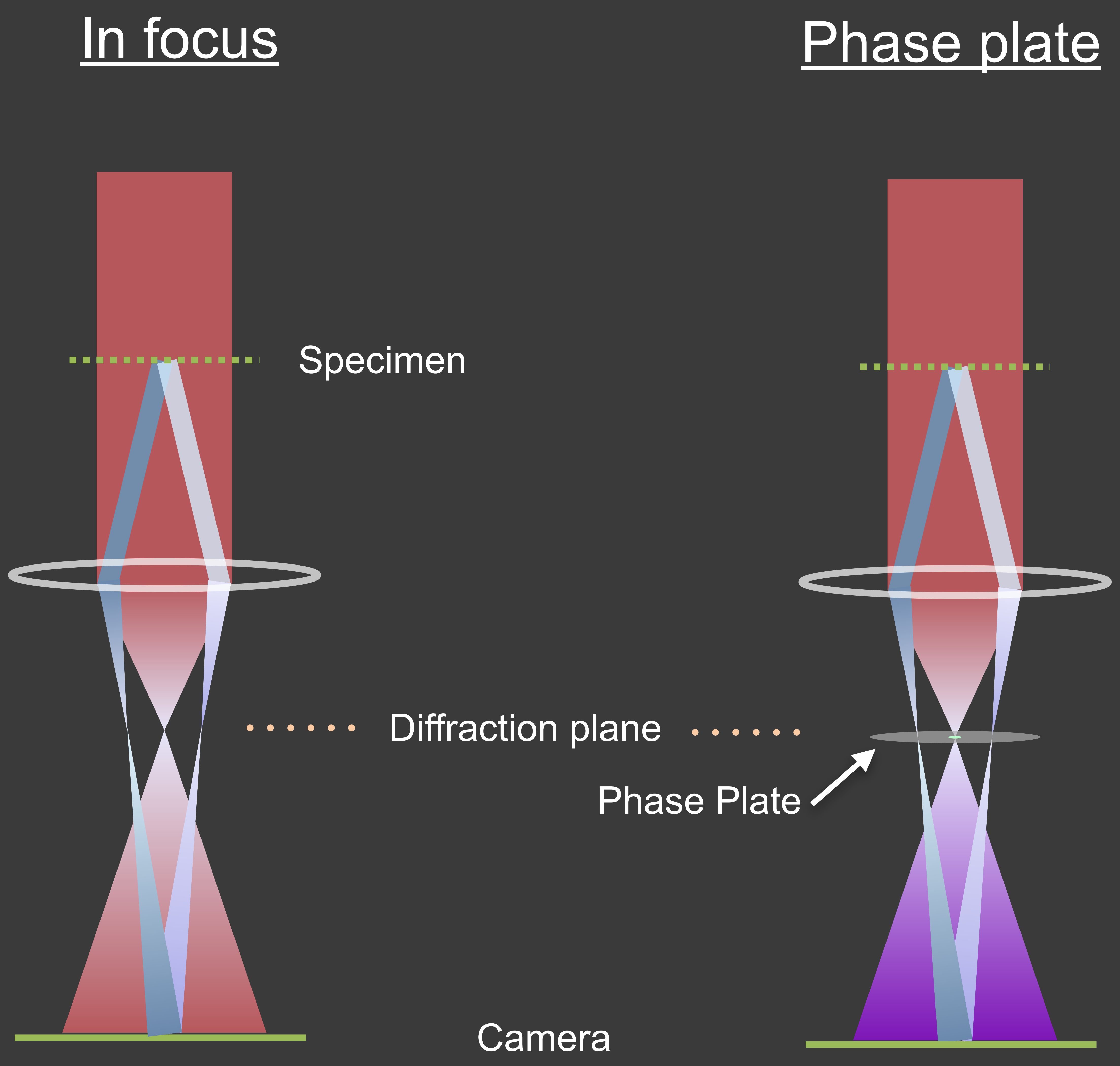
Spherical aberration can be our friend

If we're not using image processing to remove CTF effects, Scherzer defocus is a good solution: just enough defocus to give signal over a broad range of spatial frequencies.

It's popular in materials science but not much for cryoEM: the signal transfer at low frequencies is poor.



A phase plate modifies the interference of electron waves at the camera



The phase plate shifts the phase of the undiffracted beam Ψ_0 by some angle ϕ.

Then $\mathrm{CTF} = \sin(\chi - \phi)$.

If $\phi = 90^o$ then

$$\mathrm{CTF} = -\cos(\chi)$$

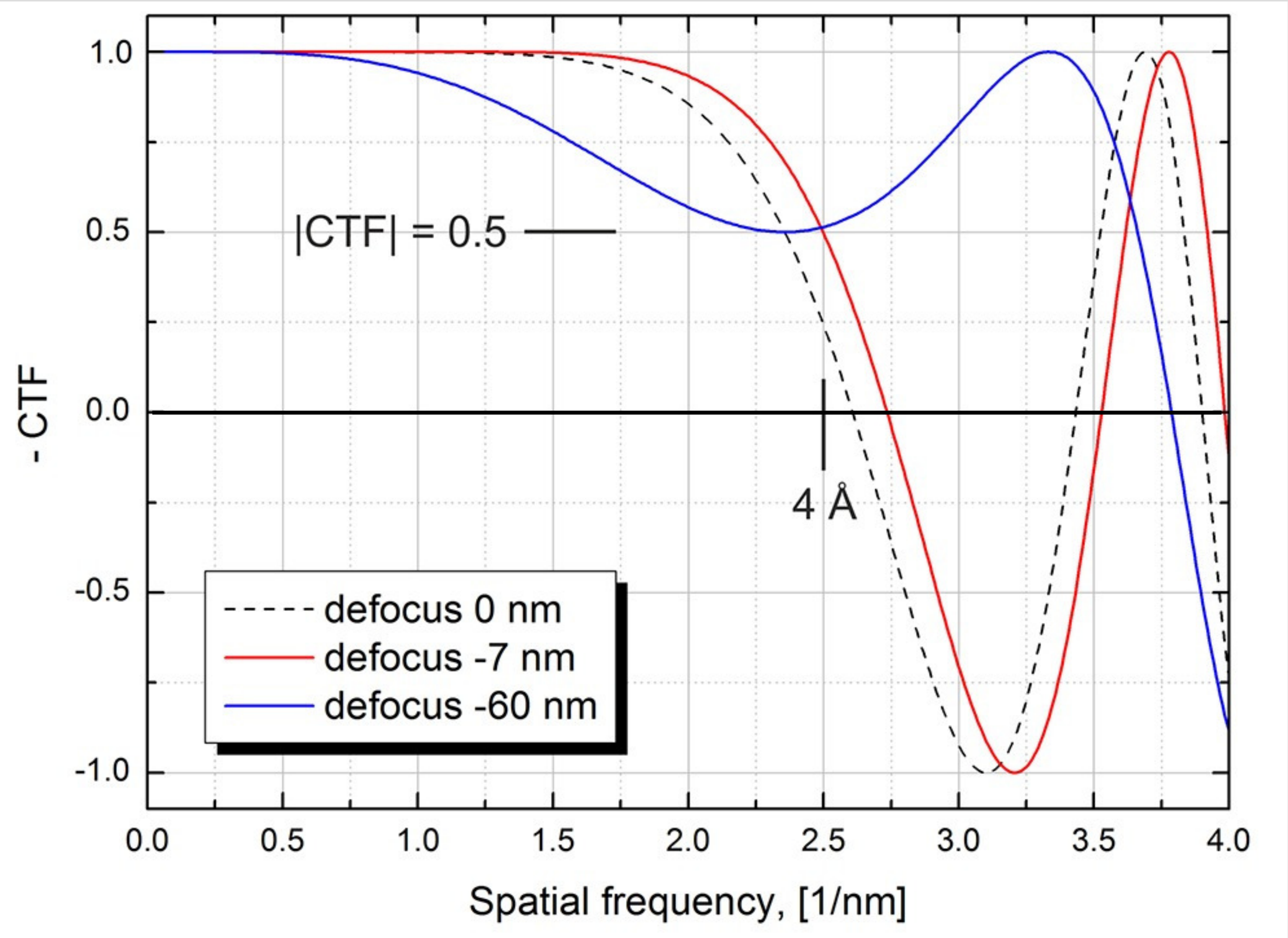
The phase plate allows in-focus imaging, given precise focusing.

Cryo-EM single particle analysis with the Volta phase plate

Radostin Danev*, Wolfgang Baumeister

Department of Molecular Structural Biology, Max Planck Institute of Biochemistry, Martinsried, Germany

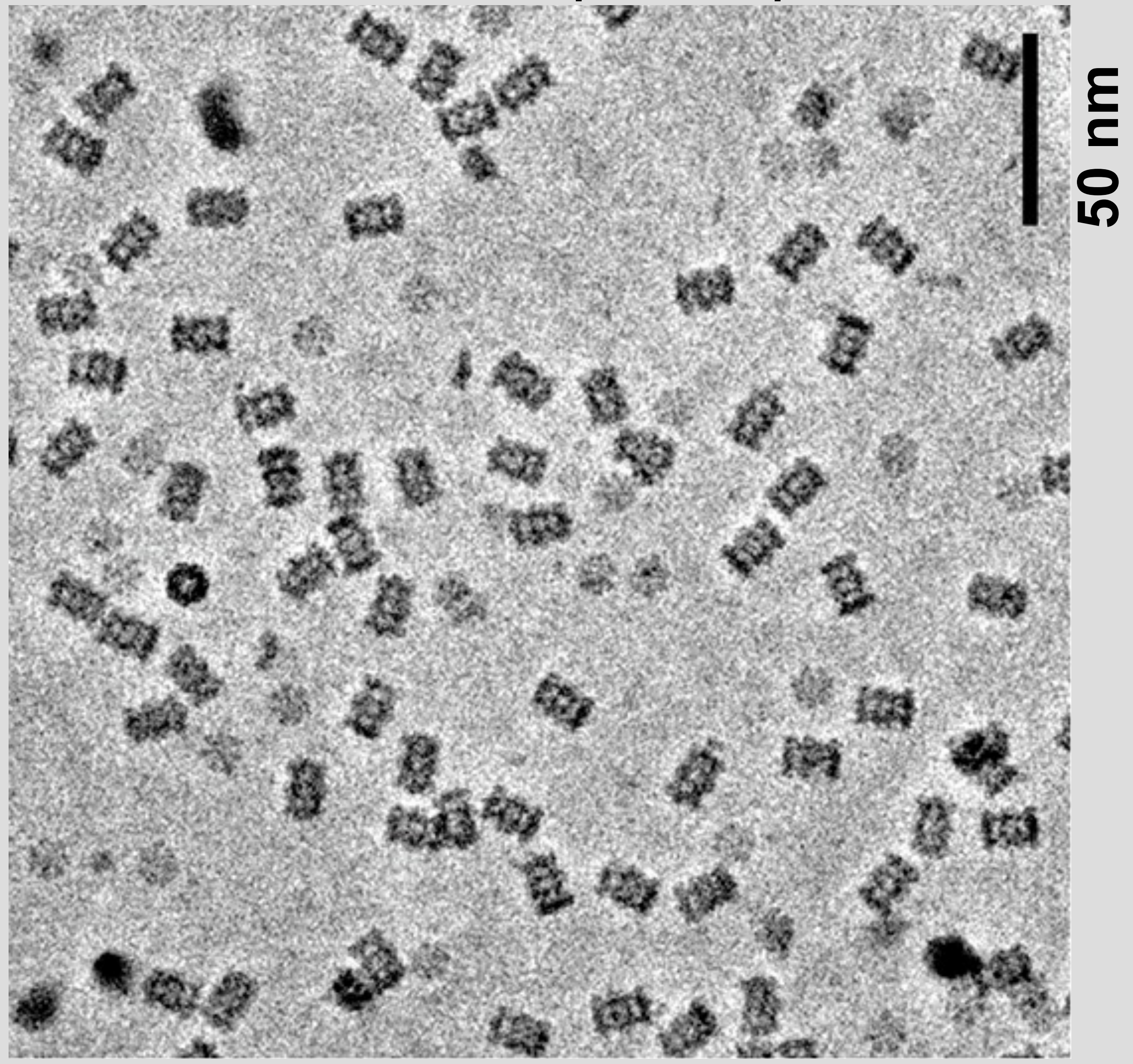
eLife 2016



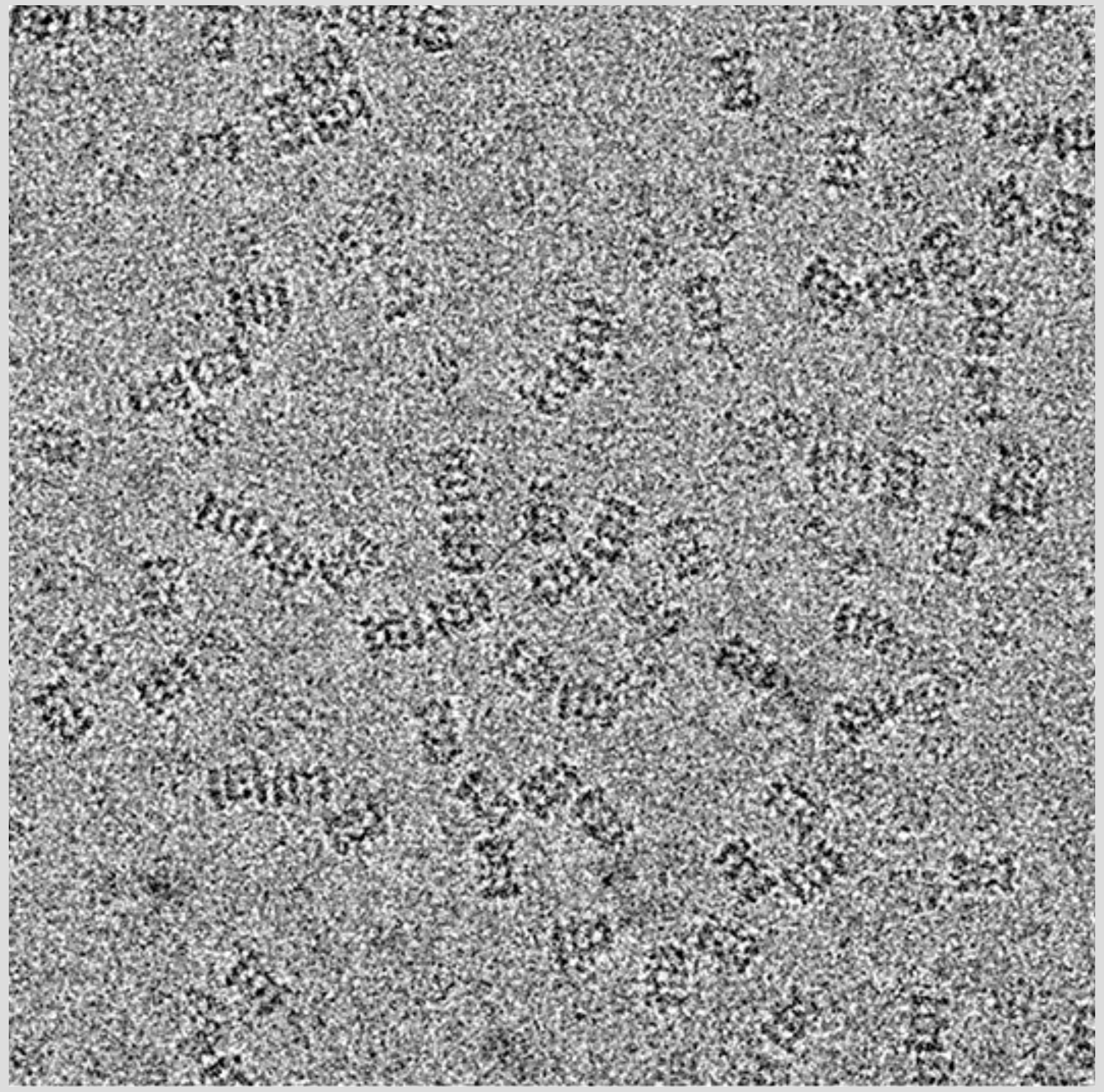
- The better low-frequency contrast makes particles much more visible.

- The defocus value must be precise within 60 nm in order to get 4 Å resolution.

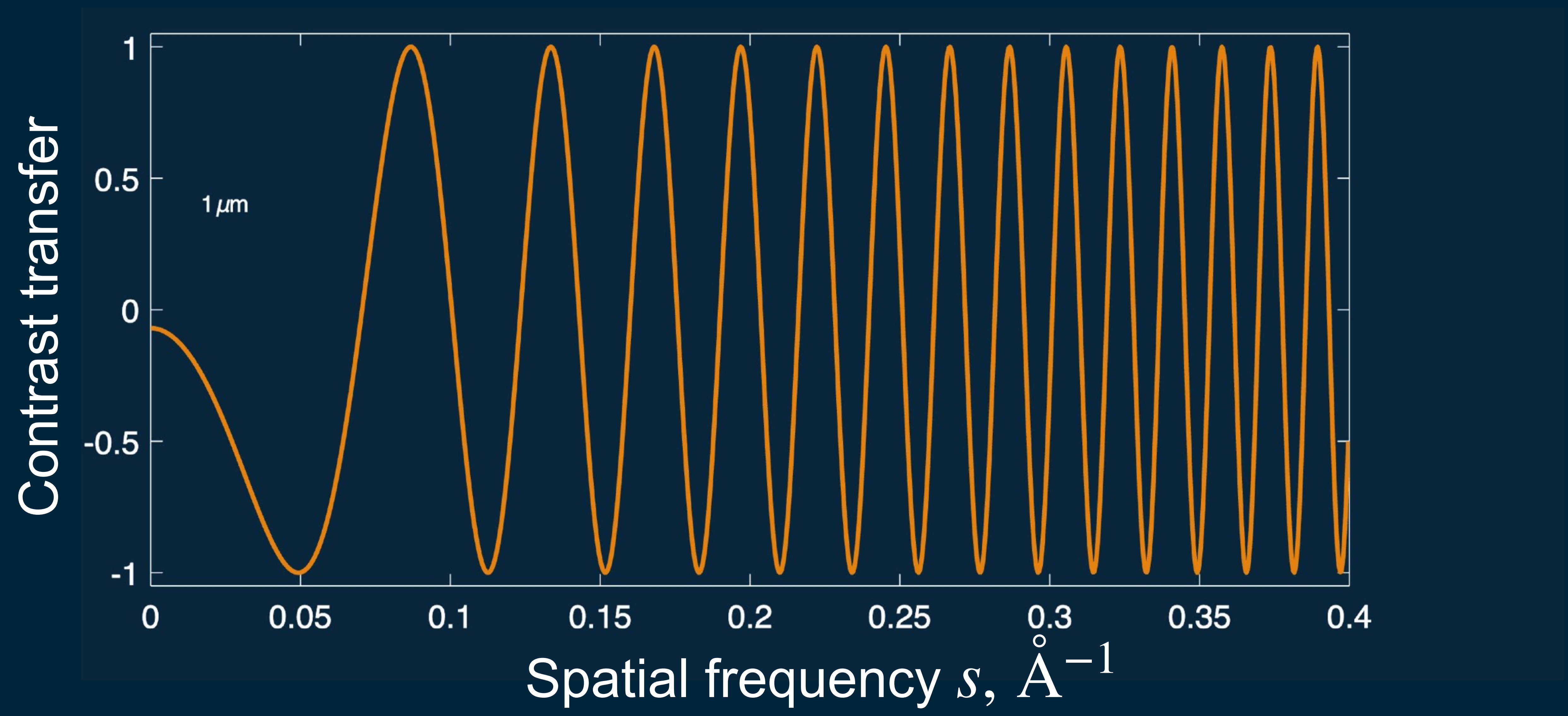
In-focus phase plate



Defocus contrast



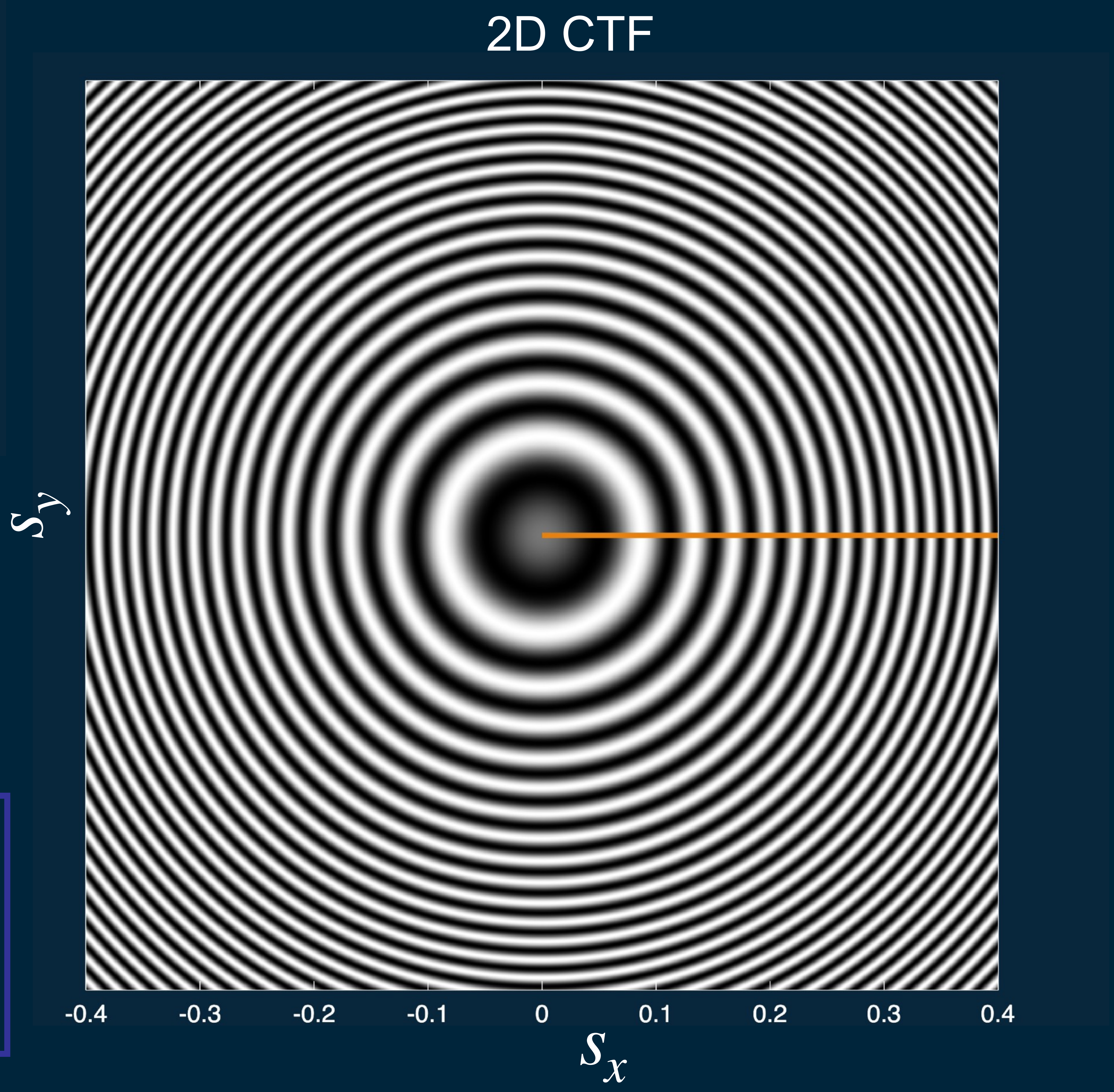
The CTF actually is a 2D function



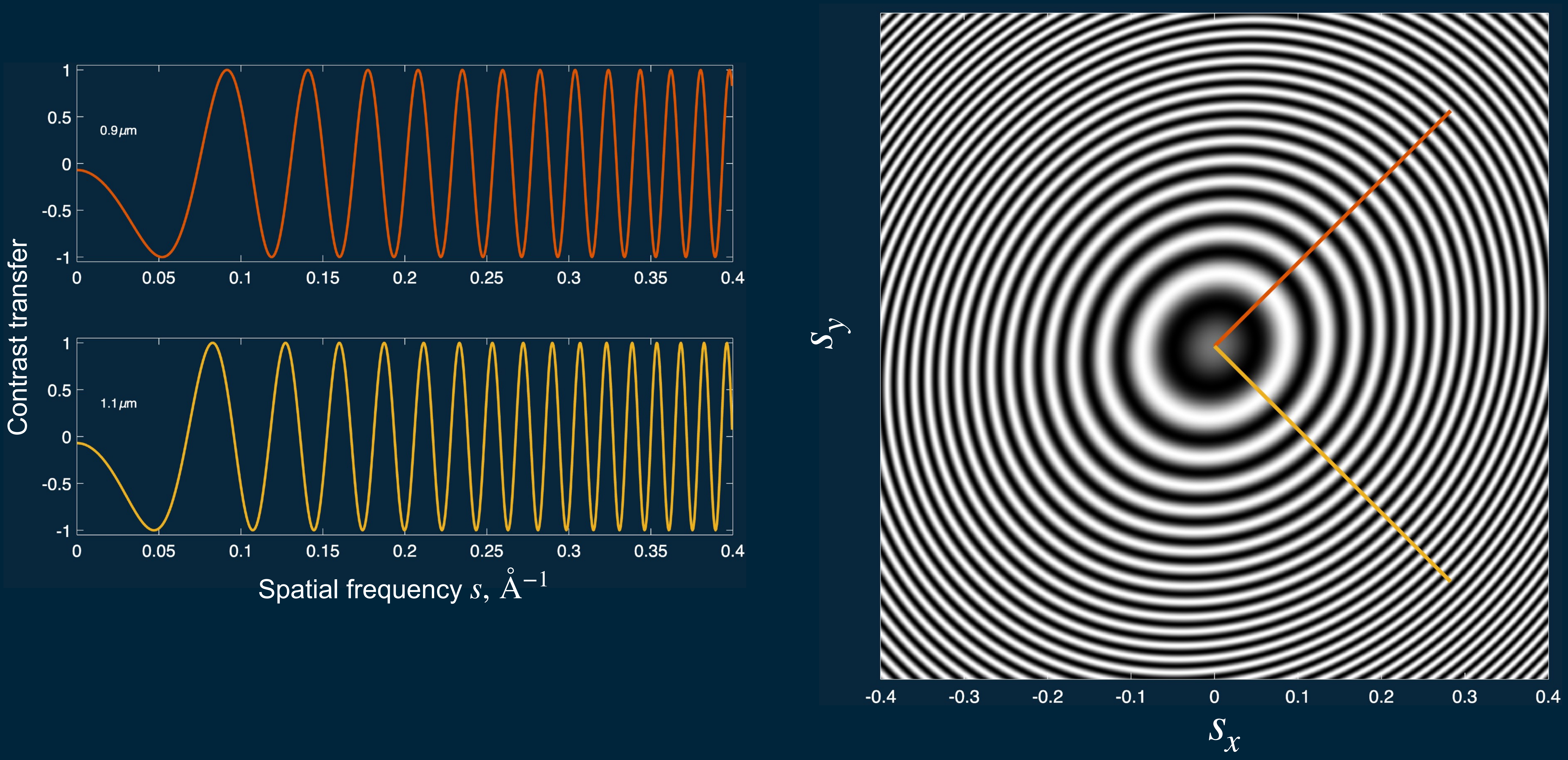
$$\mathrm{CTF} = \sin(-\pi\lambda\delta s^2 + \frac{\pi}{2}C_s\lambda^3 s^4 - \alpha)$$

defocus sphere abb. amplitude

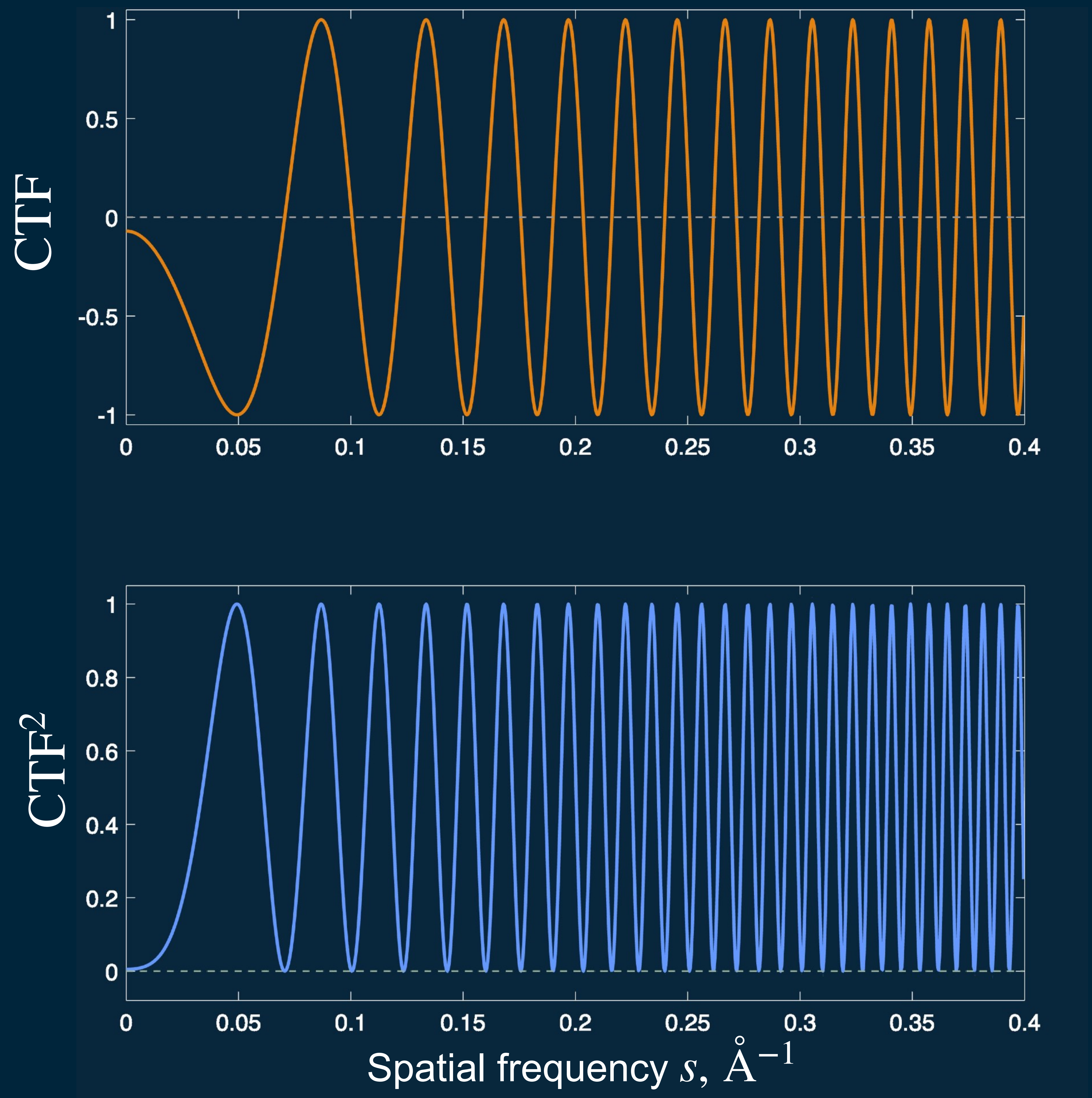
$$\mathrm{CTF} = \sin(-\pi\lambda\delta(s_x^2 + s_y^2) + \frac{\pi}{2}C_s\lambda^3(s_x^2 + s_y^2)^2 - \alpha)$$

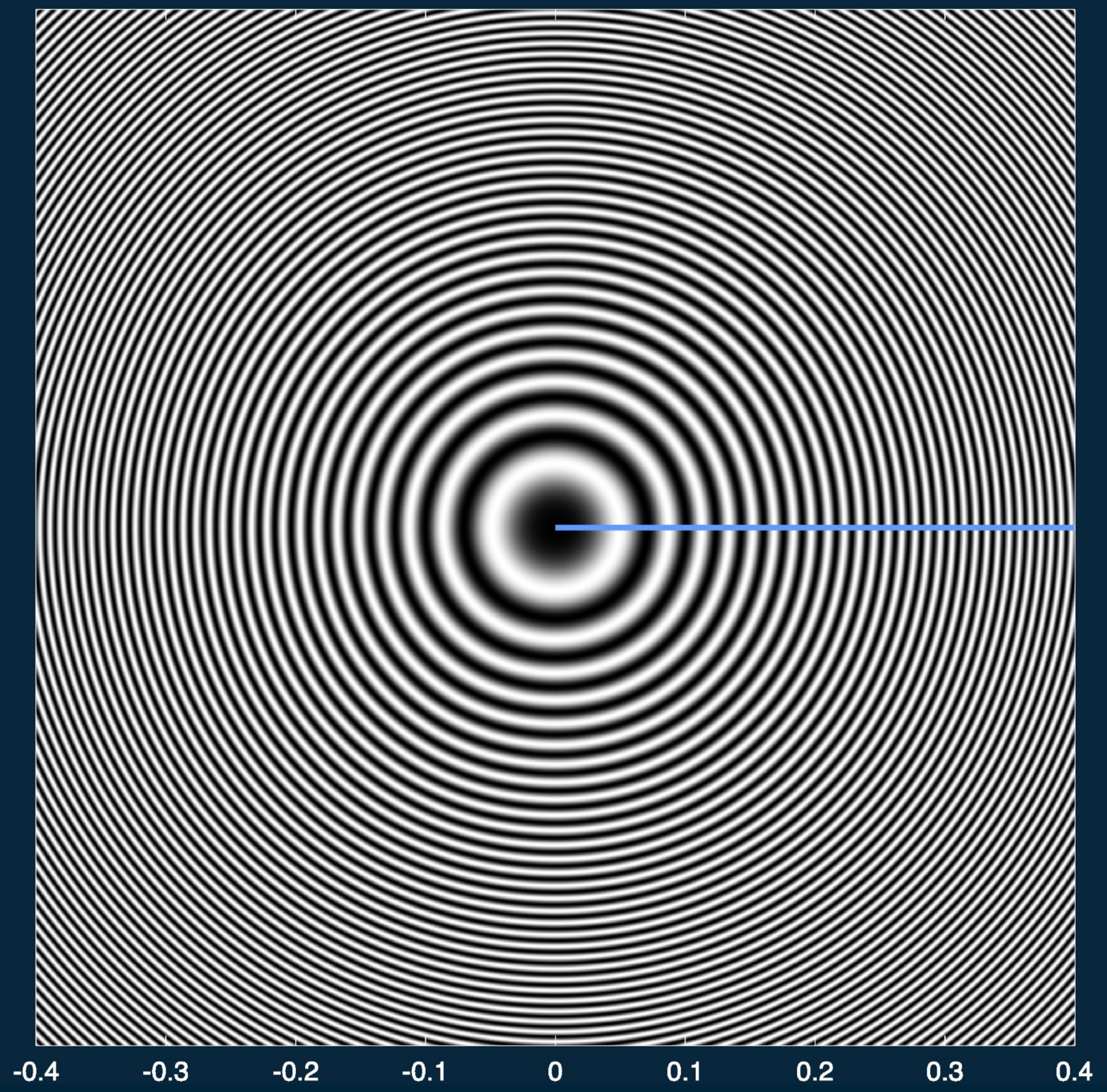


Astigmatism yields a varying defocus depending on the angle

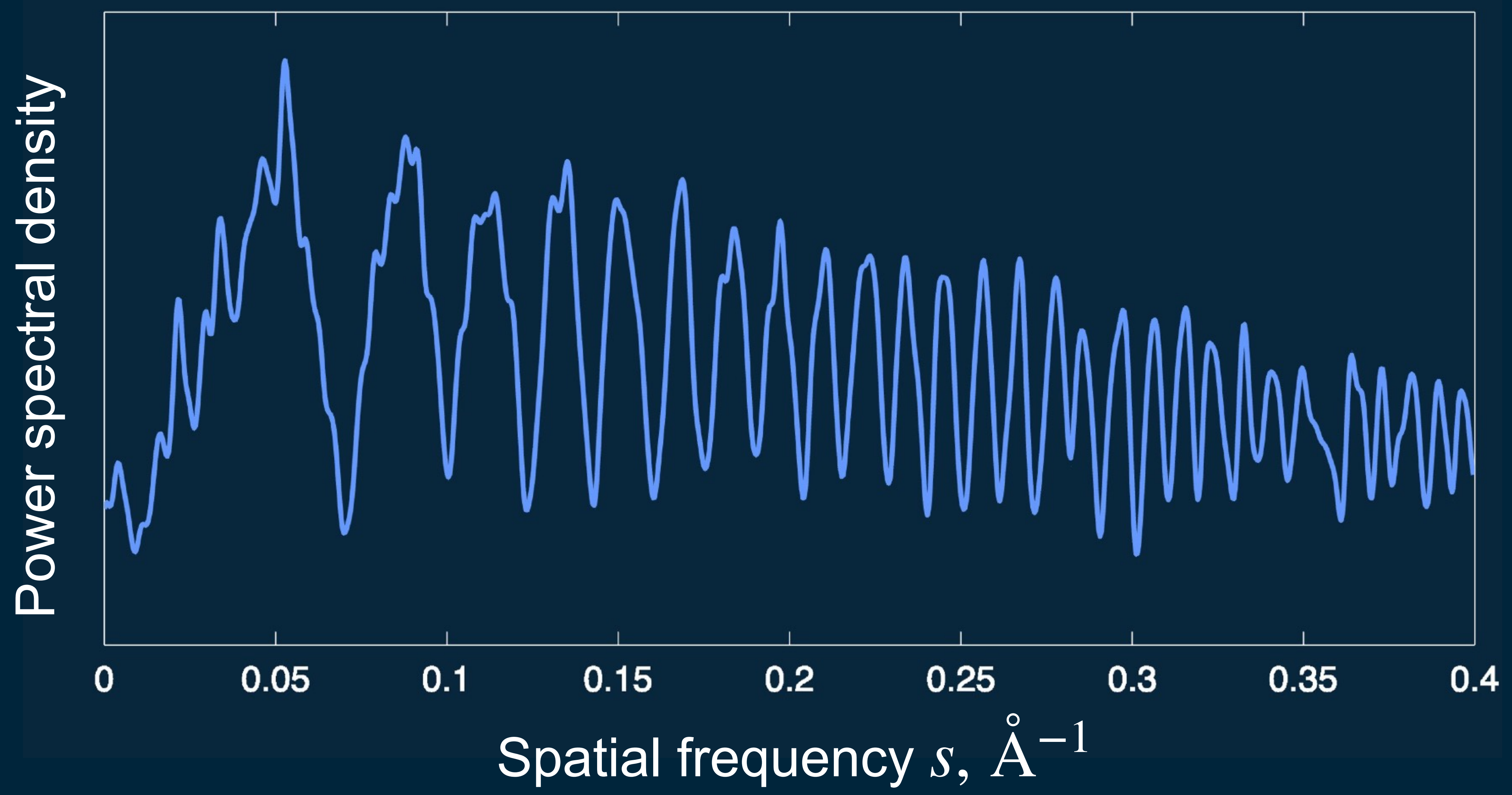


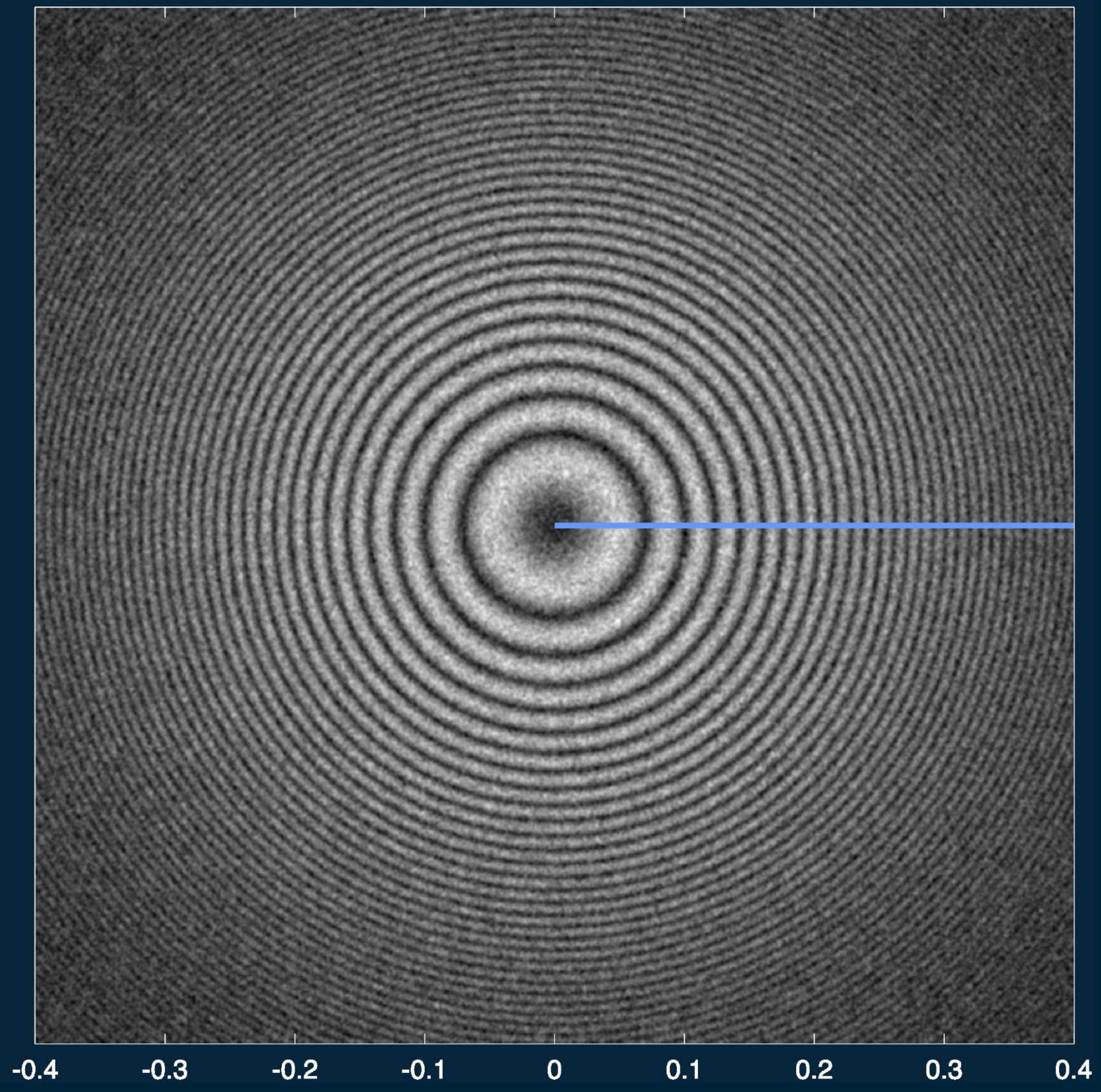
Black rings in CTF^2 show the zeros in the CTF





The power spectrum shows the same dark “Thon rings”

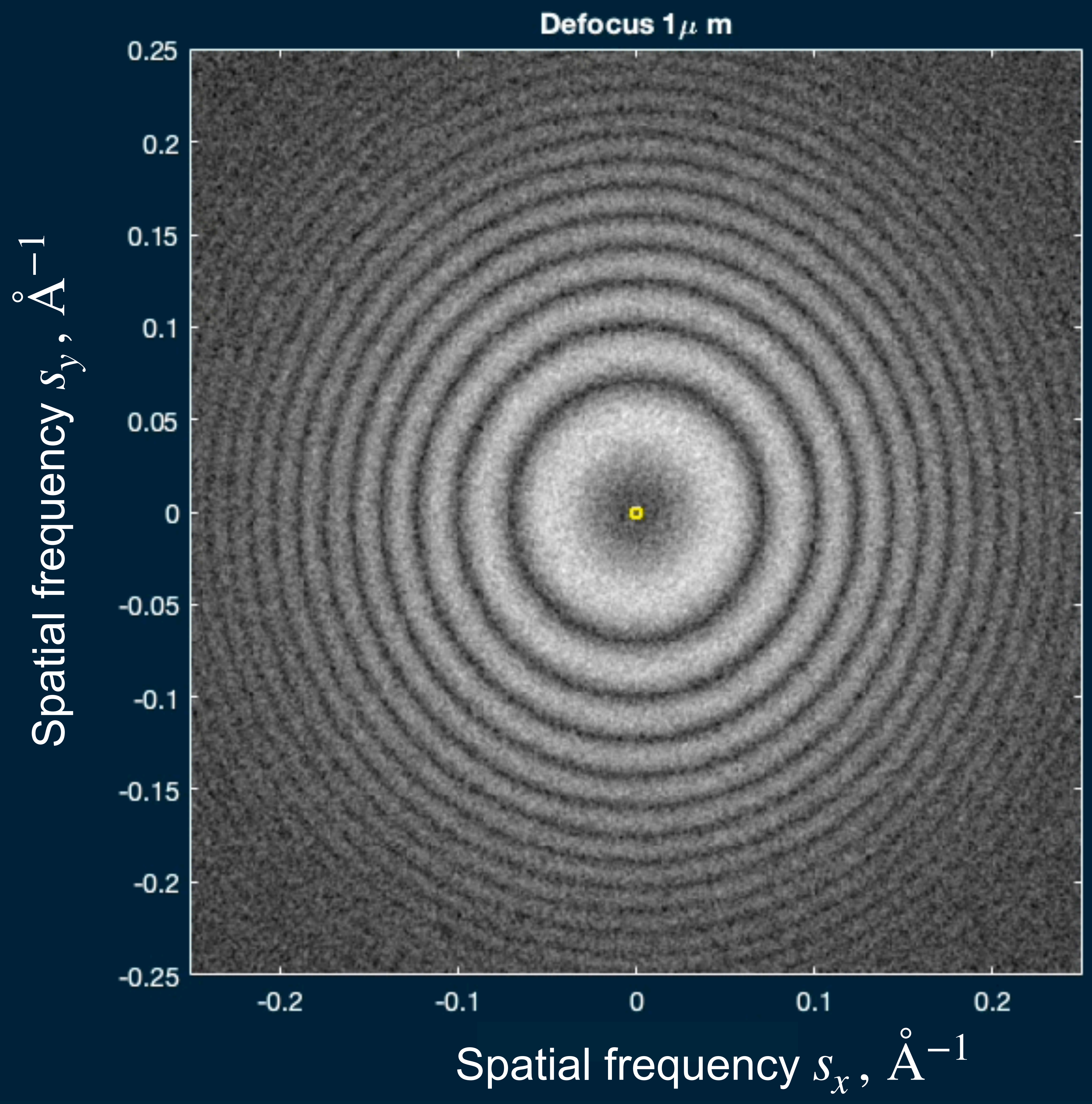




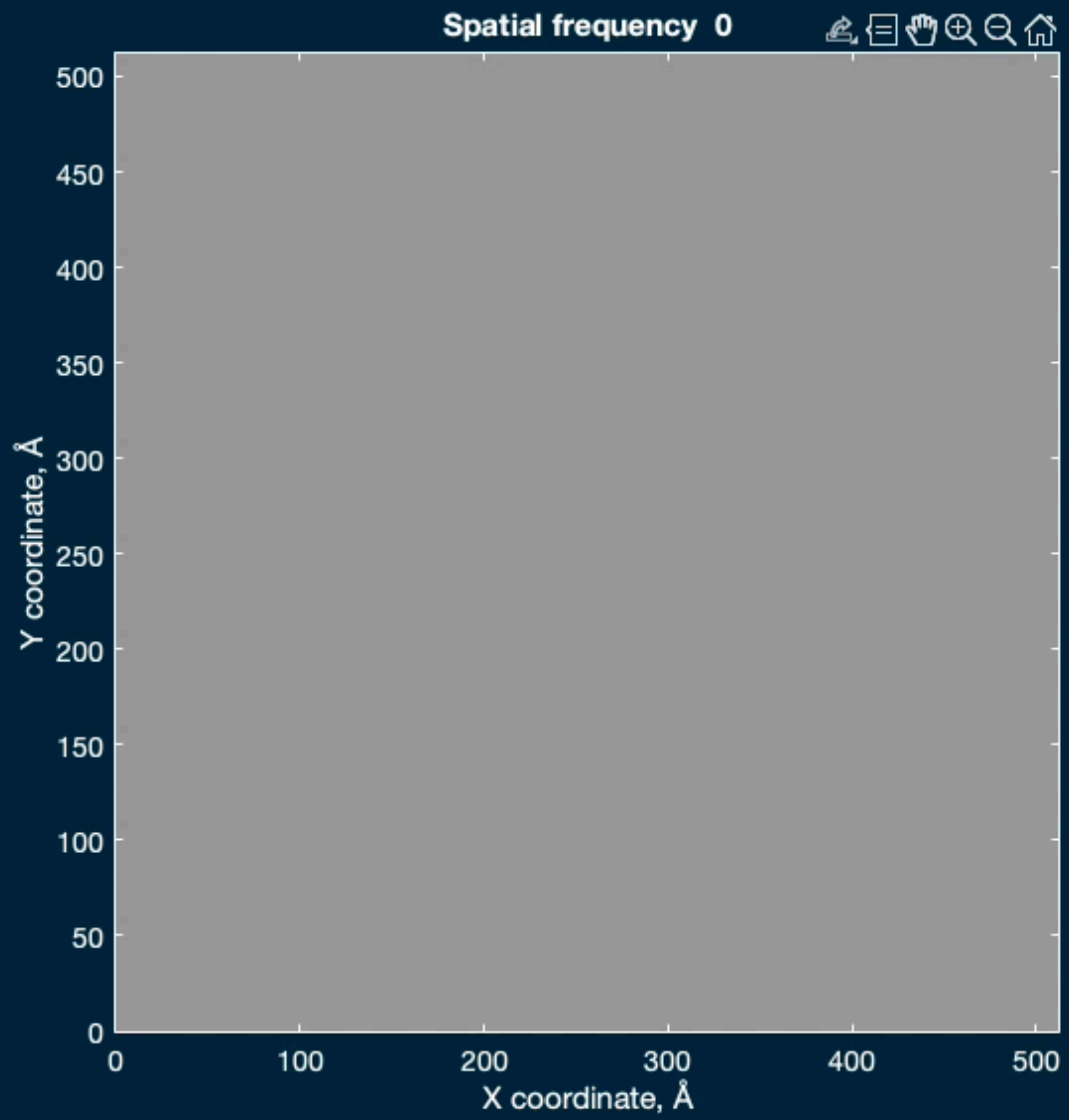
The defocus and other CTF parameters can be estimated by curve fitting.

Thon rings in the power spectrum show zeros in the CTF

Power spectrum



Grating at the spatial frequency $s = (s_x, s_y)$



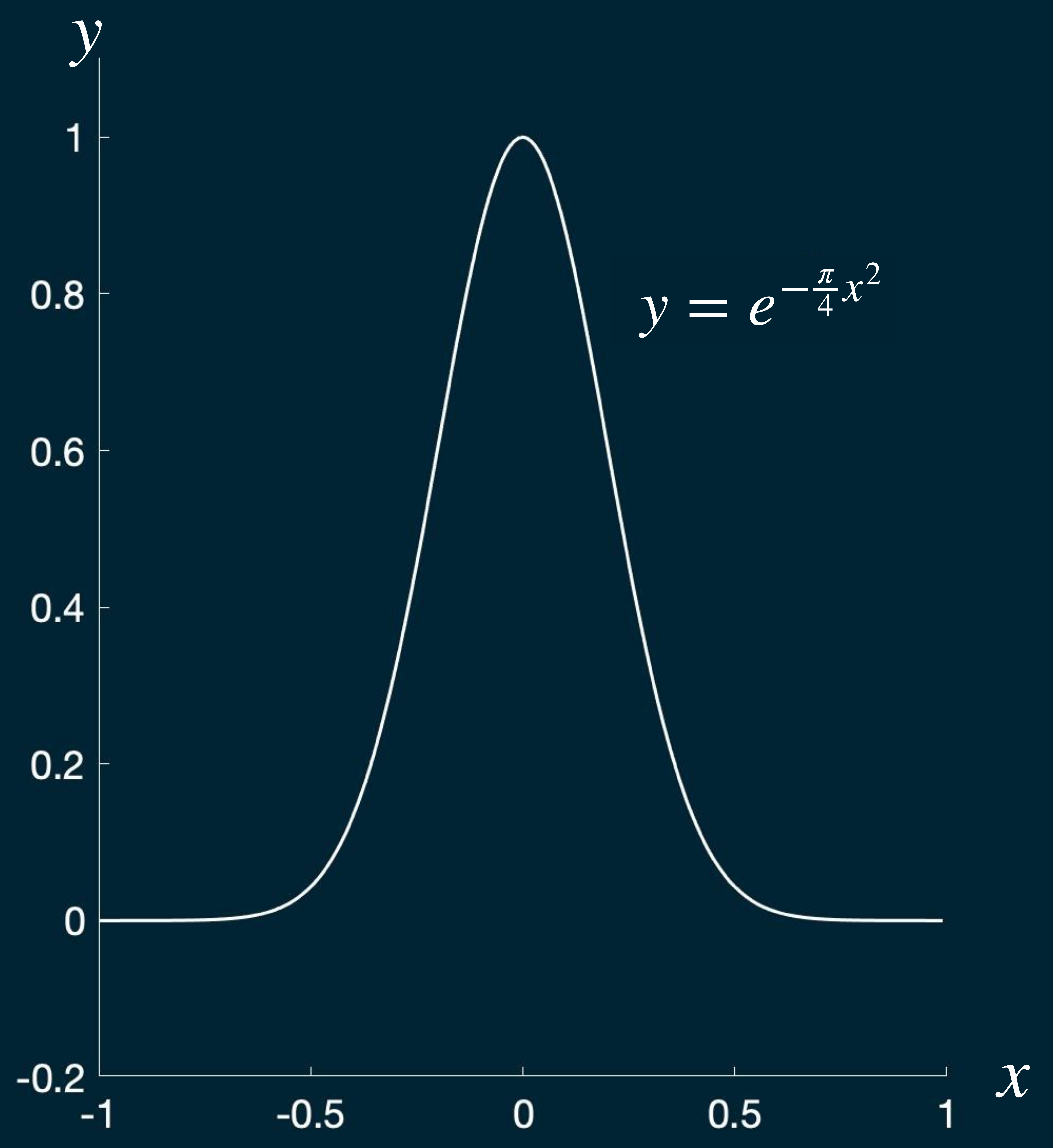
Defocus contrast: a summary

1. Electrons have really short wavelengths, and they travel through the column one by one.
2. The contrast in the image of a grating object varies with the amount of defocus
3. The grating object produces diffracted waves with shifting phase
4. When the phase of the diffracted waves is right, we have contrast.
5. A lens reproduces the wavefronts at the image plane.
6. Spherical aberration and amplitude contrast introduce new terms in the CTF.
7. A phase plate alters the wavefronts after they’ve passed through the lens.

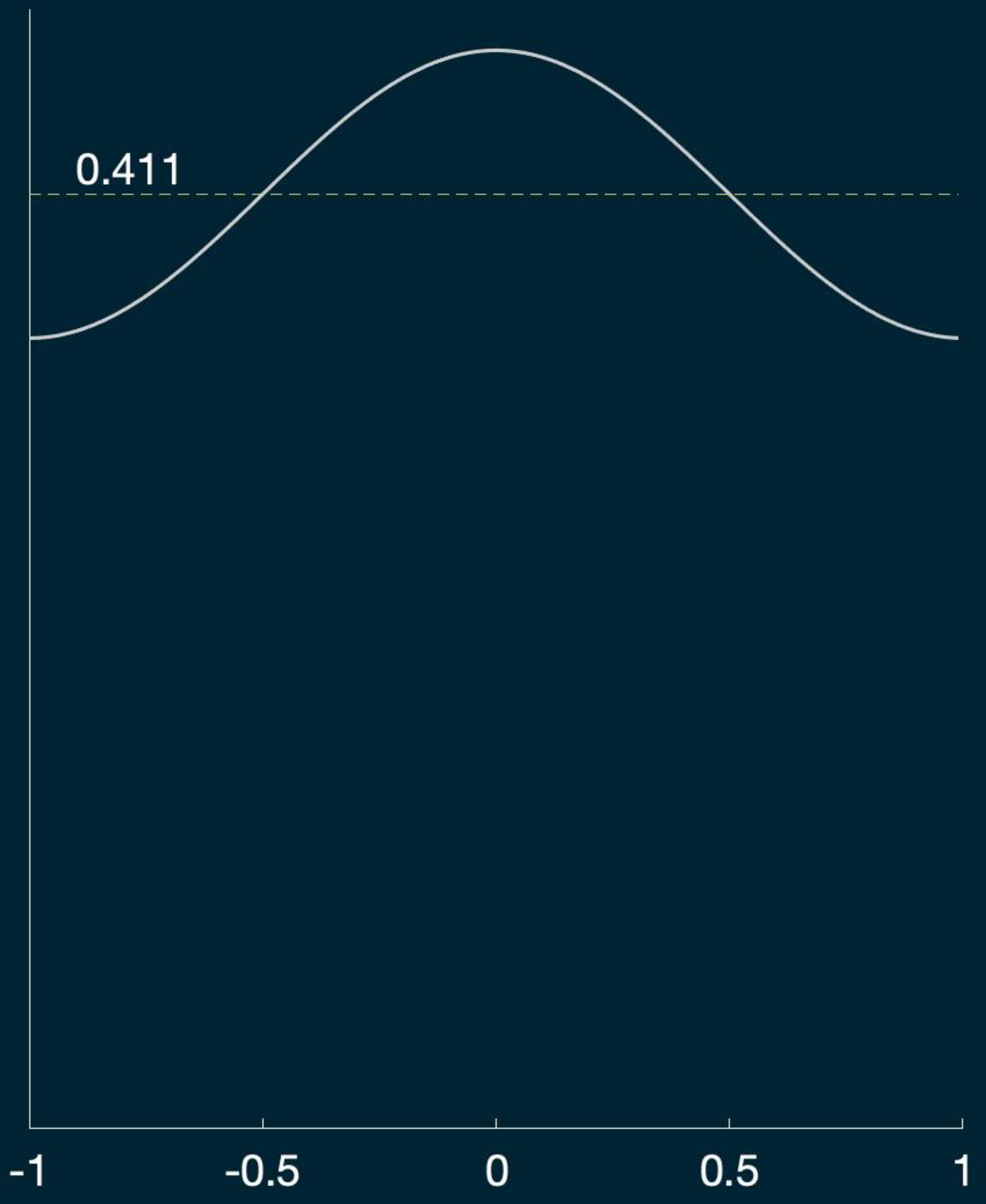
Part 2: Fourier Transforms and their properties

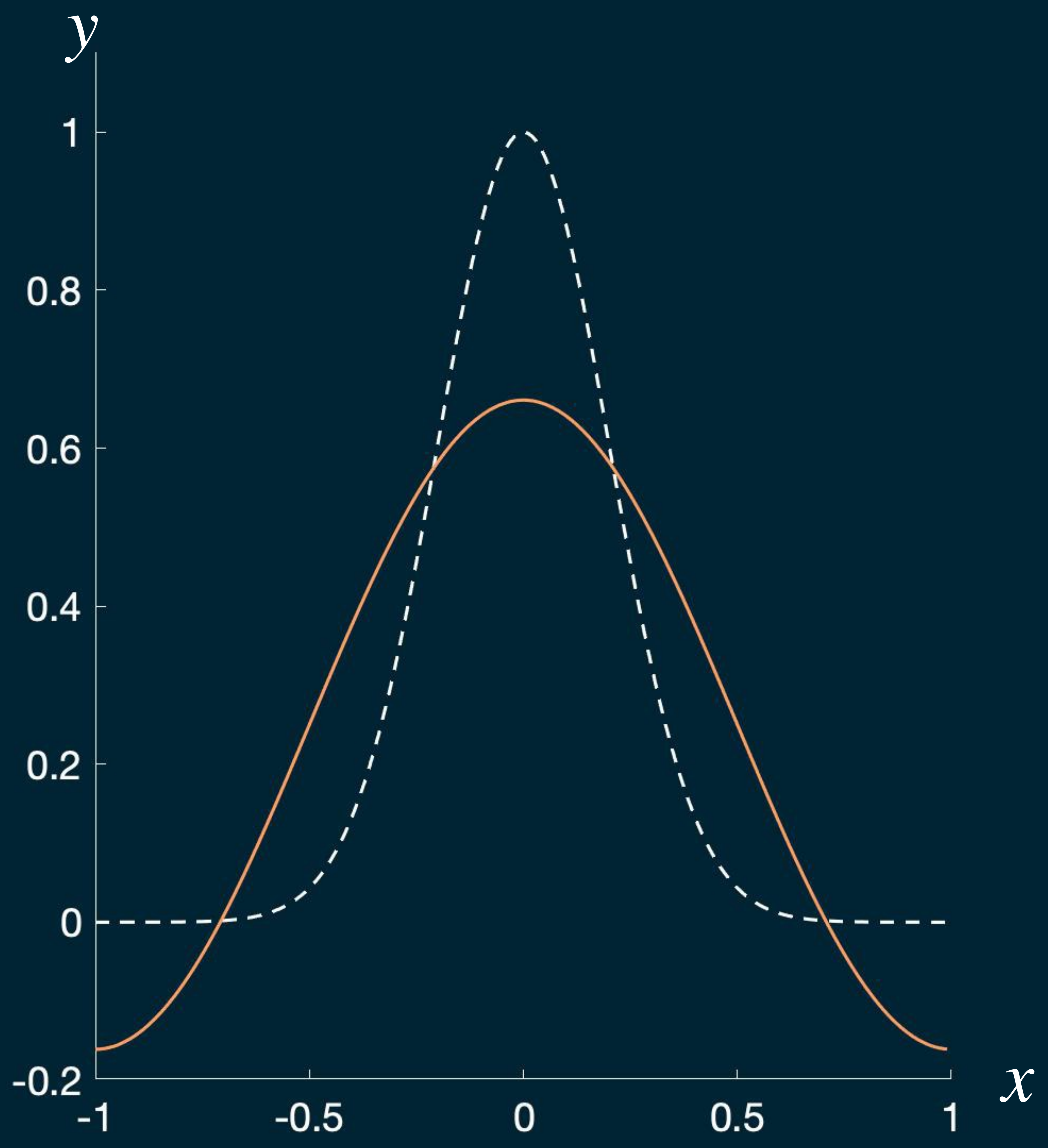
The Fourier transform in one dimension

Reconstruction of a Gaussian function from cosine waves

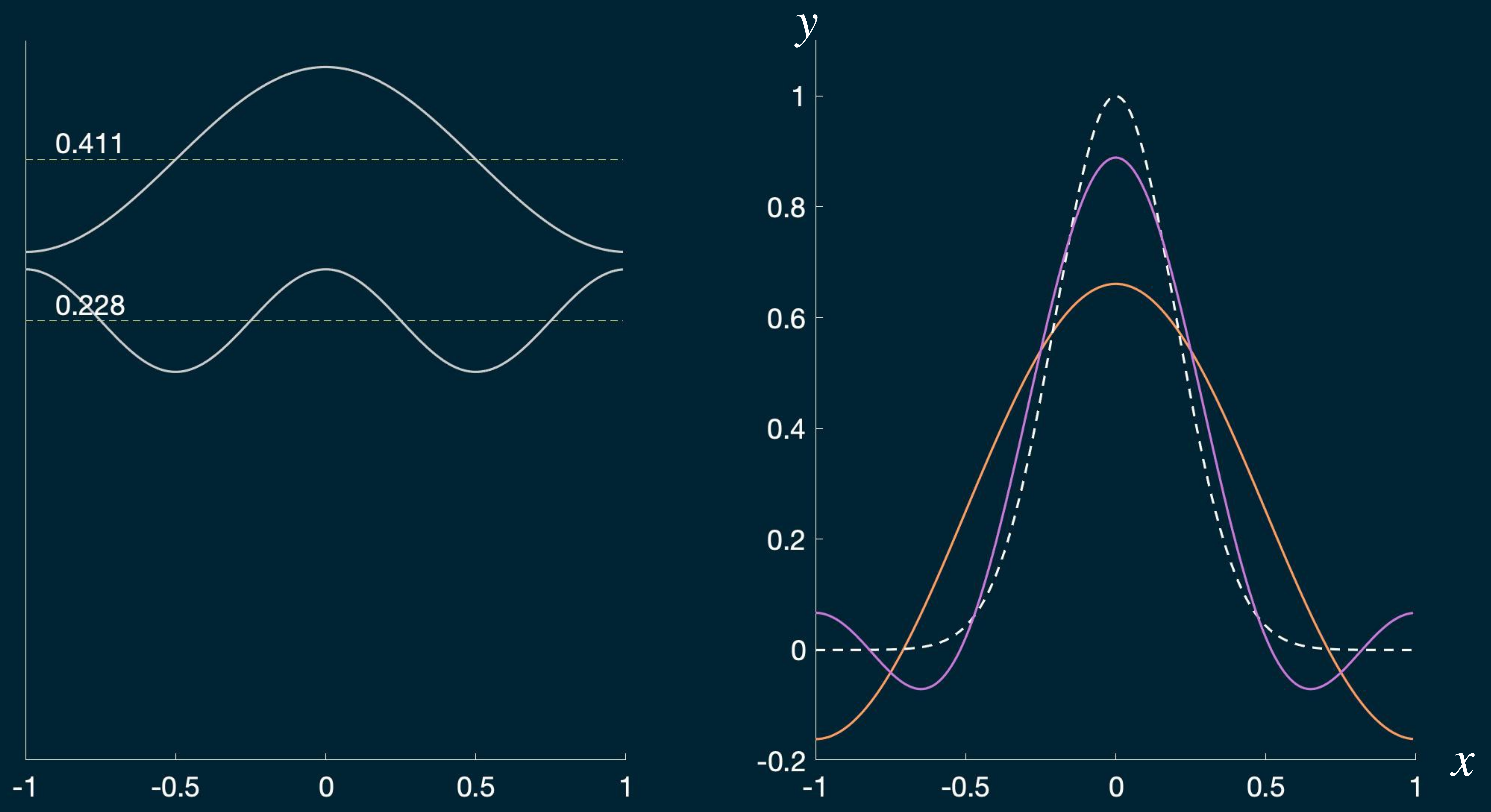


1 term

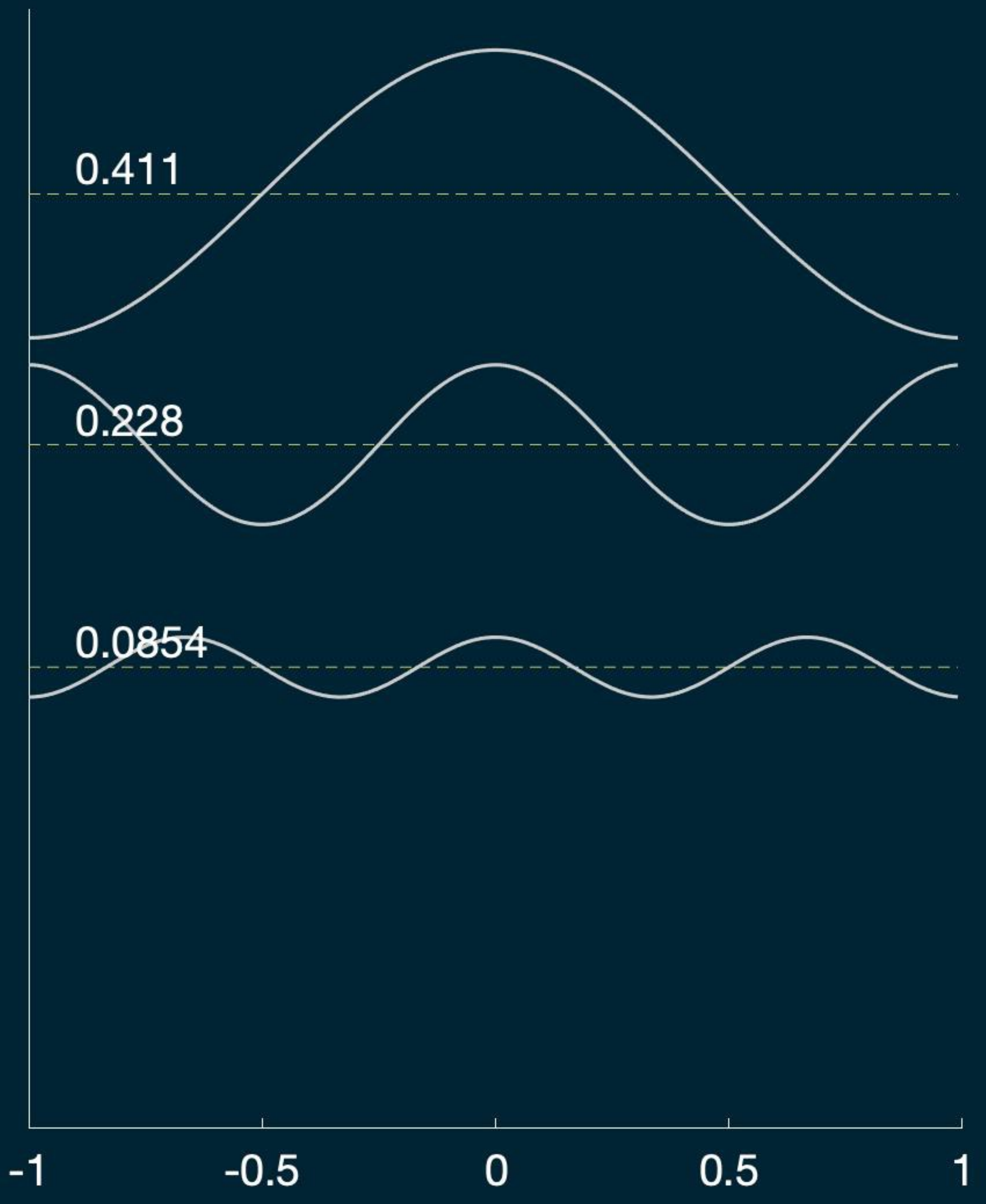


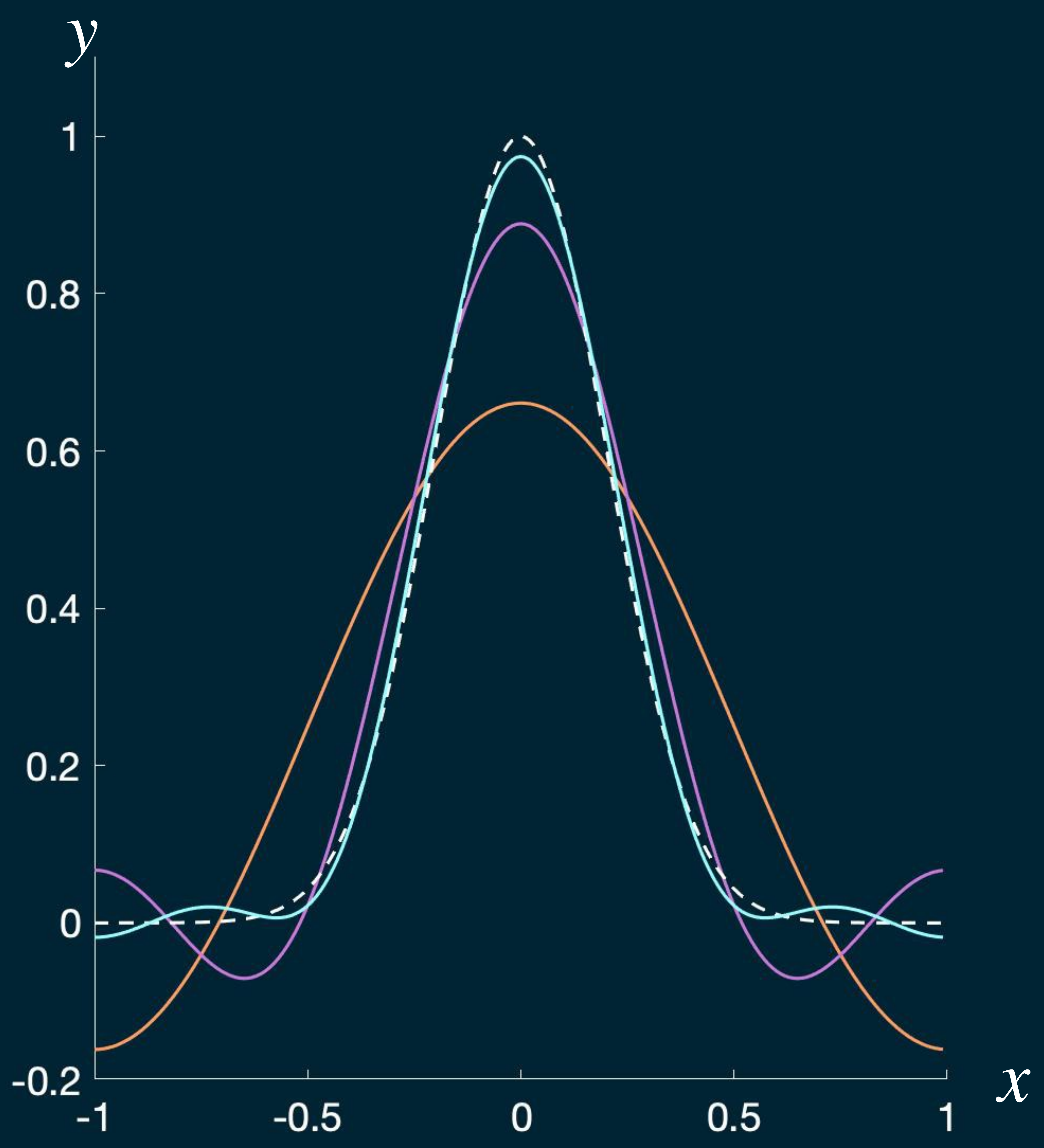


2 terms

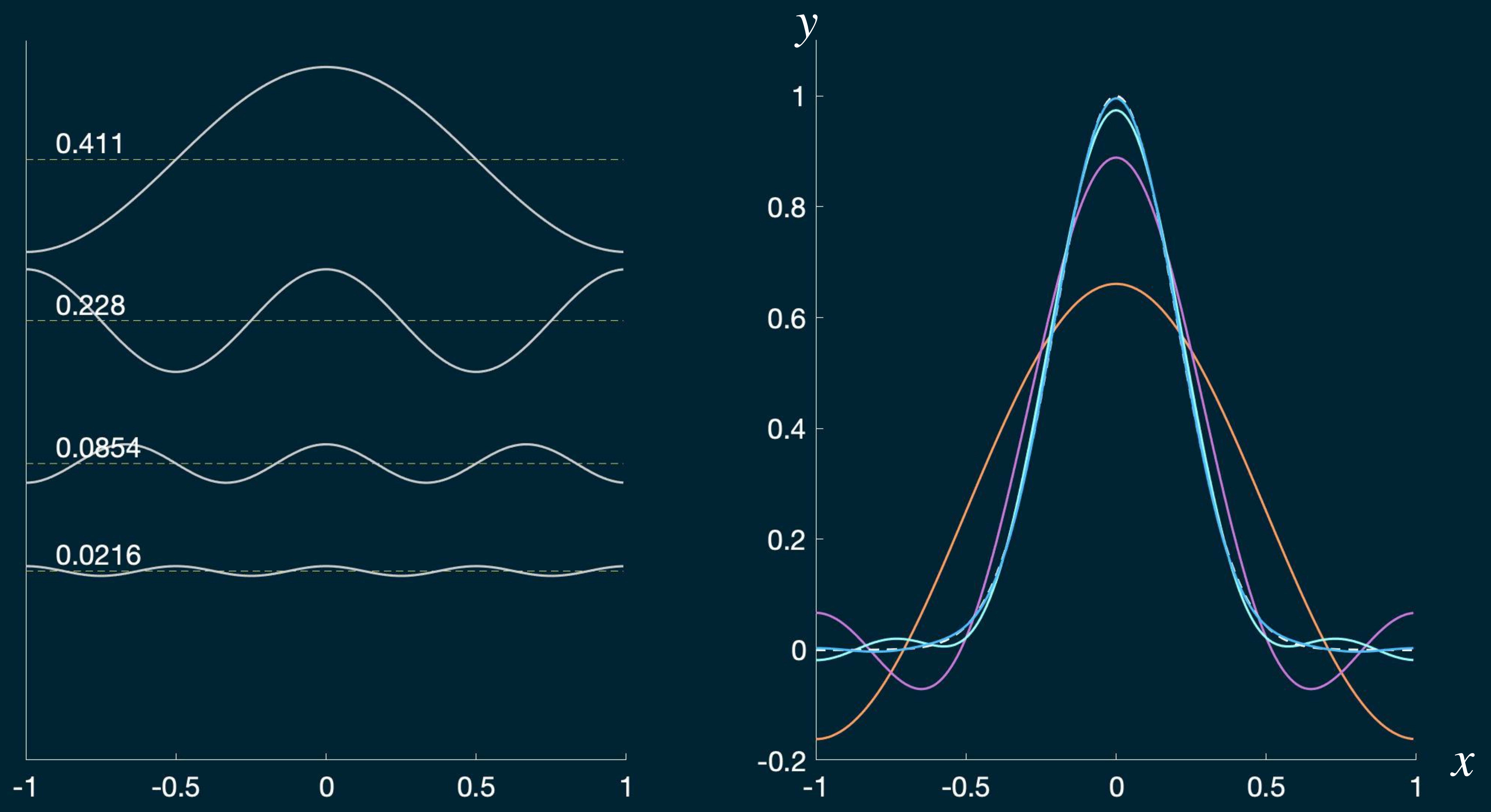


3 terms

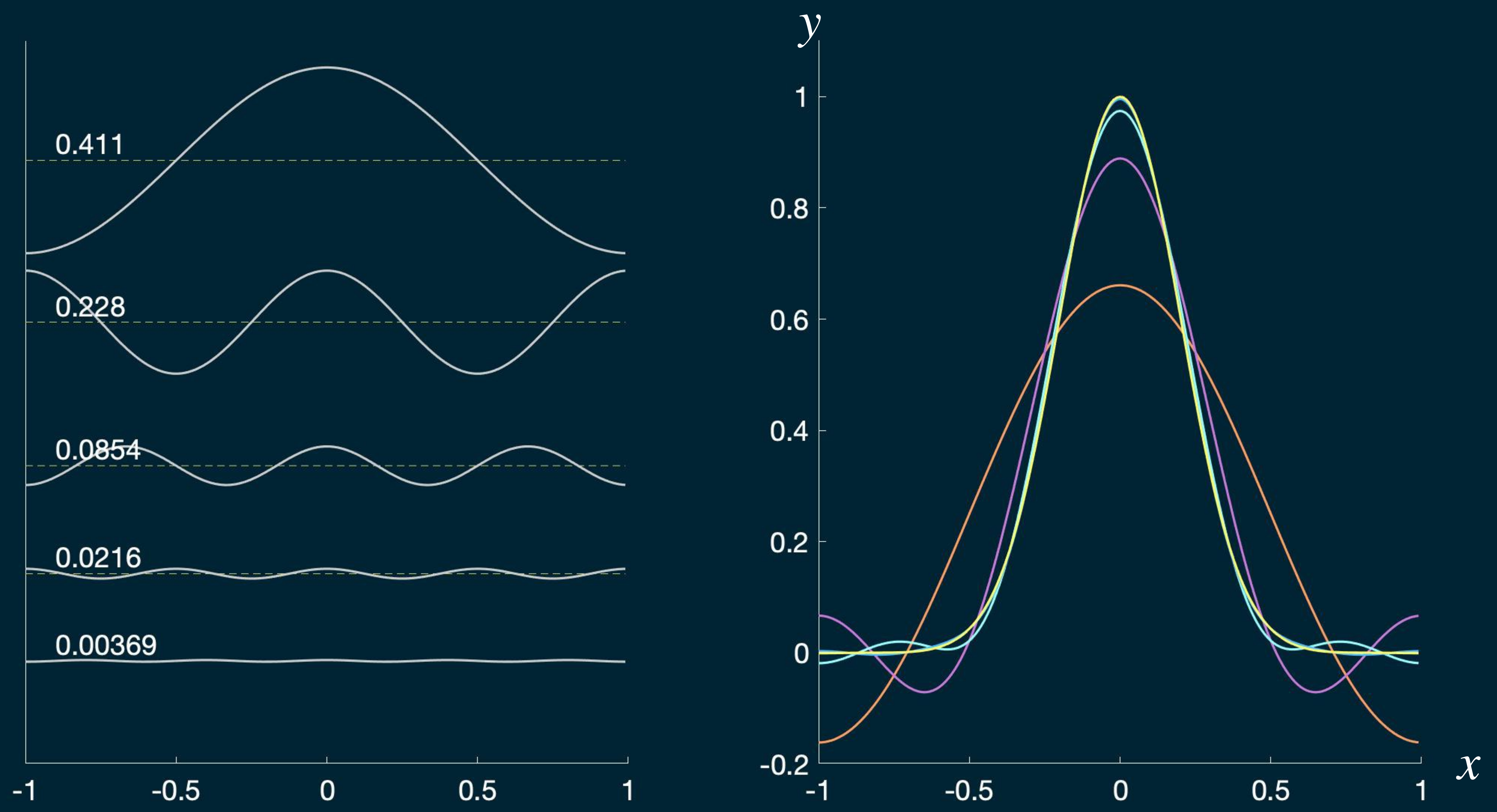




4 terms

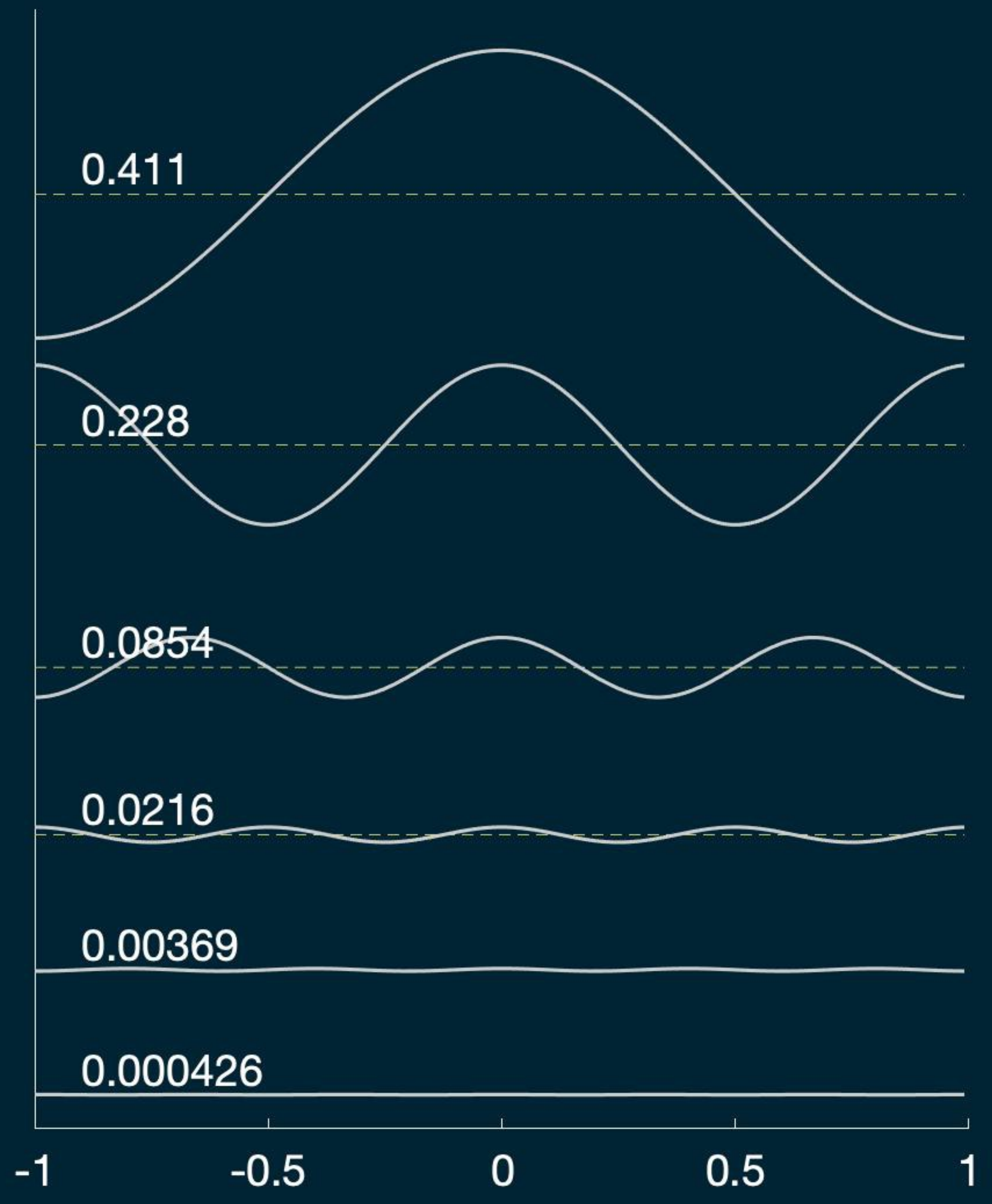


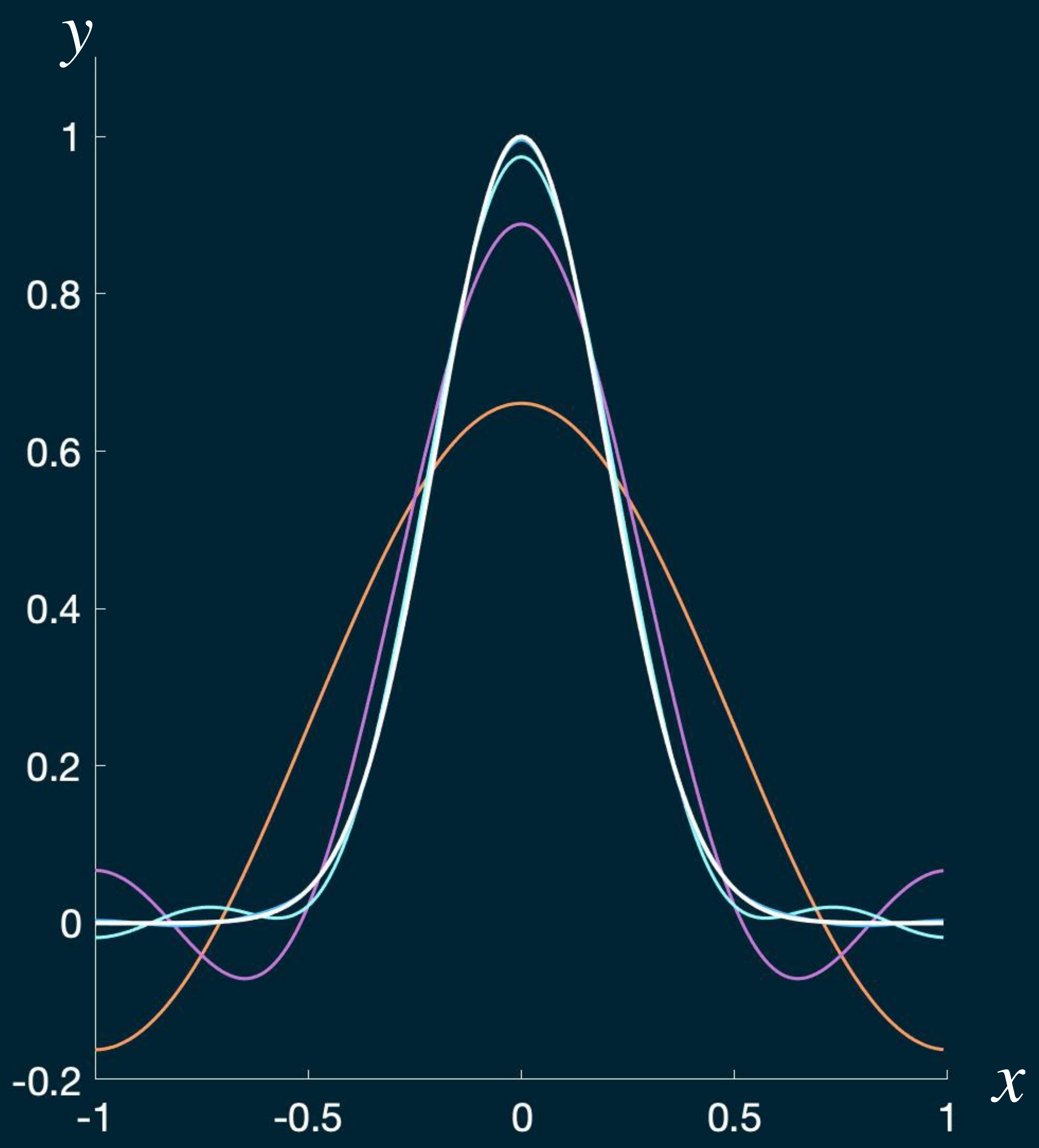
5 terms



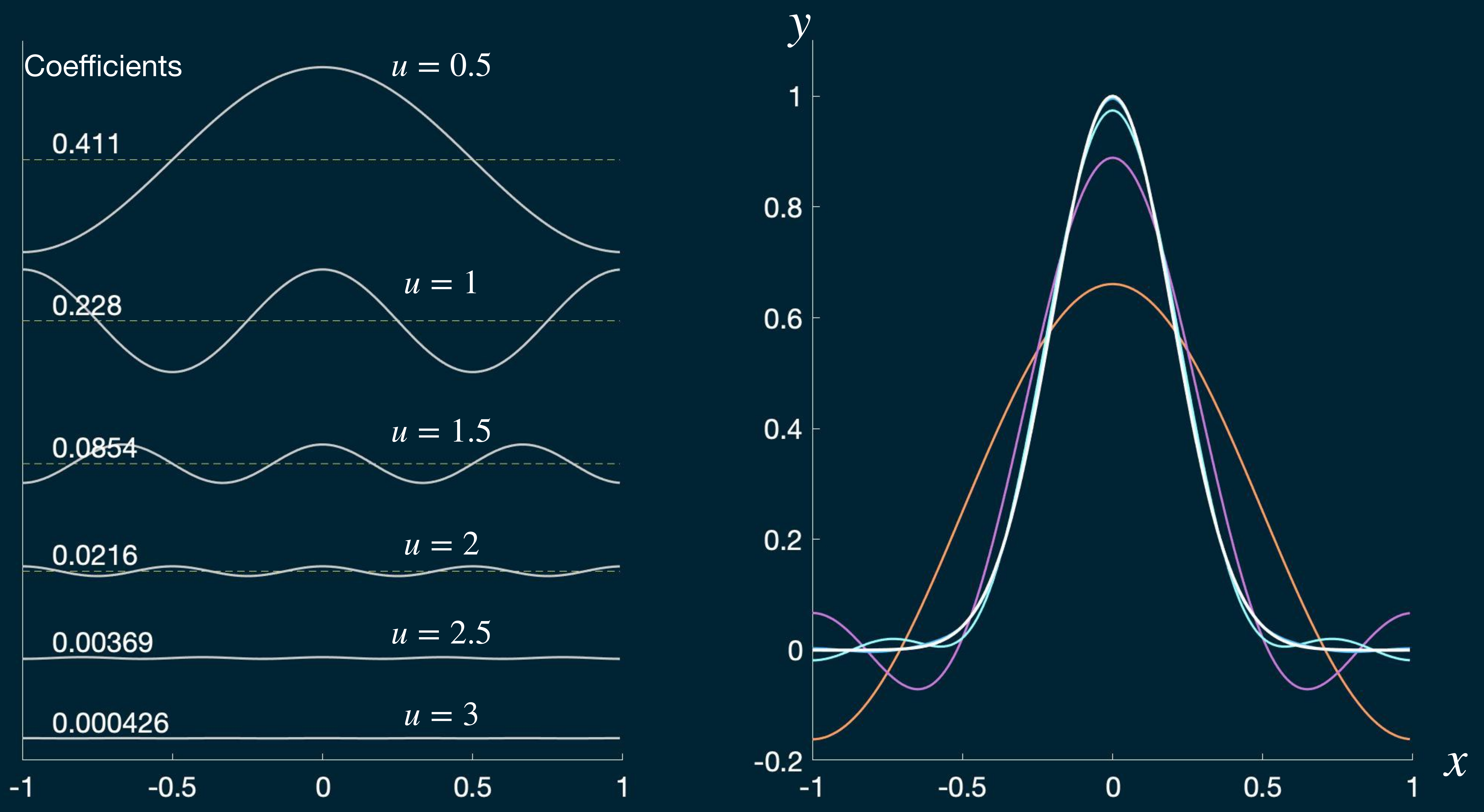
0.411
0.228
0.0854
0.0216
0.00369
-1
-0.5
0
0.5
1
y
1
0.8
0.6
0.4
0.2
0
-0.2
-1
-0.5
0
0.5
1
x

"Converged" at 6 terms

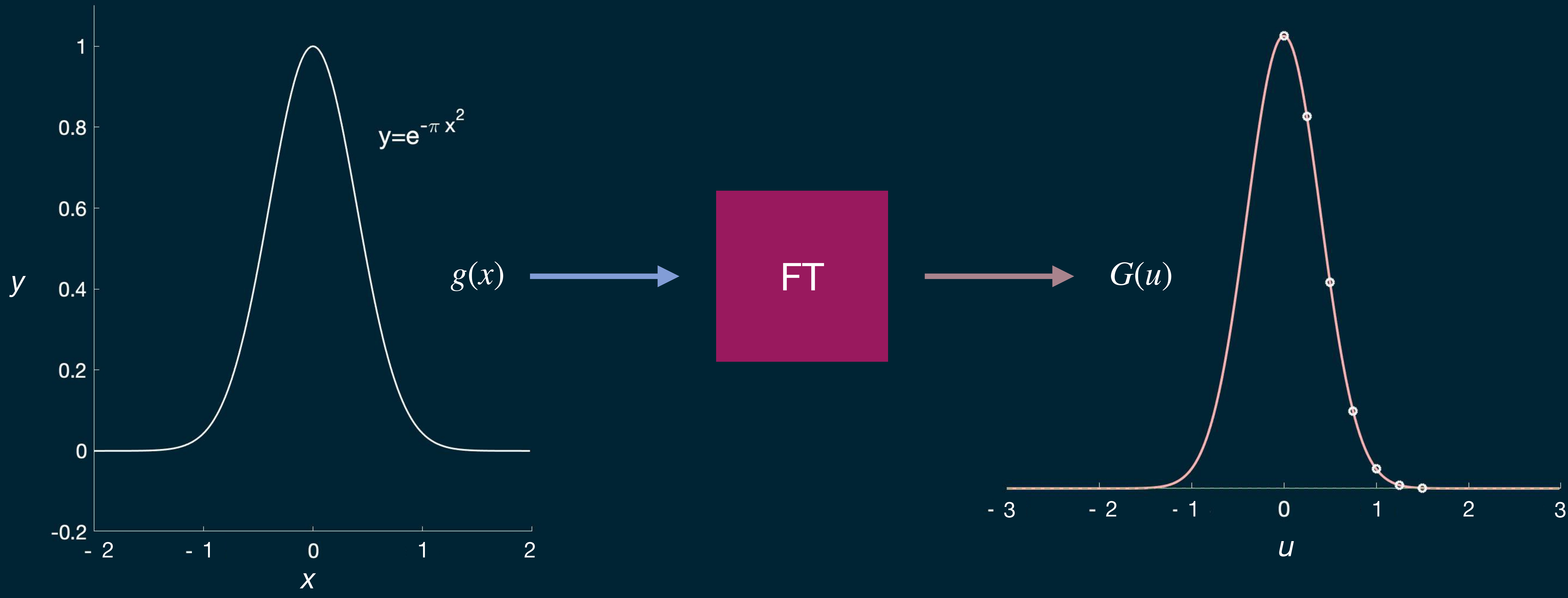




u is the frequency variable



The Fourier Transform gives us the coefficients



A transform pair $\boxed{e^{-\pi x^2} \rightarrow e^{-\pi u^2}}$

The formulas

Fourier transform

$$G(u) = \int g(x) e^{-i2\pi ux} dx$$

Inverse Fourier transform

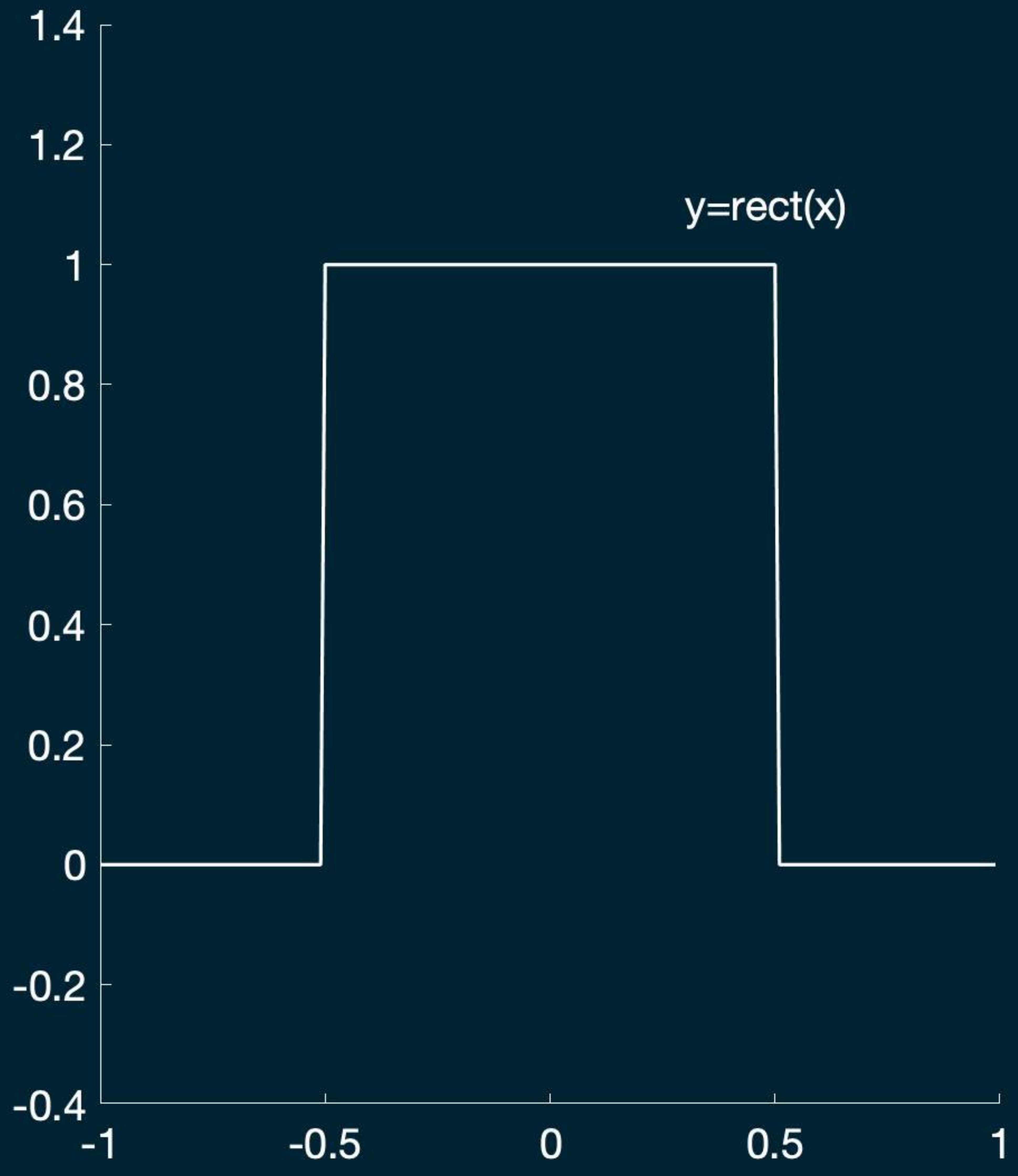
$$g(x) = \int G(u) e^{+i2\pi ux} du$$

Example:

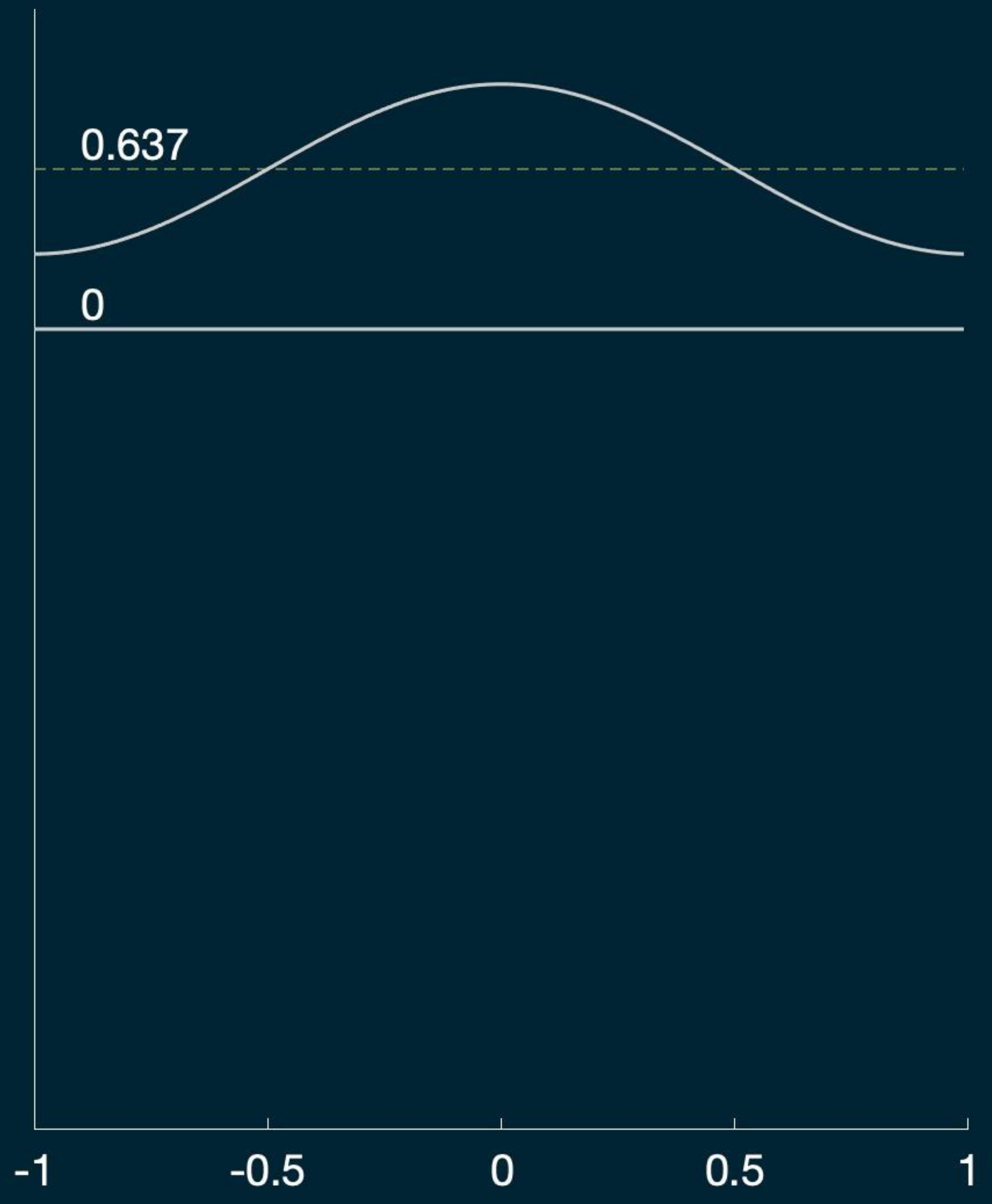
$$g(x) = e^{-\pi x^2}$$

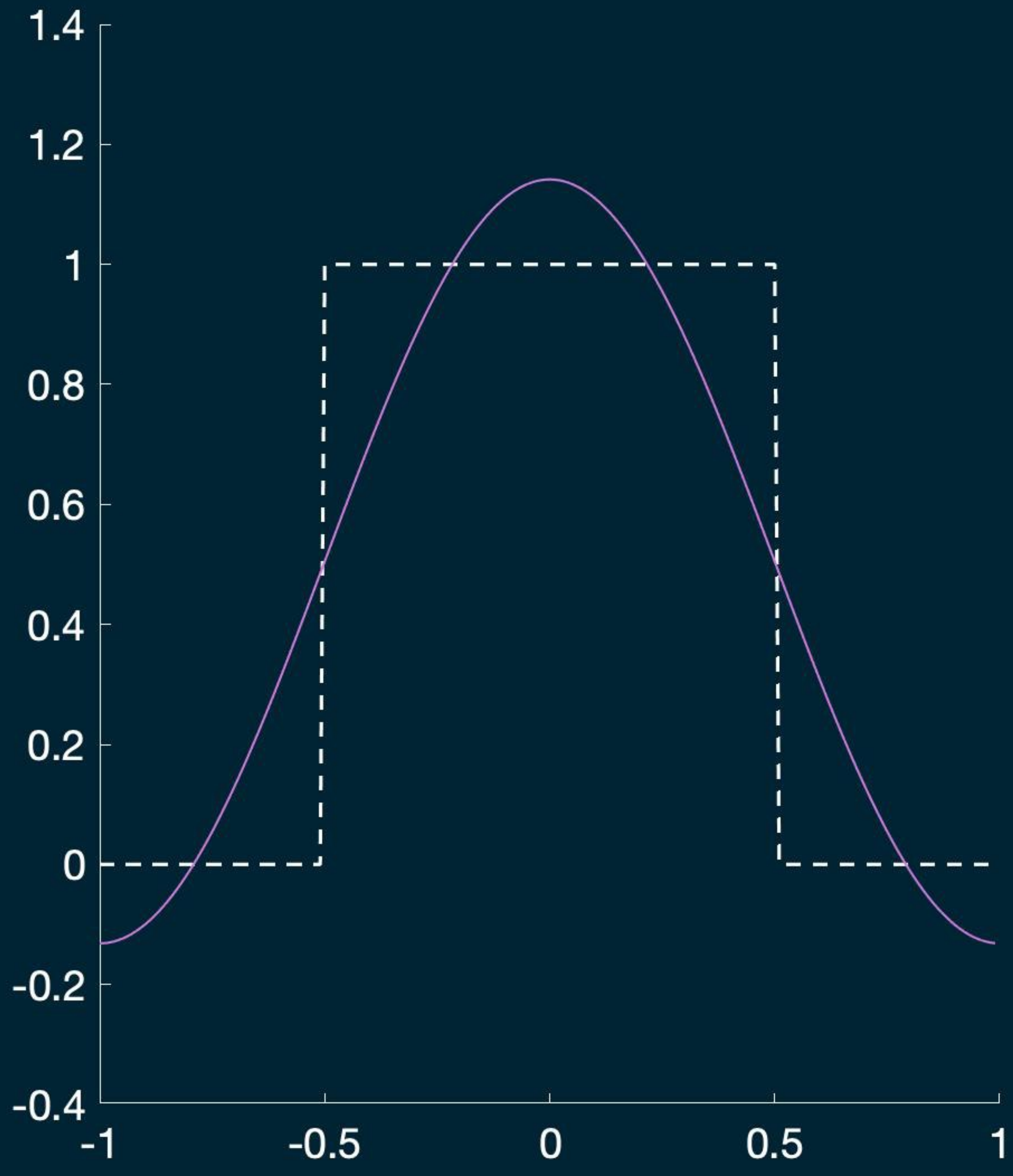
$$G(u) = e^{-\pi u^2}$$

Fourier reconstruction of a rectangular function

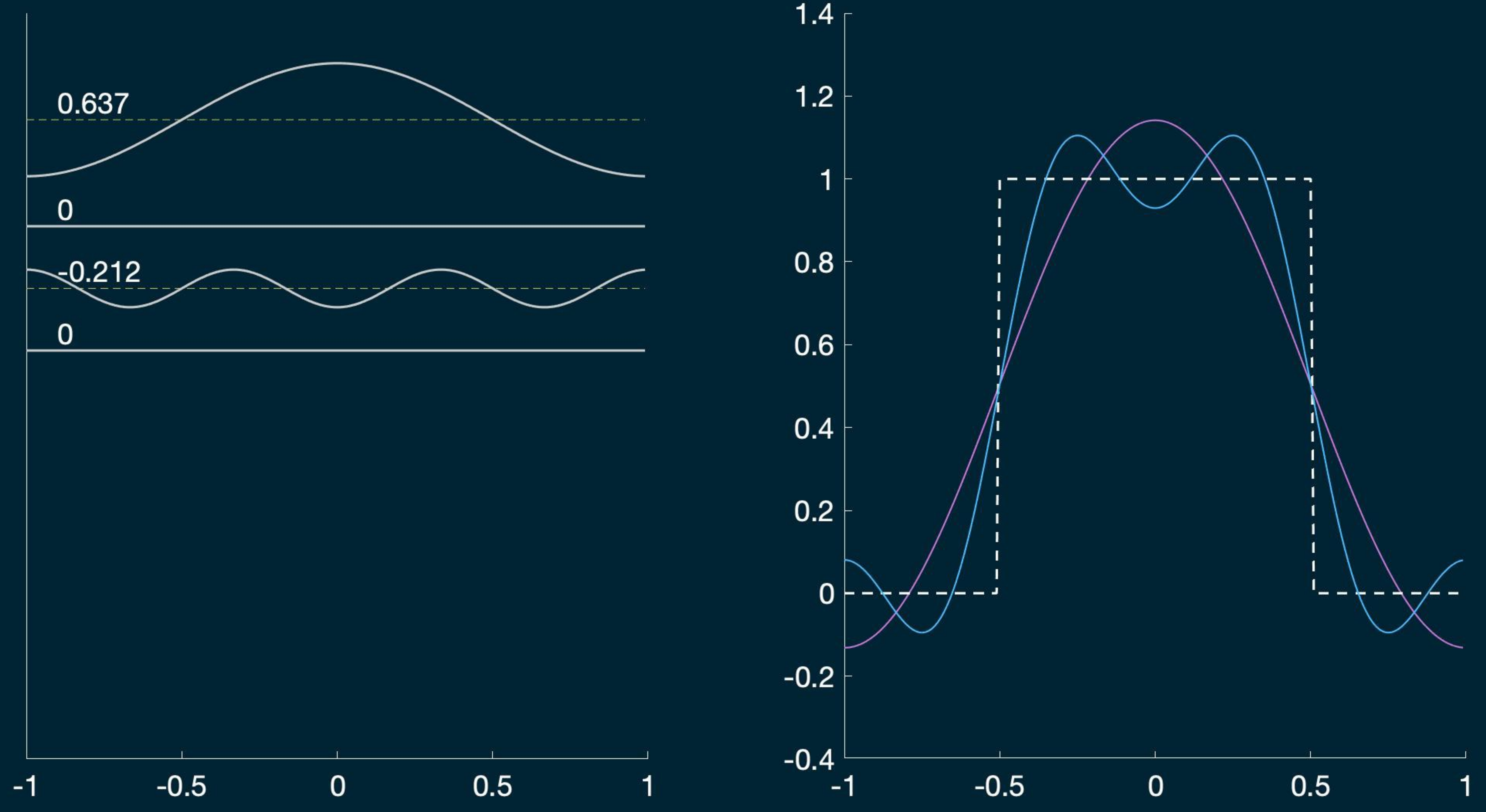


2 terms

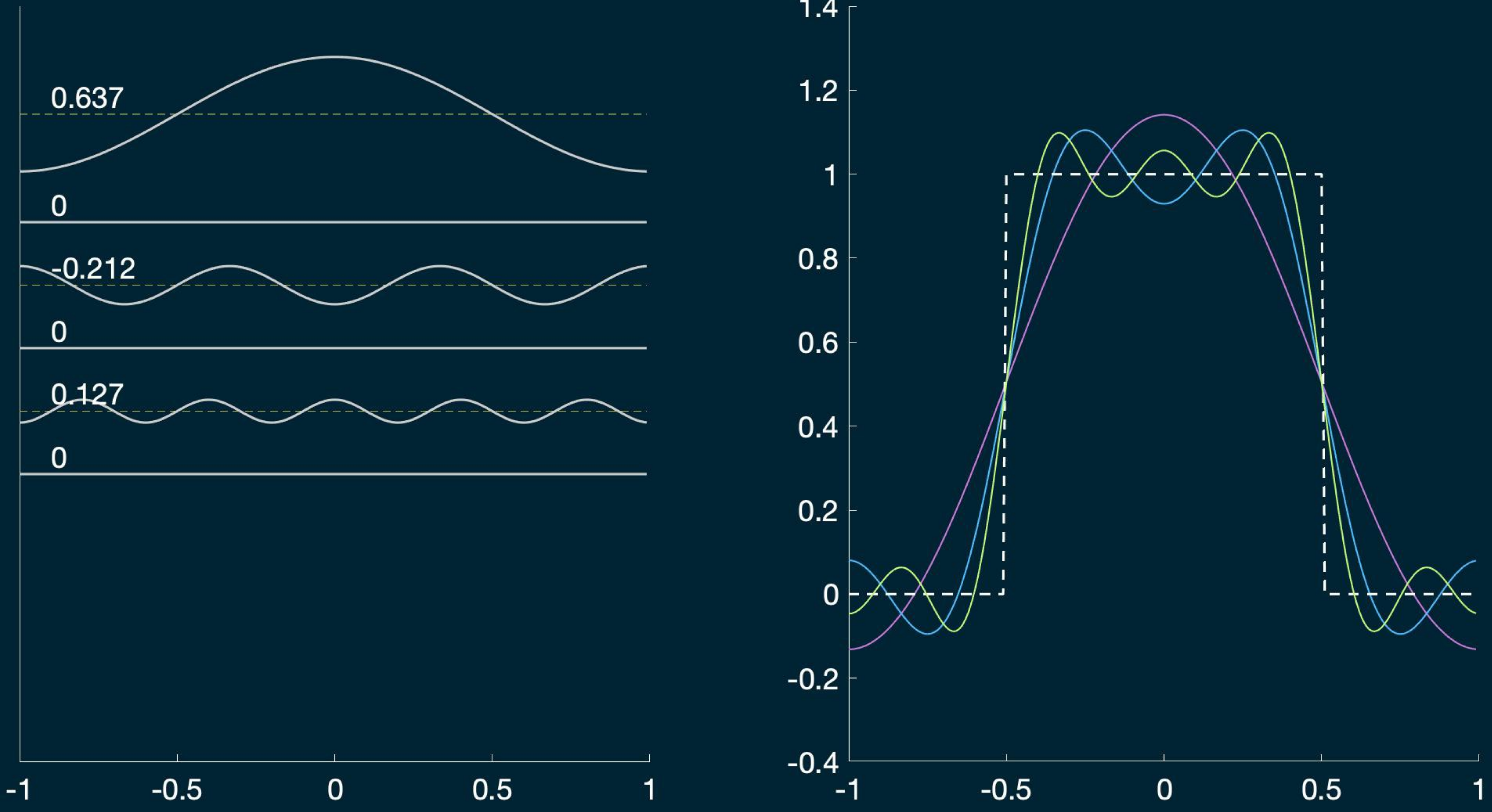




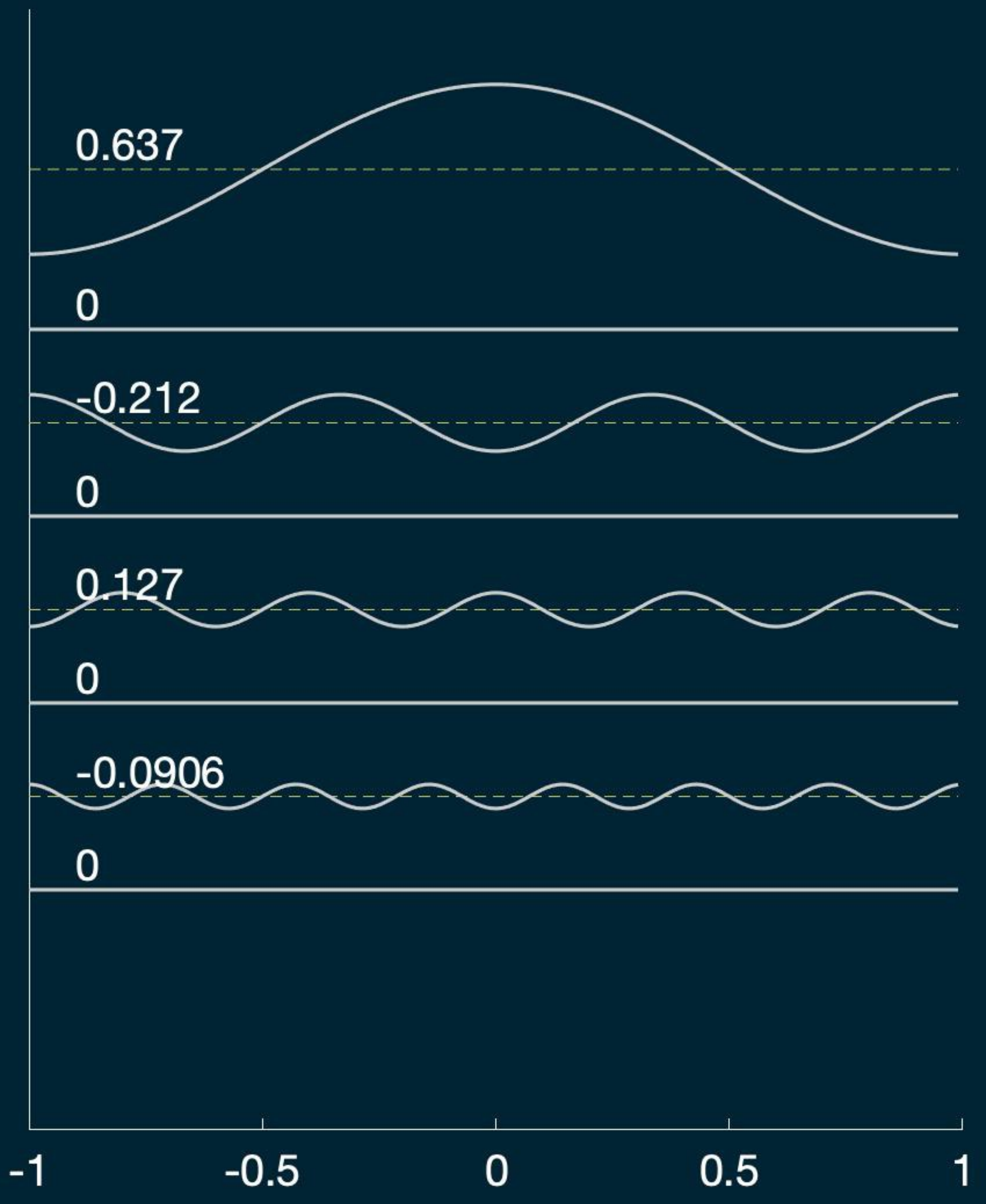
4 terms



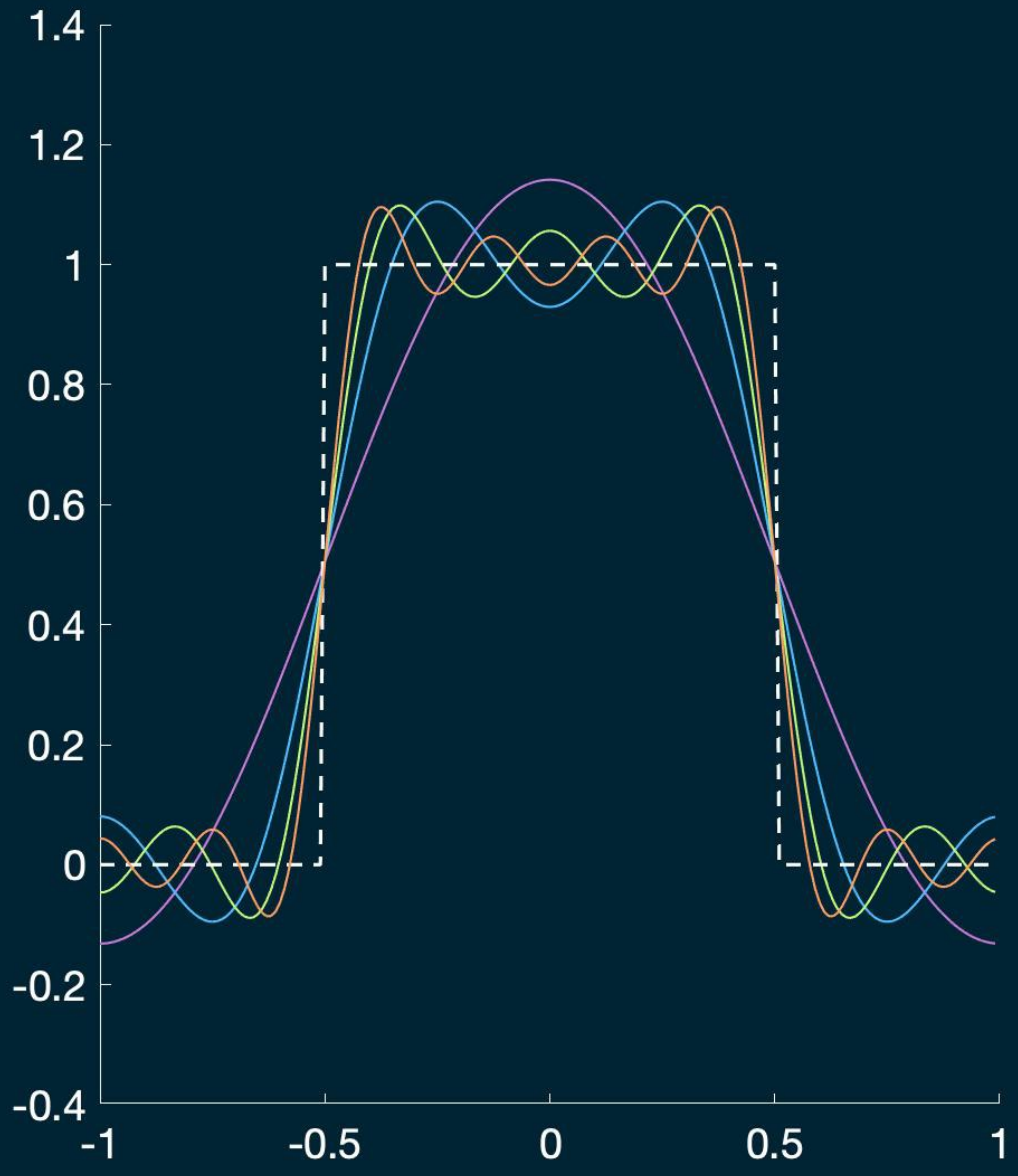
6 terms



8 terms

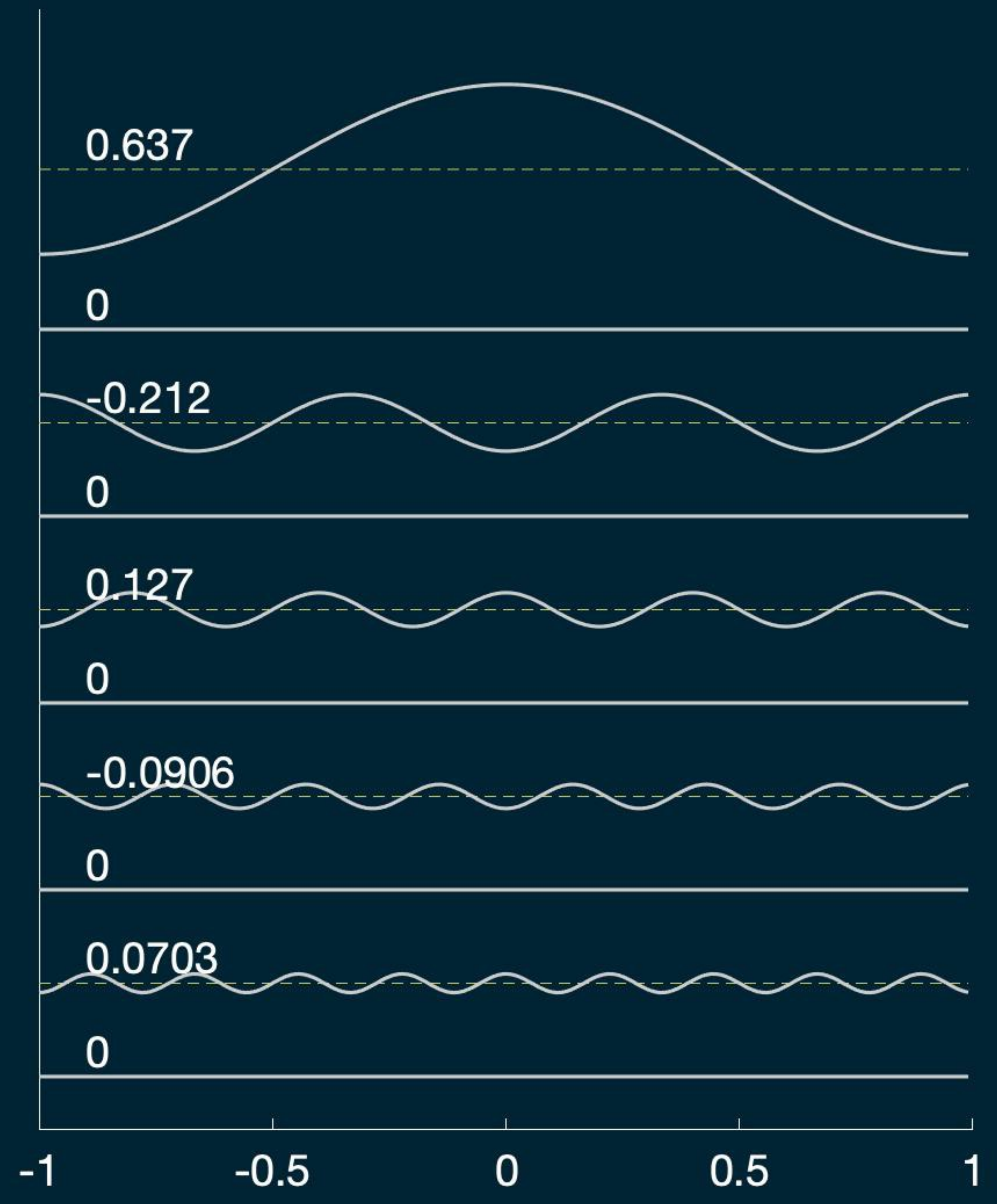


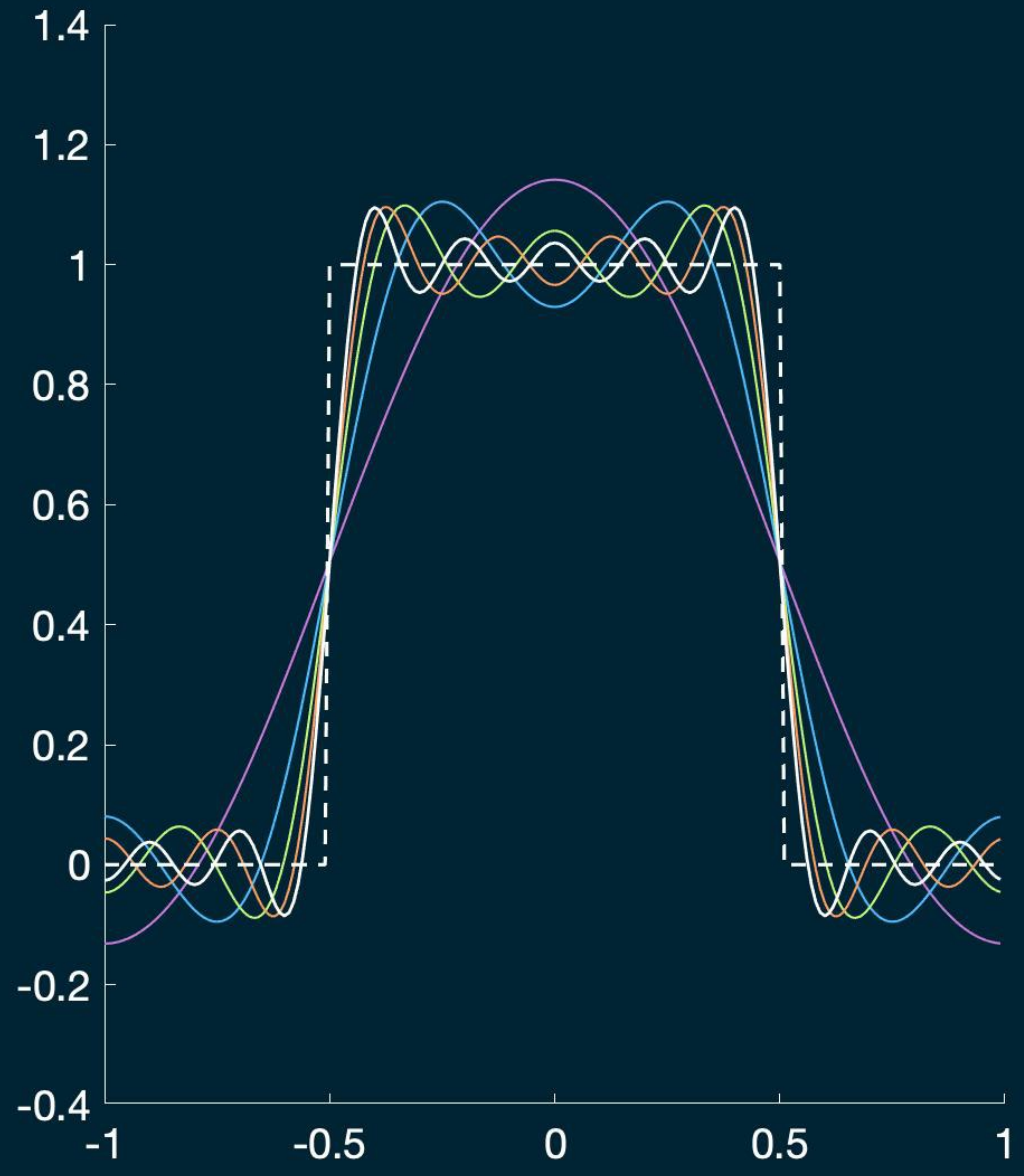
0.637
0
-0.212
0
0.127
0
-0.0906
0
-1
-0.5
0
0.5
1



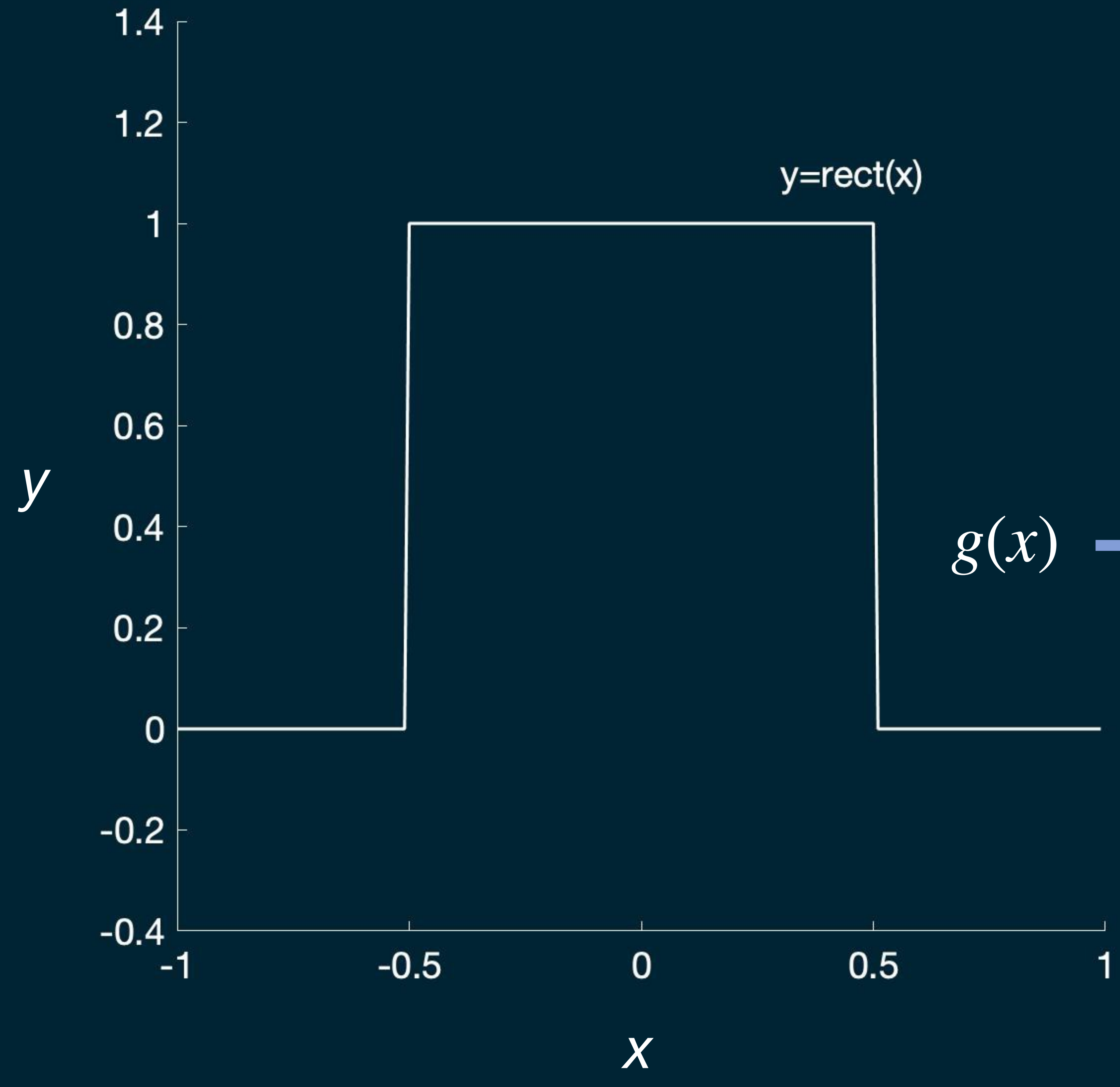
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-1
-0.5
0
0.5
1

Nowhere near convergence at 10 terms

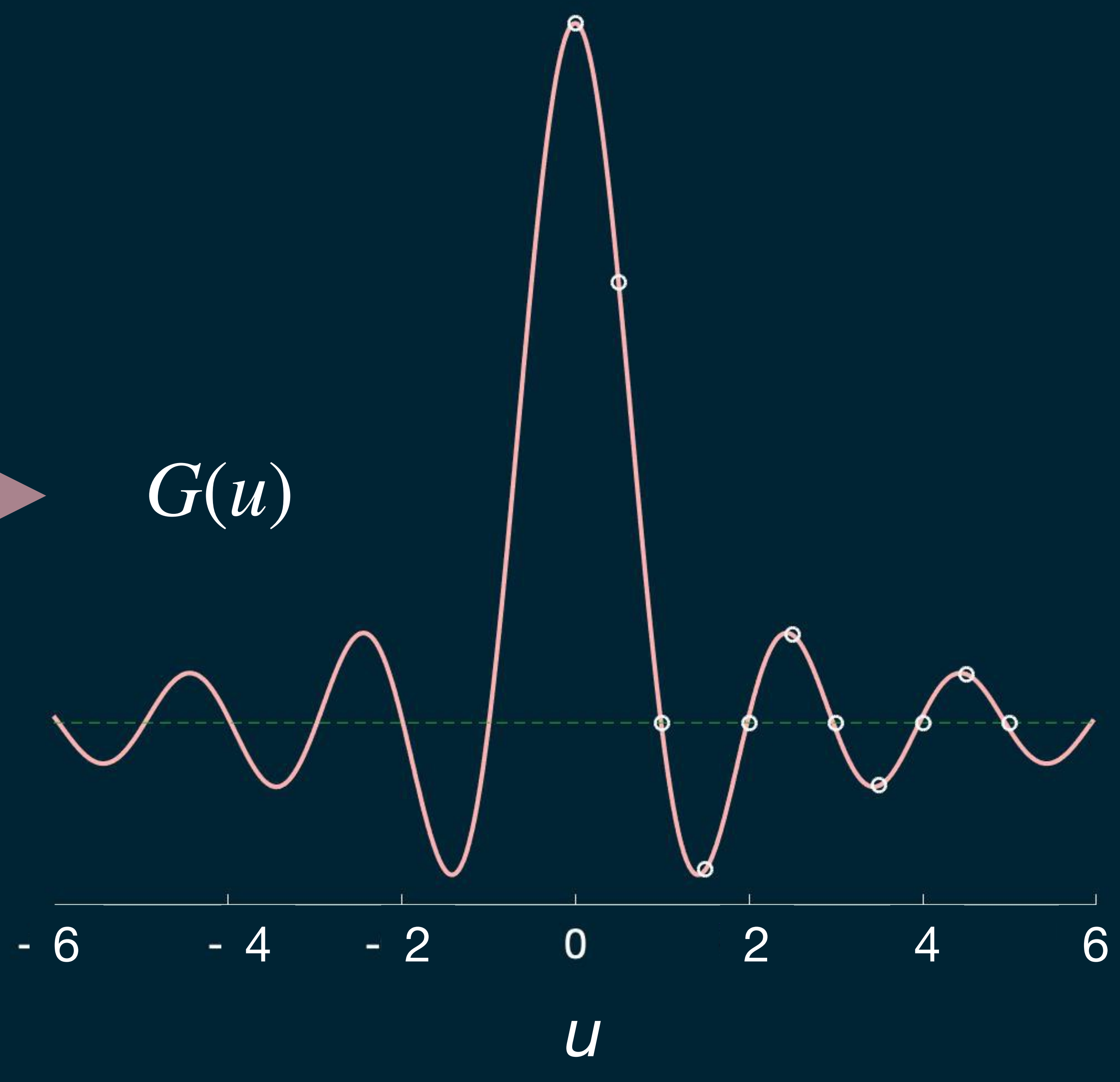




The Fourier Transform of rect(x) is sinc(u)



$g(x)$ → FT → $G(u)$

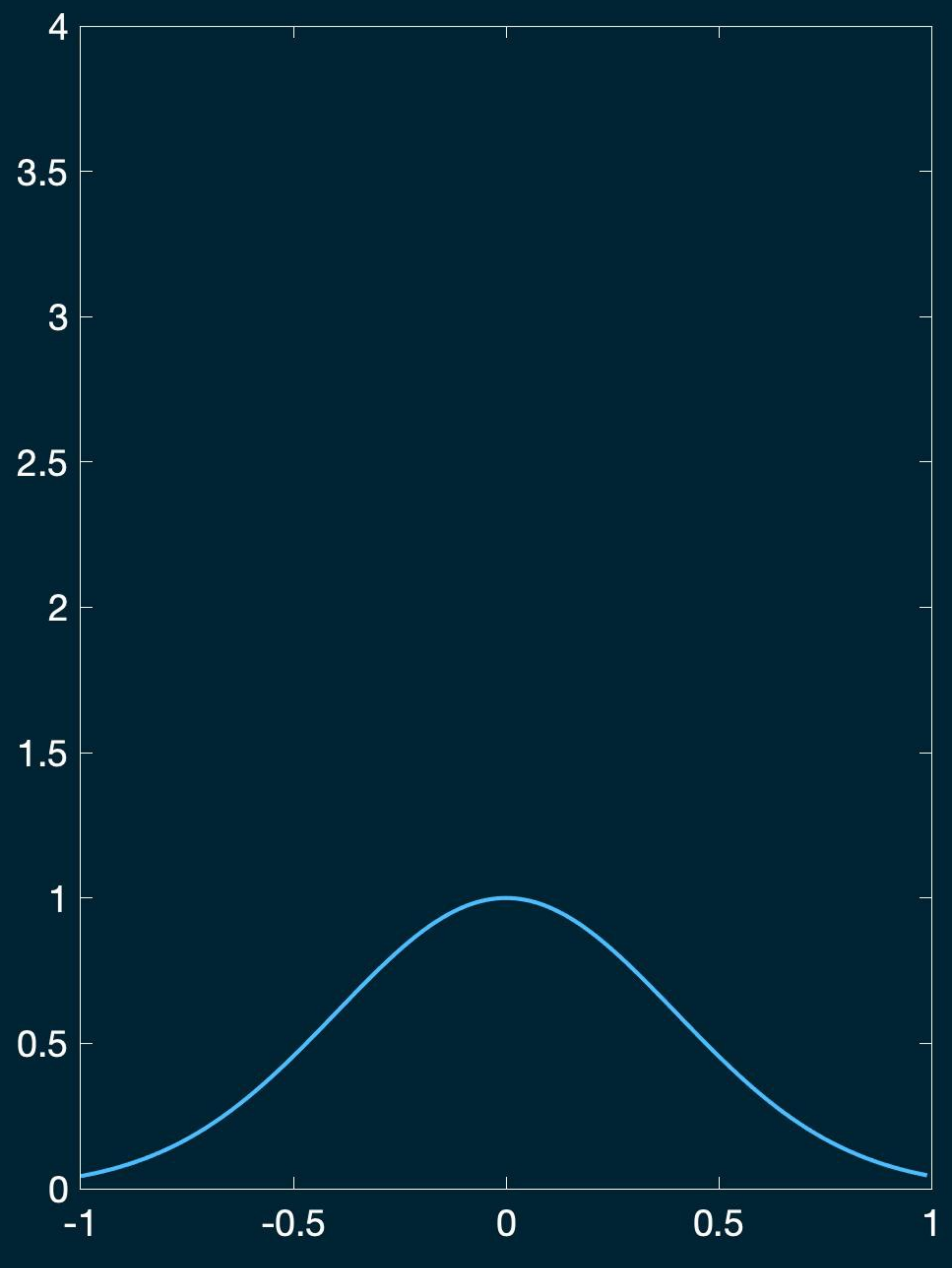


$$\mathrm{rect(x)} \rightarrow \frac{\sin(\pi \mathrm{u})}{\pi \mathrm{u}}$$

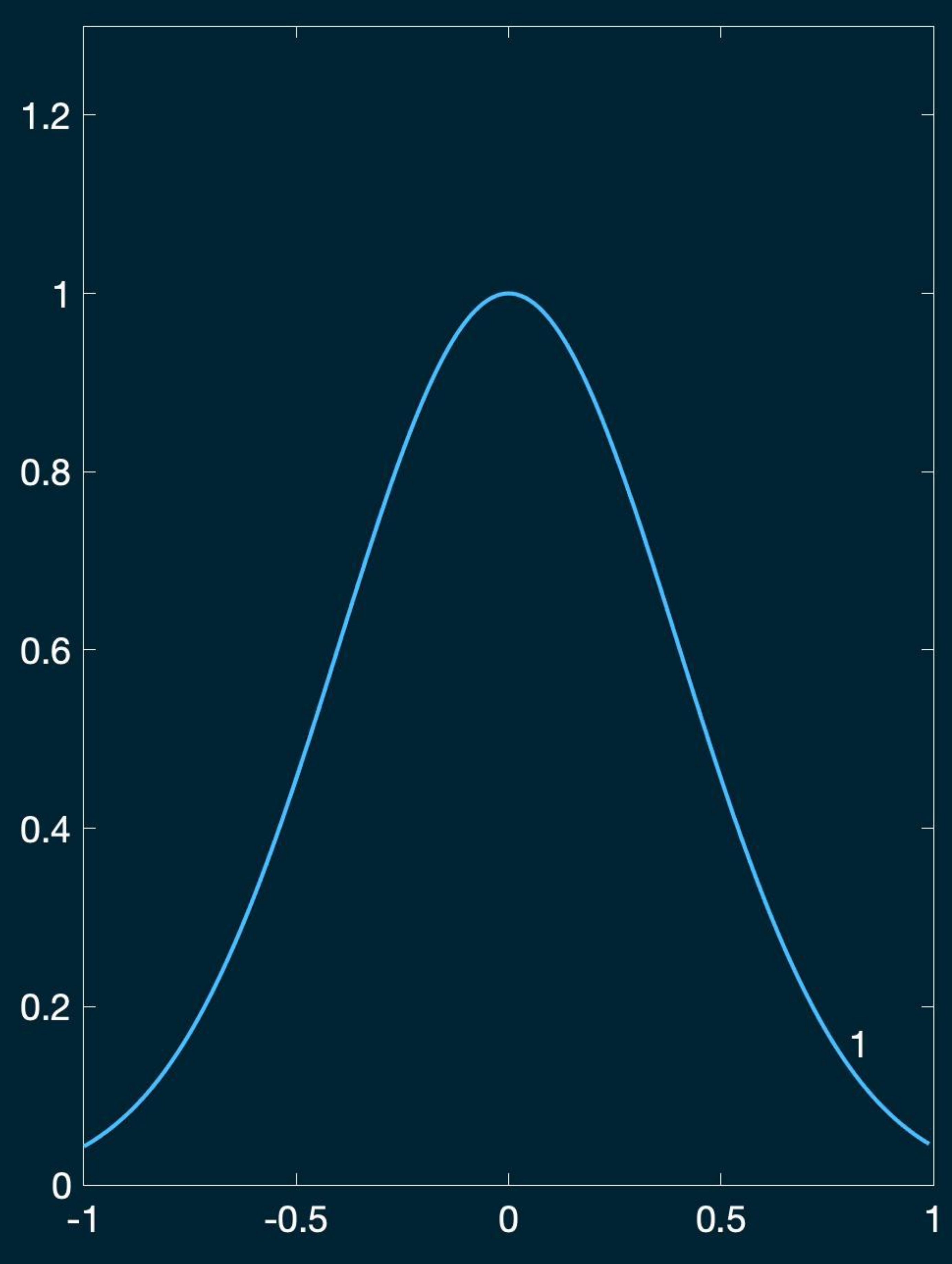
$\dfrac{\sin(\pi u)}{\pi u}$ is also known as: $\mathrm{sinc}(u)$

Reciprocal scaling of FT pairs

$$g(x) = ae^{-\pi(ax)^2}$$

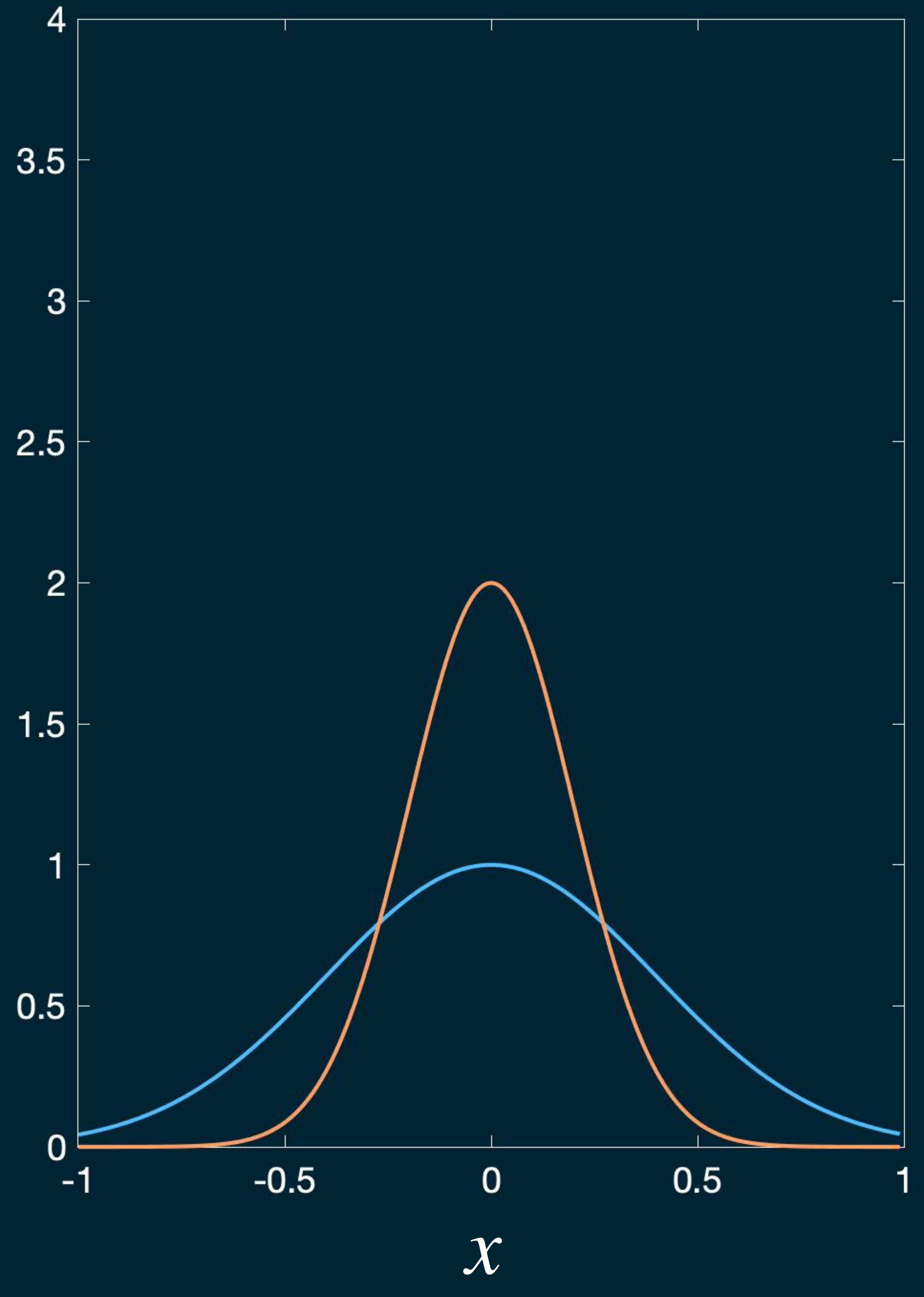


$$G(u) = e^{-\pi(u/a)^2}$$

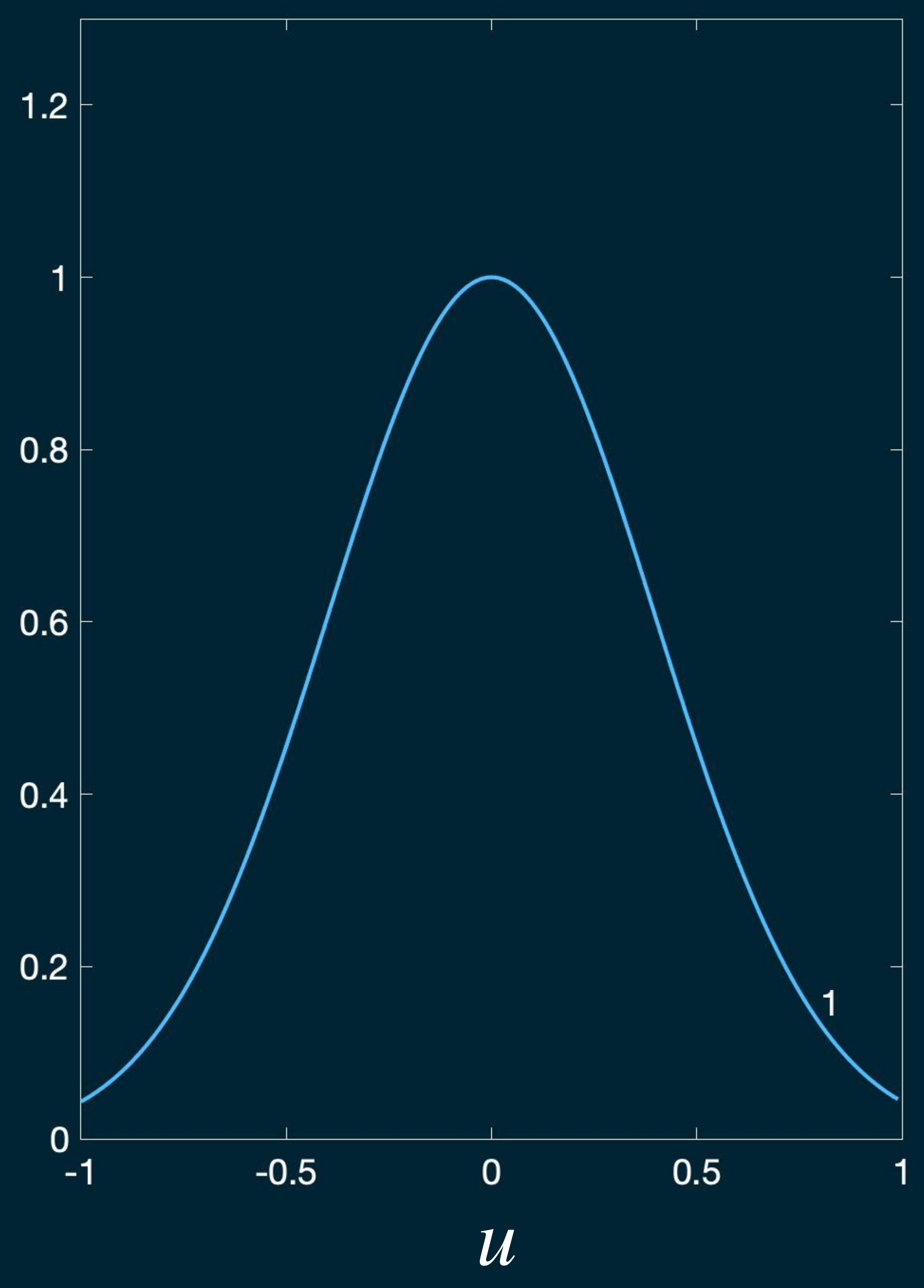


Reciprocal scaling of FT pairs

$$g(x) = ae^{-\pi(ax)^2}$$

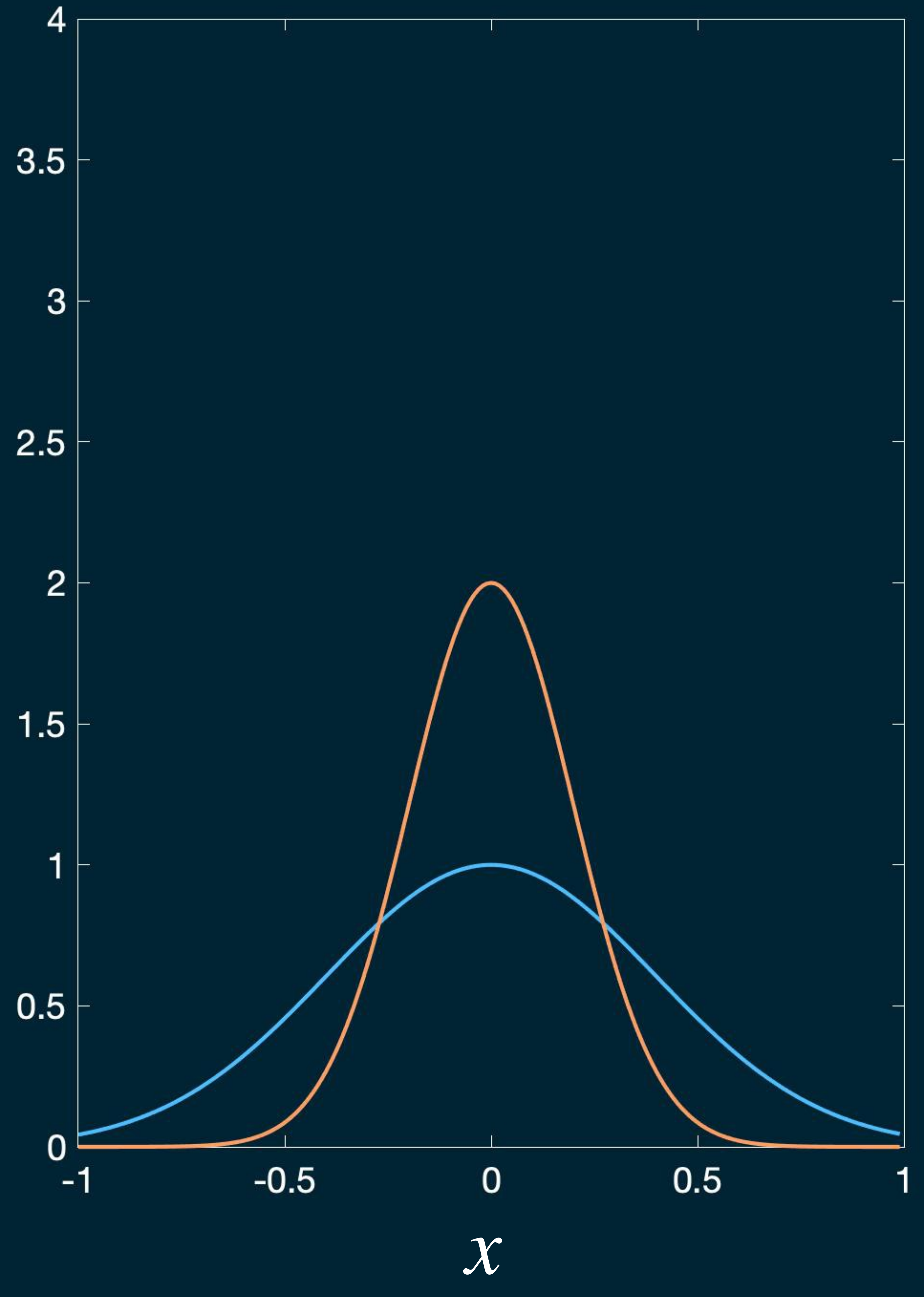


$$G(u) = e^{-\pi(u/a)^2}$$

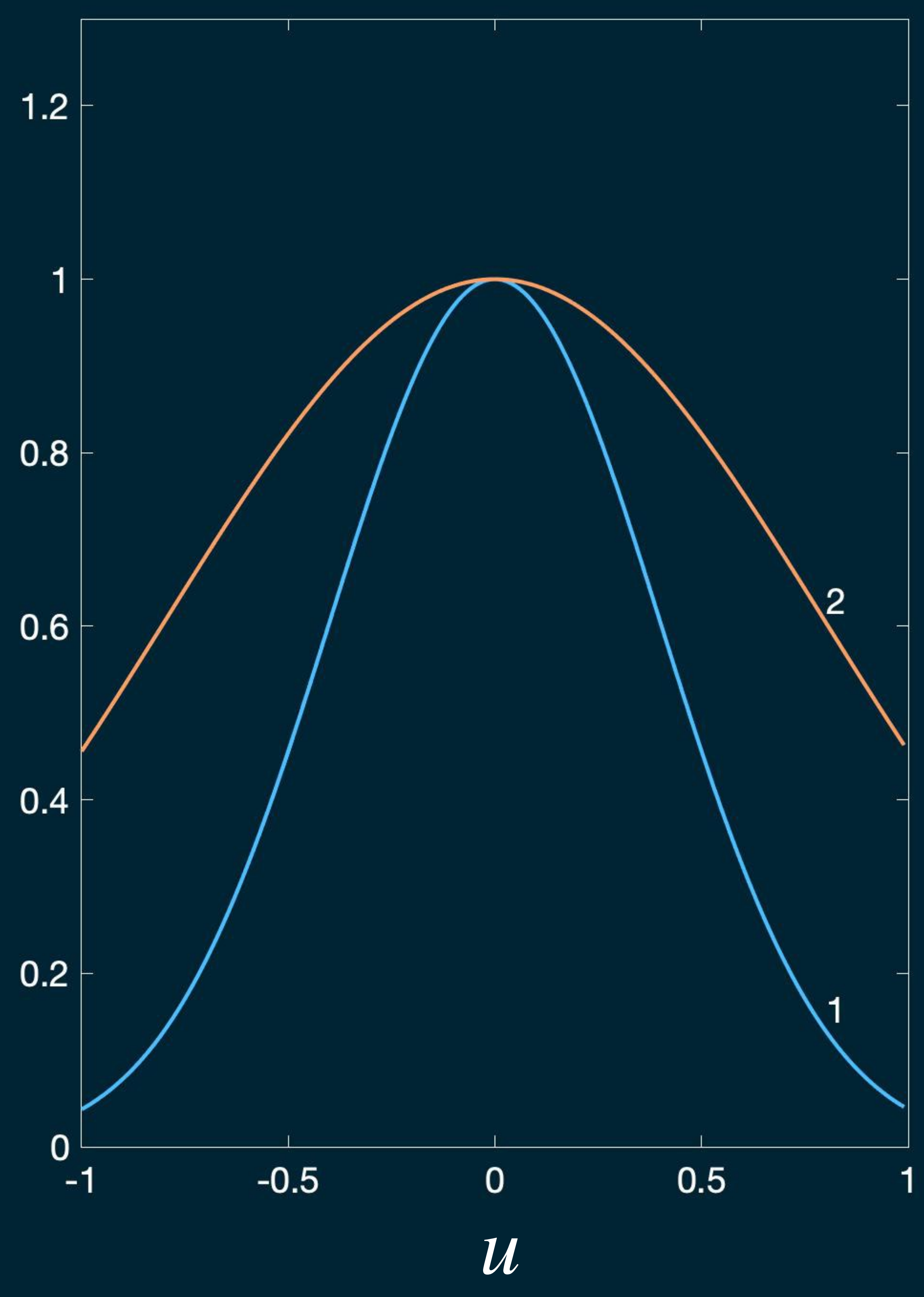


Reciprocal scaling of FT pairs

$$g(x) = ae^{-\pi(ax)^2}$$

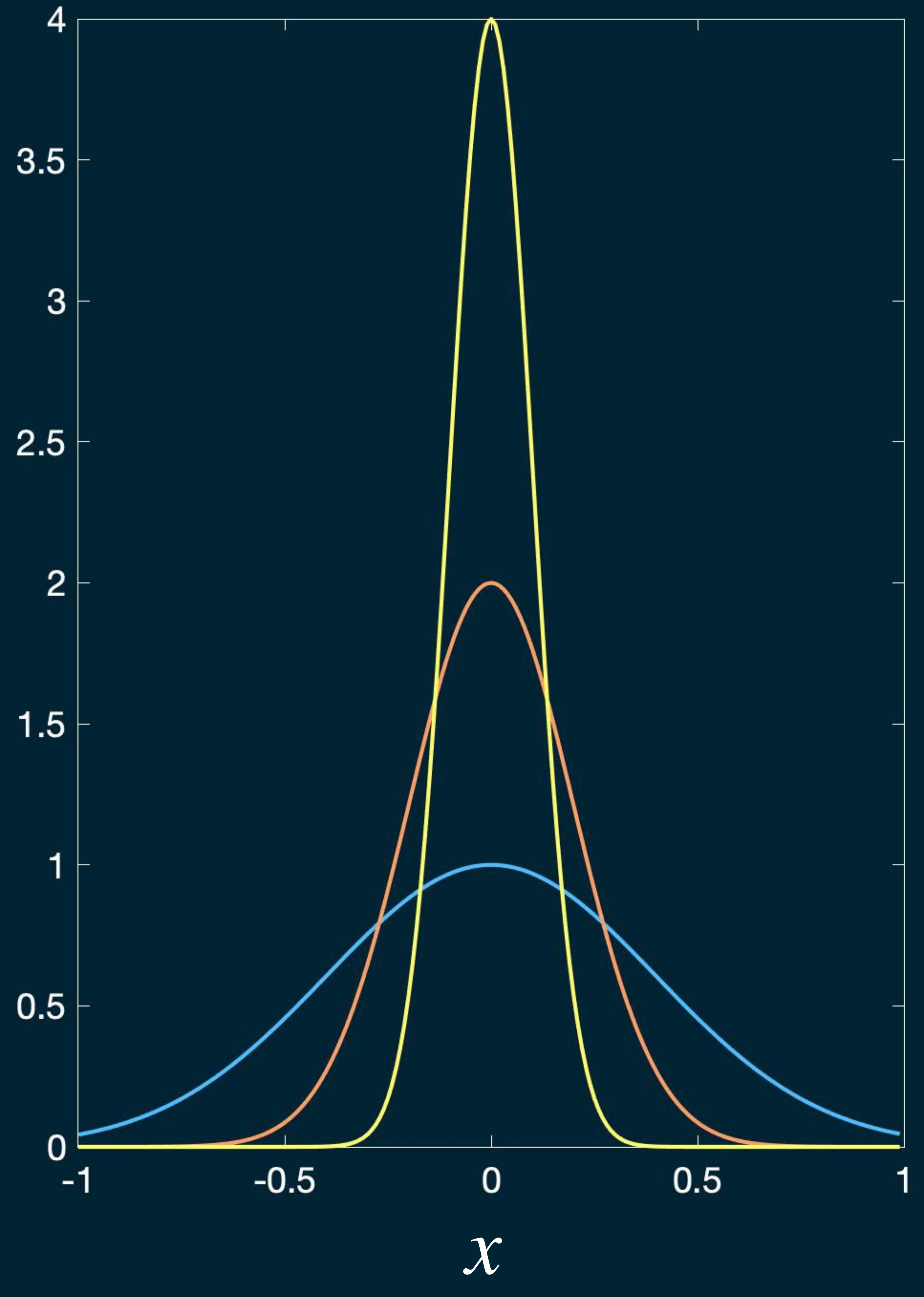


$$G(u) = e^{-\pi(u/a)^2}$$

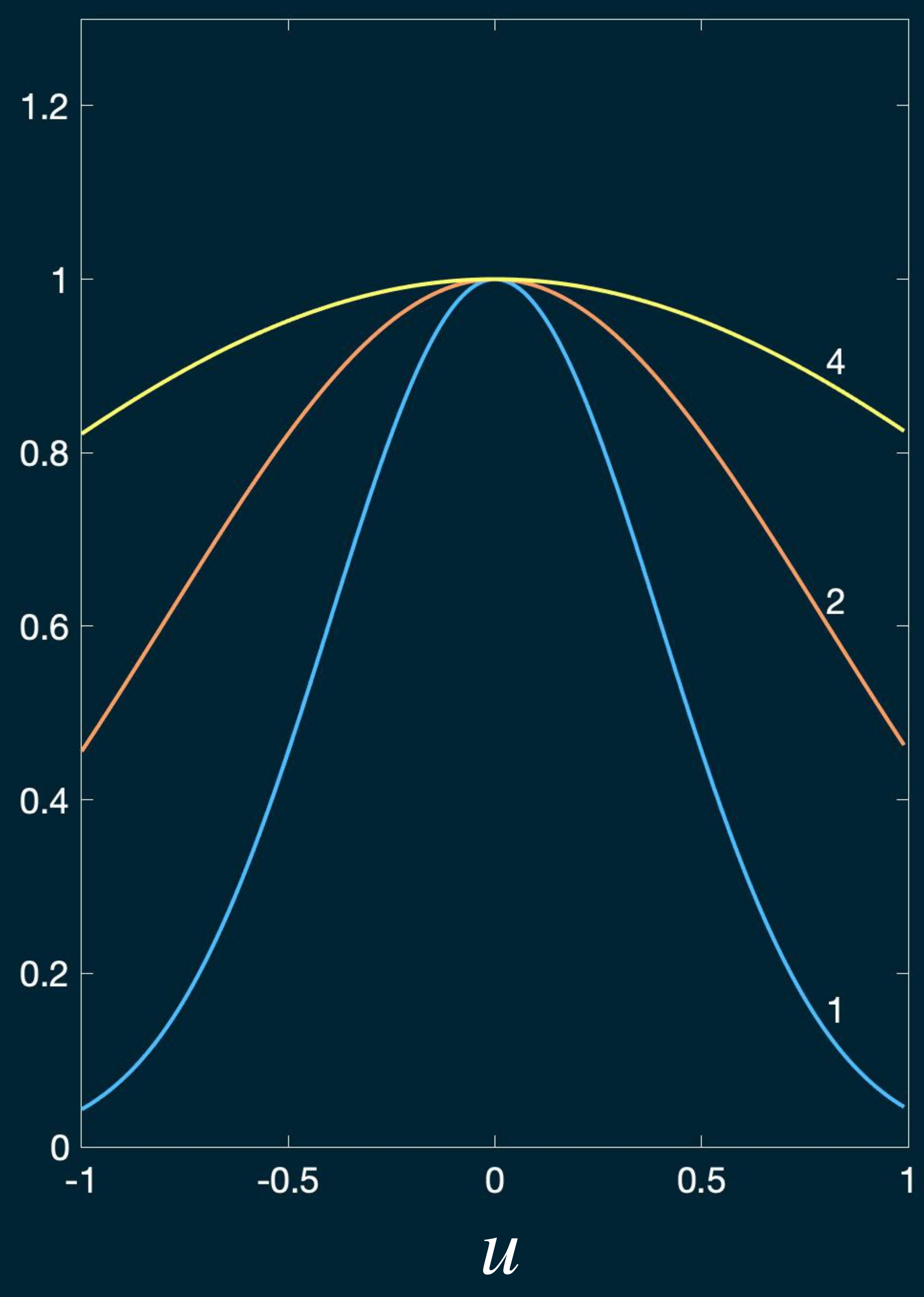


Reciprocal scaling of FT pairs

$$g(x) = ae^{-\pi(ax)^2}$$



$$G(u) = e^{-\pi(u/a)^2}$$

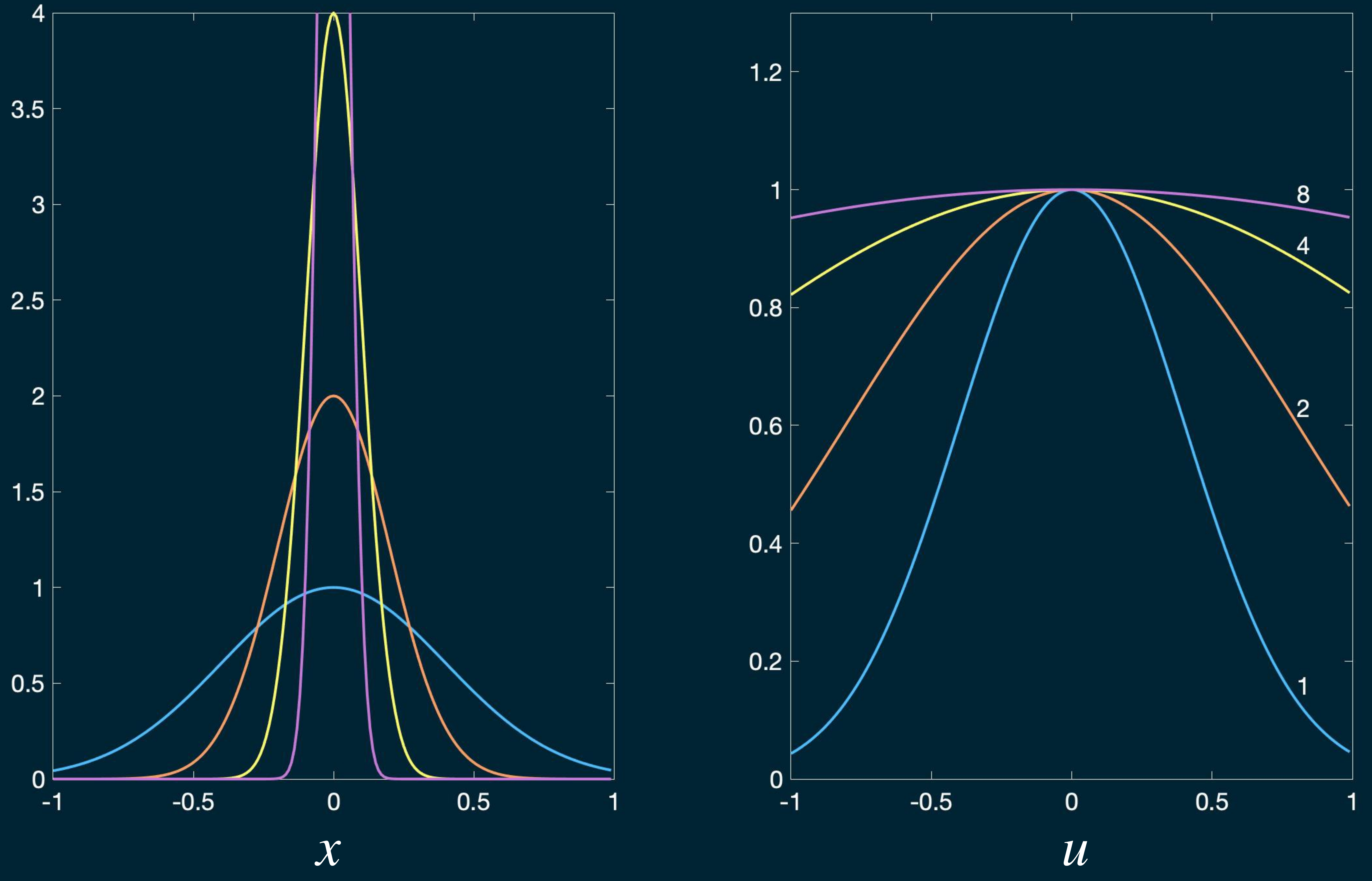


Scale property

$$ag(ax) \rightarrow G(u/a)$$

Reciprocal scaling of FT pairs

$$g(x) = ae^{-\pi(ax)^2} \qquad G(u) = e^{-\pi(u/a)^2}$$

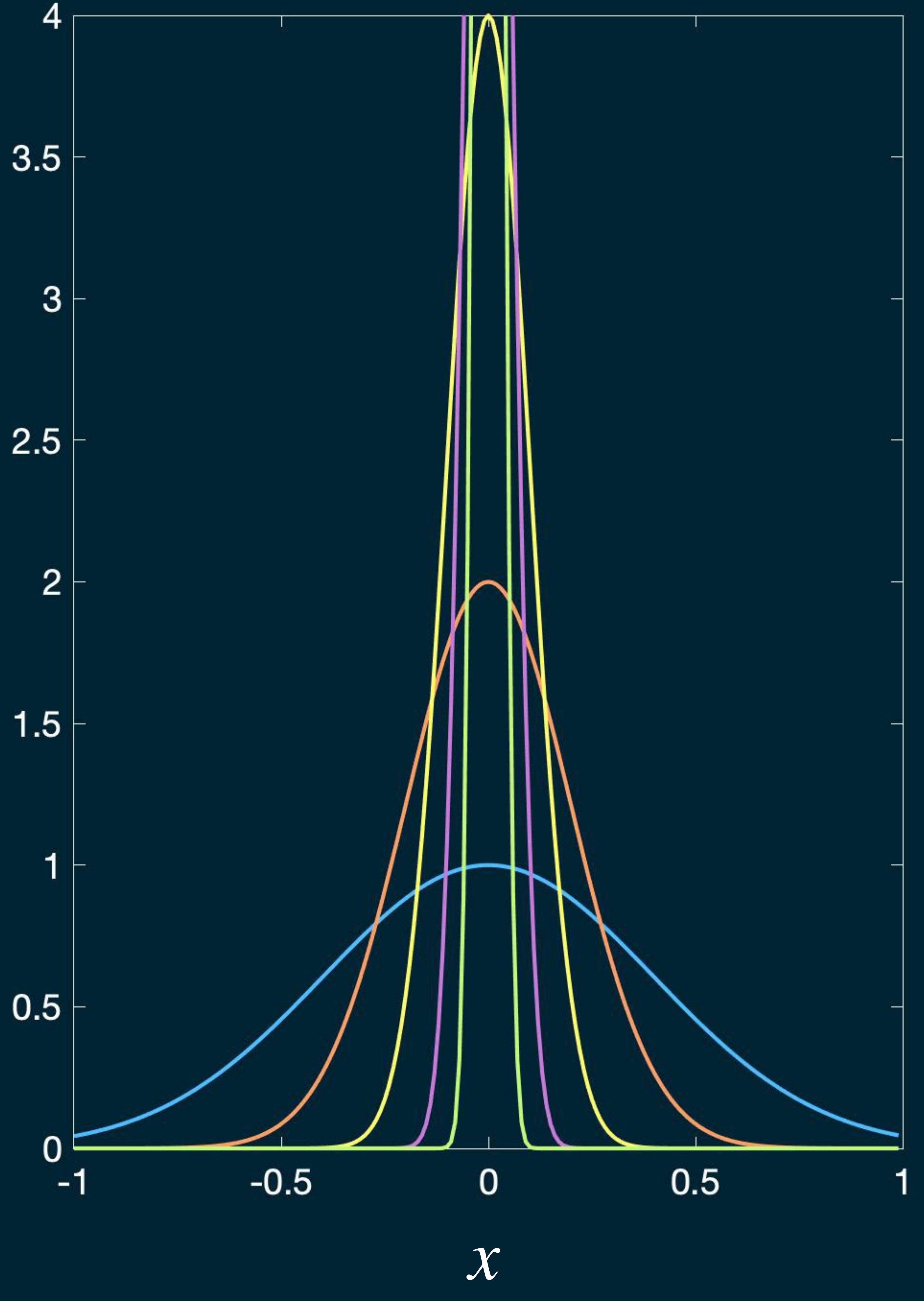


Scale property

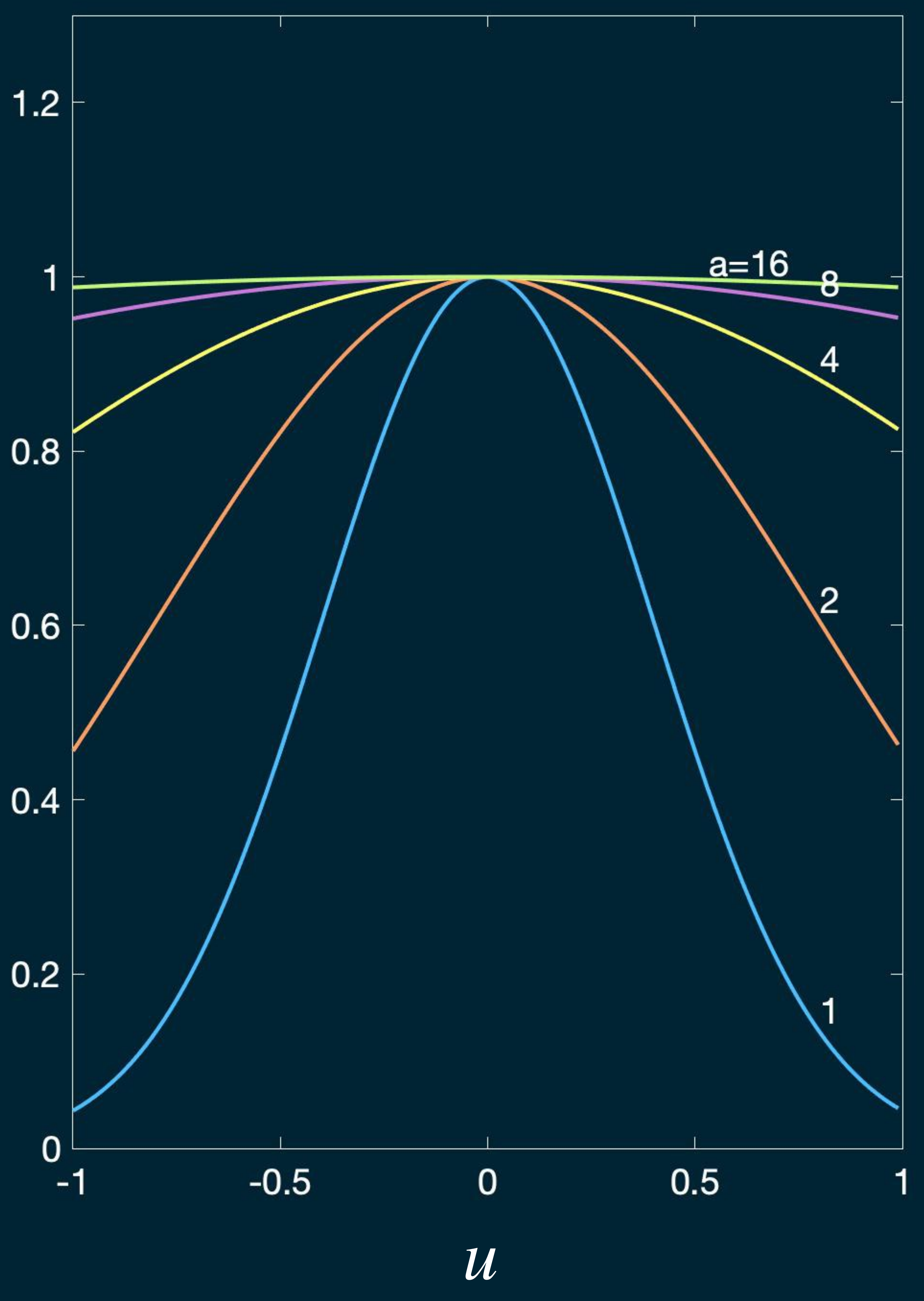
$$ag(ax) \rightarrow G(u/a)$$

Reciprocal scaling of FT pairs

$$g(x) = ae^{-\pi(ax)^2}$$



$$G(u) = e^{-\pi(u/a)^2}$$



Scale property

$$ag(ax) \rightarrow G(u/a)$$

Delta function

$$\delta(x) = \lim_{a\to\infty} ae^{-\pi(ax)^2}$$

FT Pair

$$\delta(x) \rightarrow 1$$

Fourier transform pairs

$$e^{-\pi x^2} \rightarrow e^{-\pi u^2}$$

$$\mathrm{rect}(\mathrm{x}) \rightarrow \frac{\sin(\pi \mathrm{u})}{\pi \mathrm{u}}$$

$$\delta(x) \rightarrow 1$$

1D Fourier transform properties

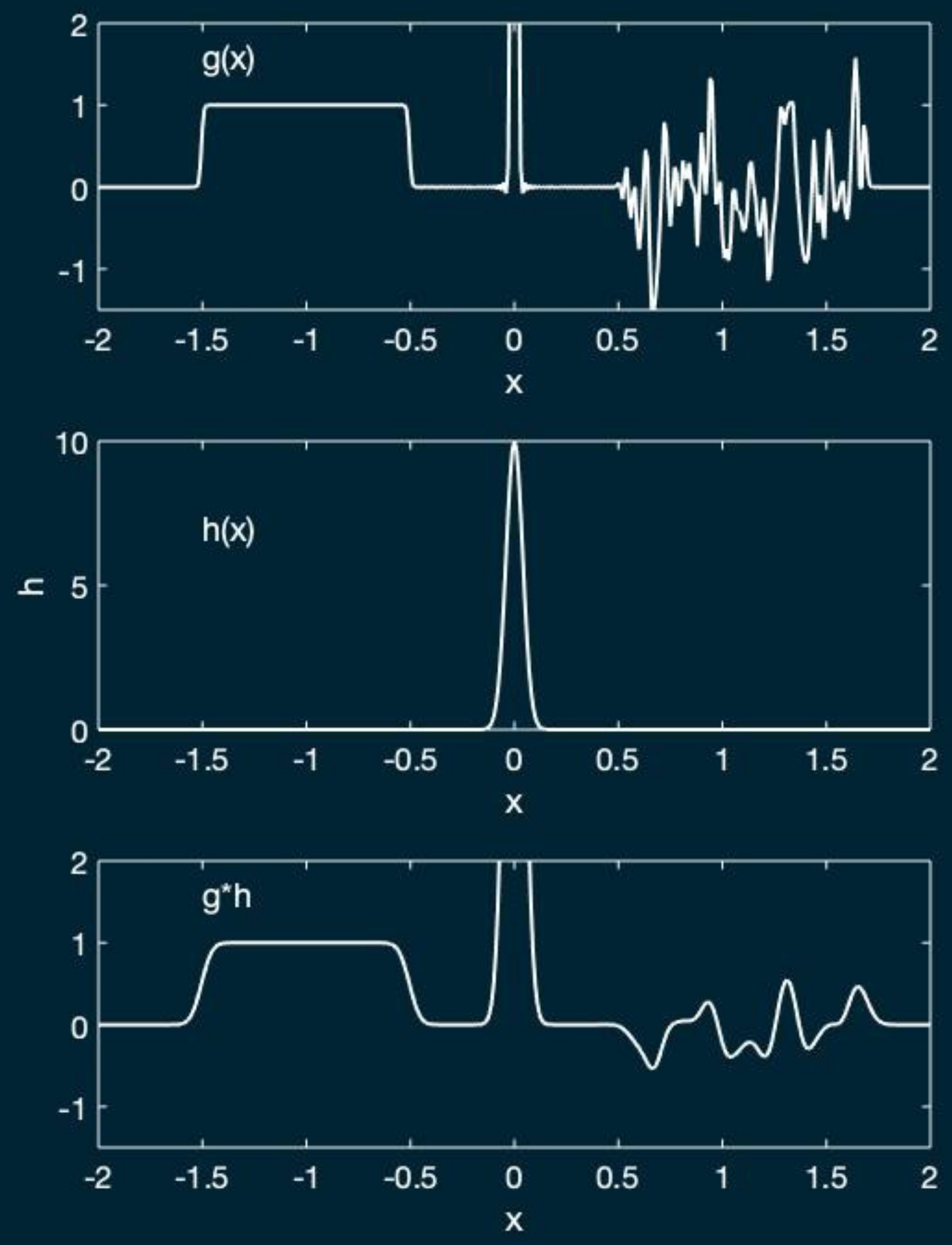
$$g(x) + h(x) \to G(x) + H(x)$$ **Linearity**

$$ag(ax) \to G(u/a)$$ **Scale**

$$g(x-b) \to G(u)e^{-i2\pi ub}$$ **Shift**

$$g \star h \to G(u)H(u)$$ **Convolution**

Convolution with a Gaussian kernel



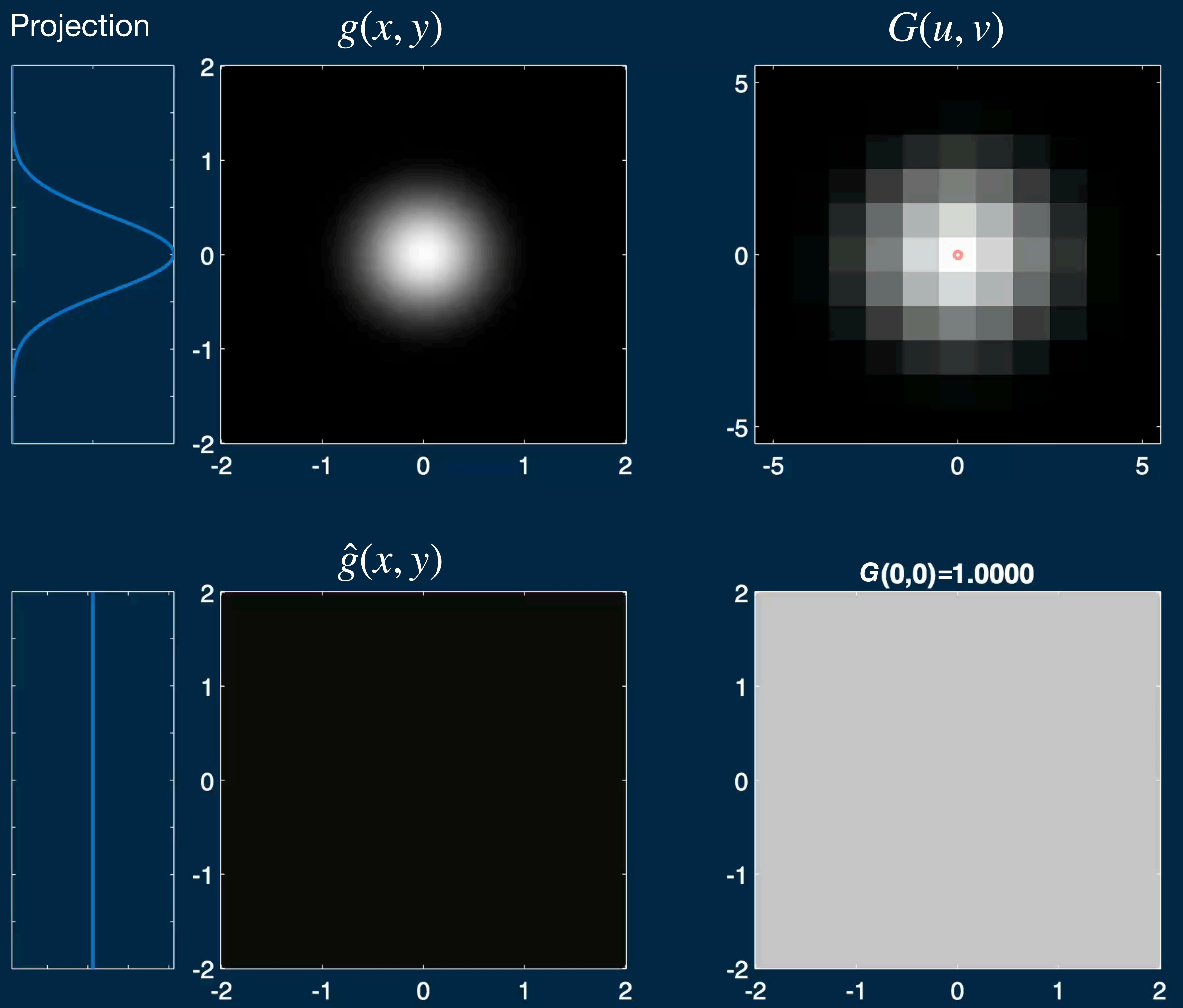
Convolution

$f(x) = g \star h$ means:

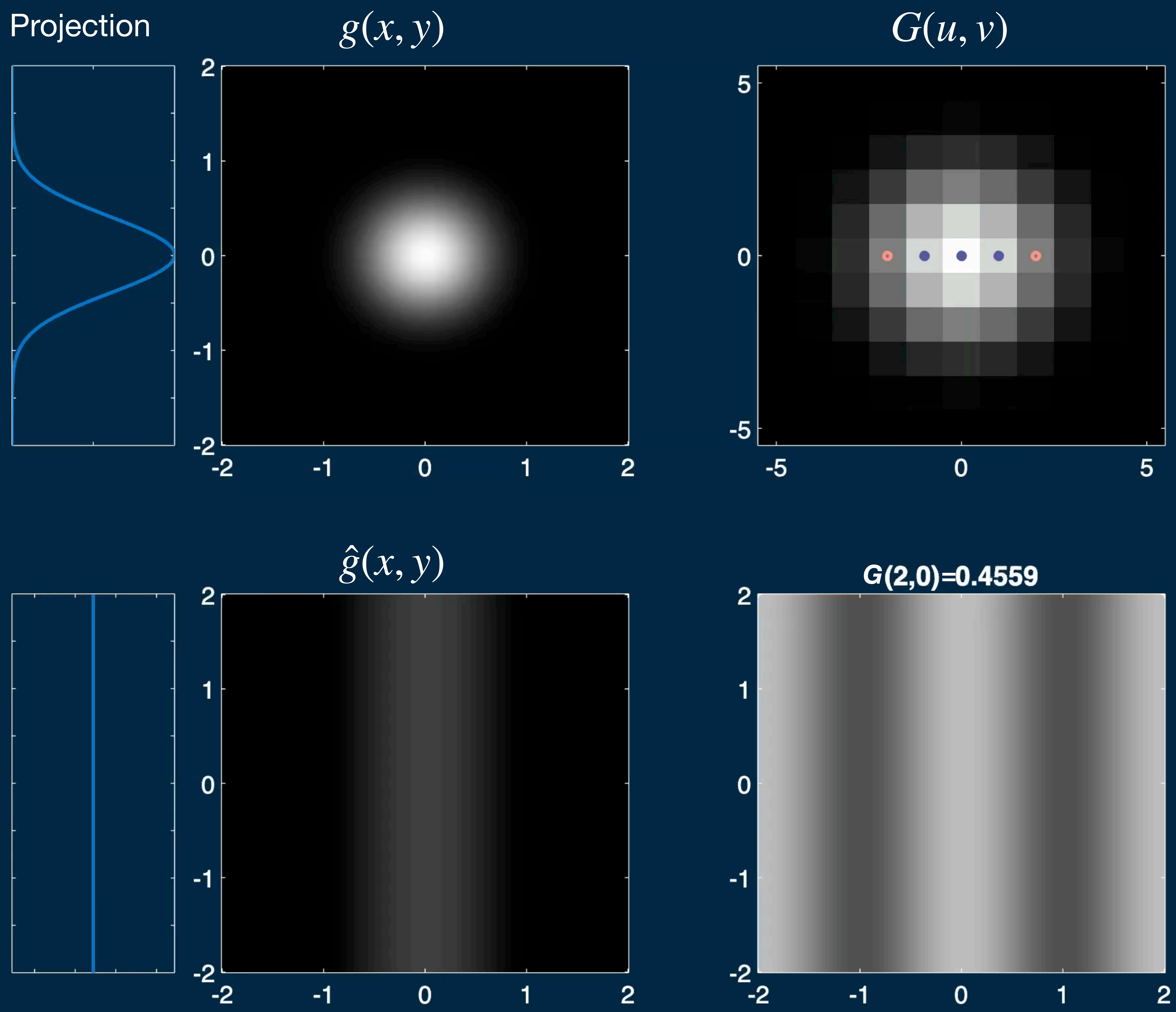
$$f(x) = \int g(x-s)h(s)ds$$

The Fourier transform in two dimensions

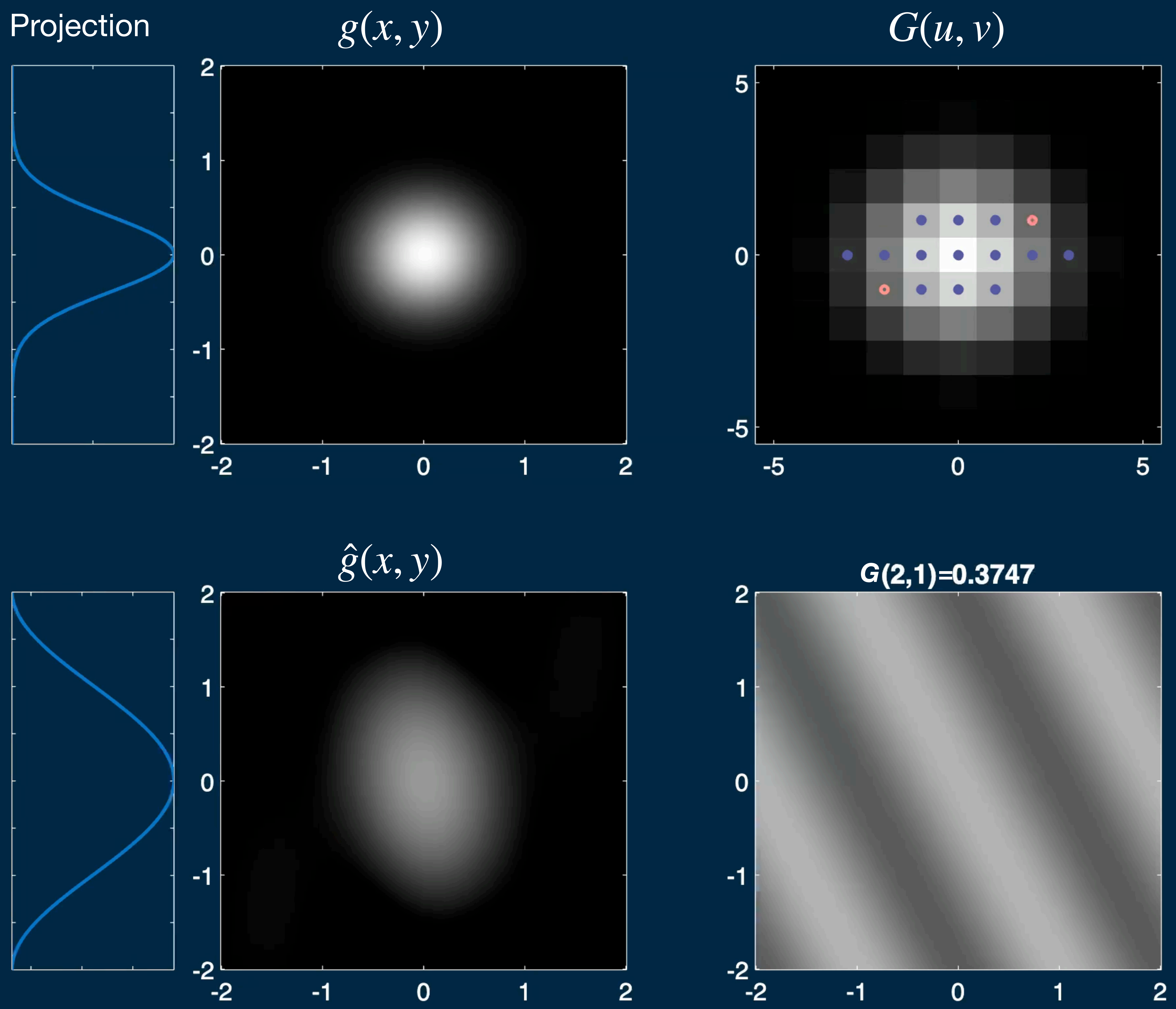
Fourier reconstruction of a 2D Gaussian function



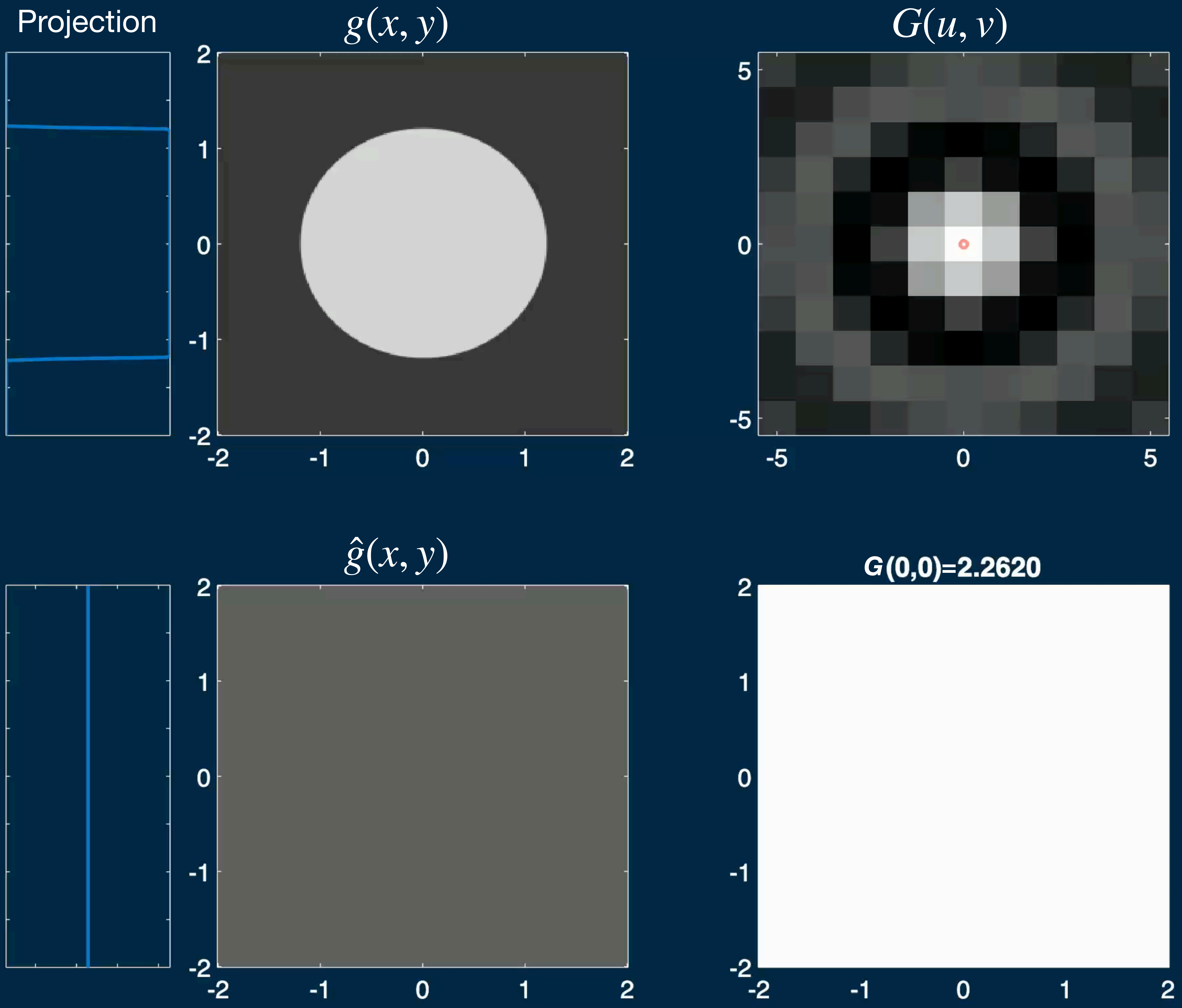
Fourier reconstruction of a 2D Gaussian function



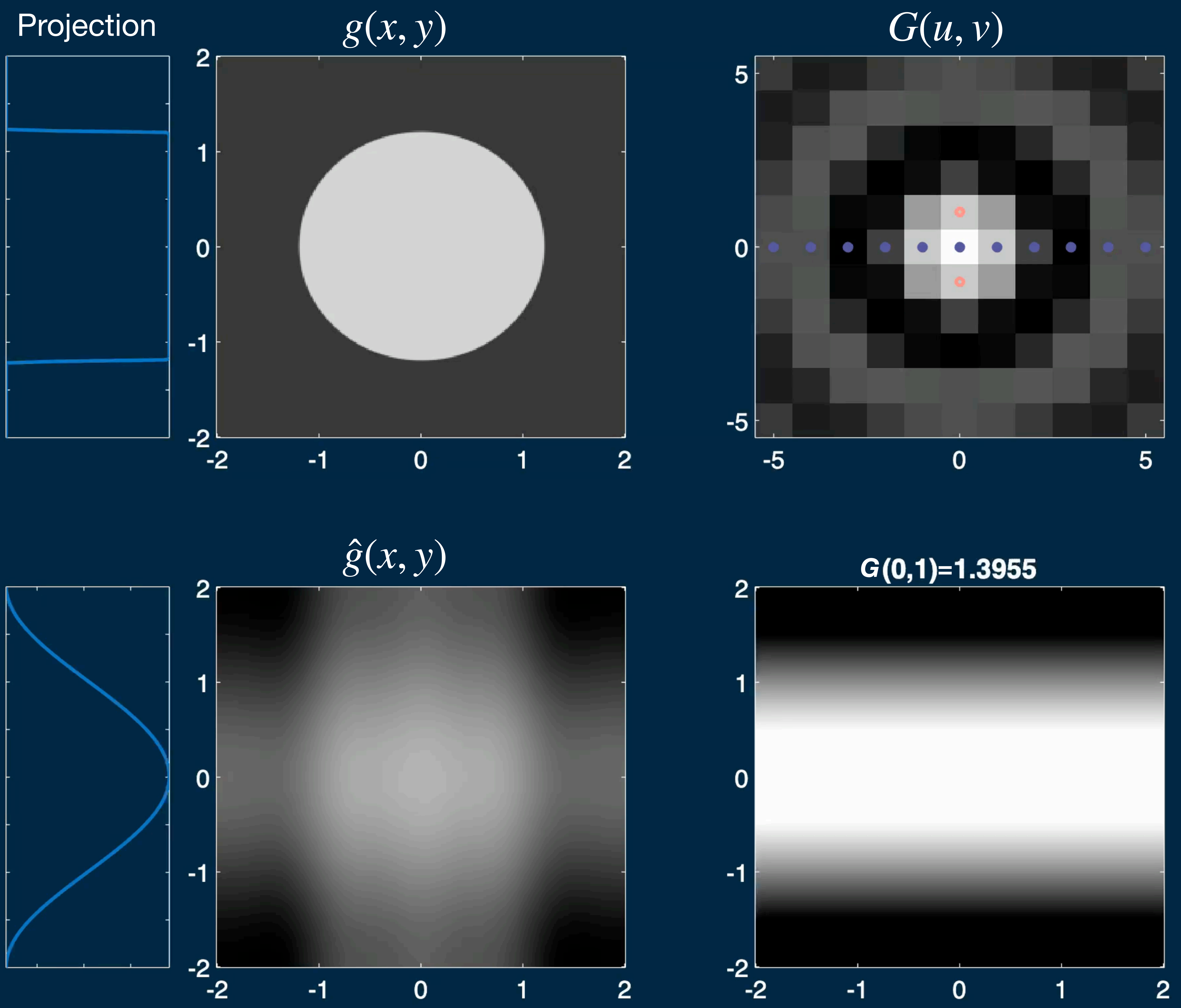
Fourier reconstruction of a 2D Gaussian function



Fourier reconstruction of a 2D circ function



Fourier reconstruction of a 2D circ function



2D transform pairs

$$e^{-\pi(x^2+y^2)} \to e^{-\pi(u^2+v^2)}$$

$$\mathrm{circ}(\mathrm{x},\mathrm{y}) \to \frac{\mathrm{J}_1(2\pi\rho)}{\pi\rho}, \quad \rho = \sqrt{\mathrm{u}^2+\mathrm{v}^2}$$

$$\delta(x)\delta(y) \to 1$$

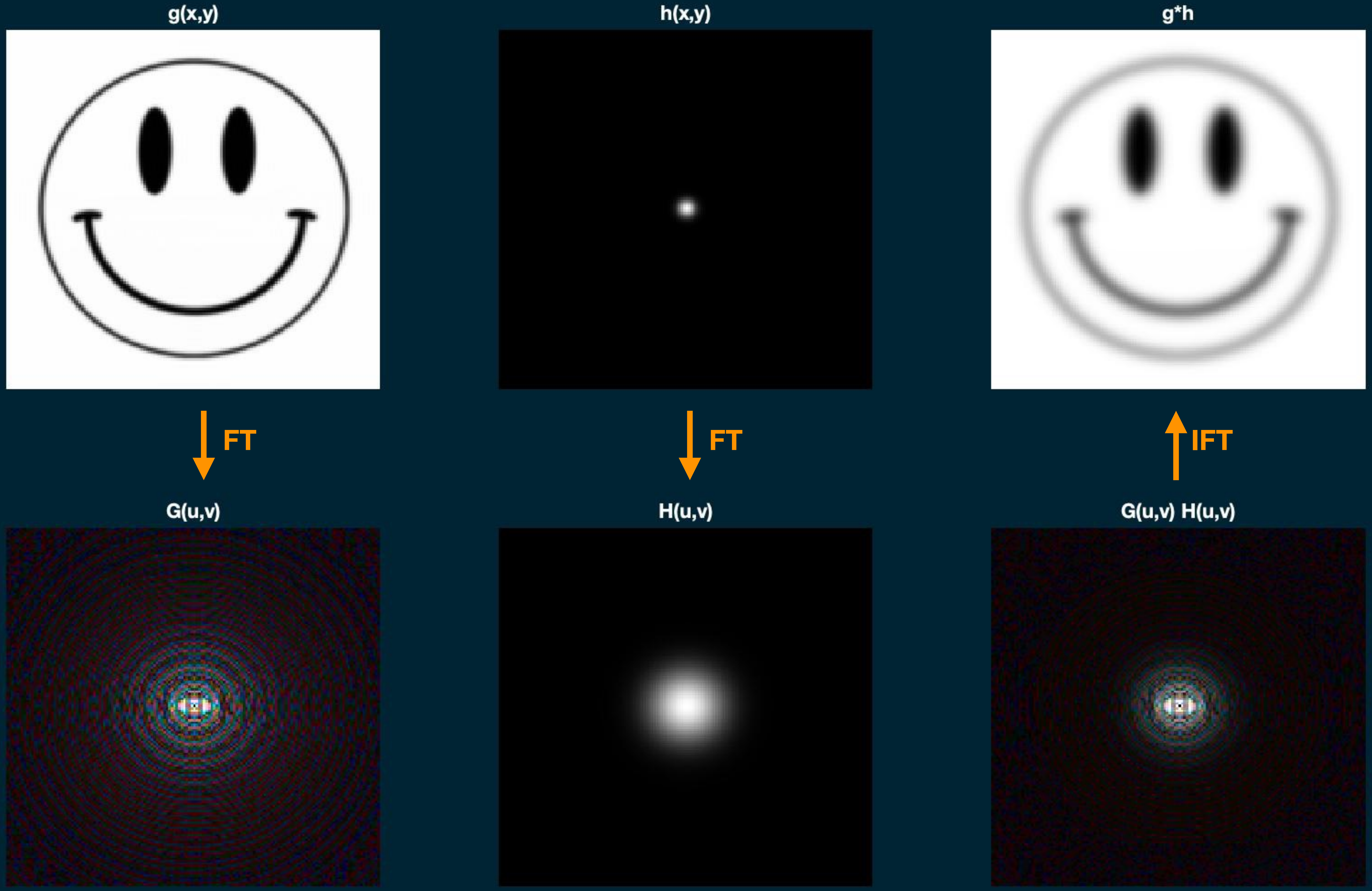
2D Fourier transform properties

$ab\, g(ax, by) \rightarrow G(u/a, v/b)$	Scale
$g(x - a, y - b) \rightarrow G(u, v)e^{-i2\pi(au+bv)}$	Shift
$g * h \rightarrow GH$	Convolution
$g(x', y') \rightarrow G(u', v')$	Rotation
$P_y\, g(x, y) \rightarrow G(u,0)$	Projection

Convolution in 2D

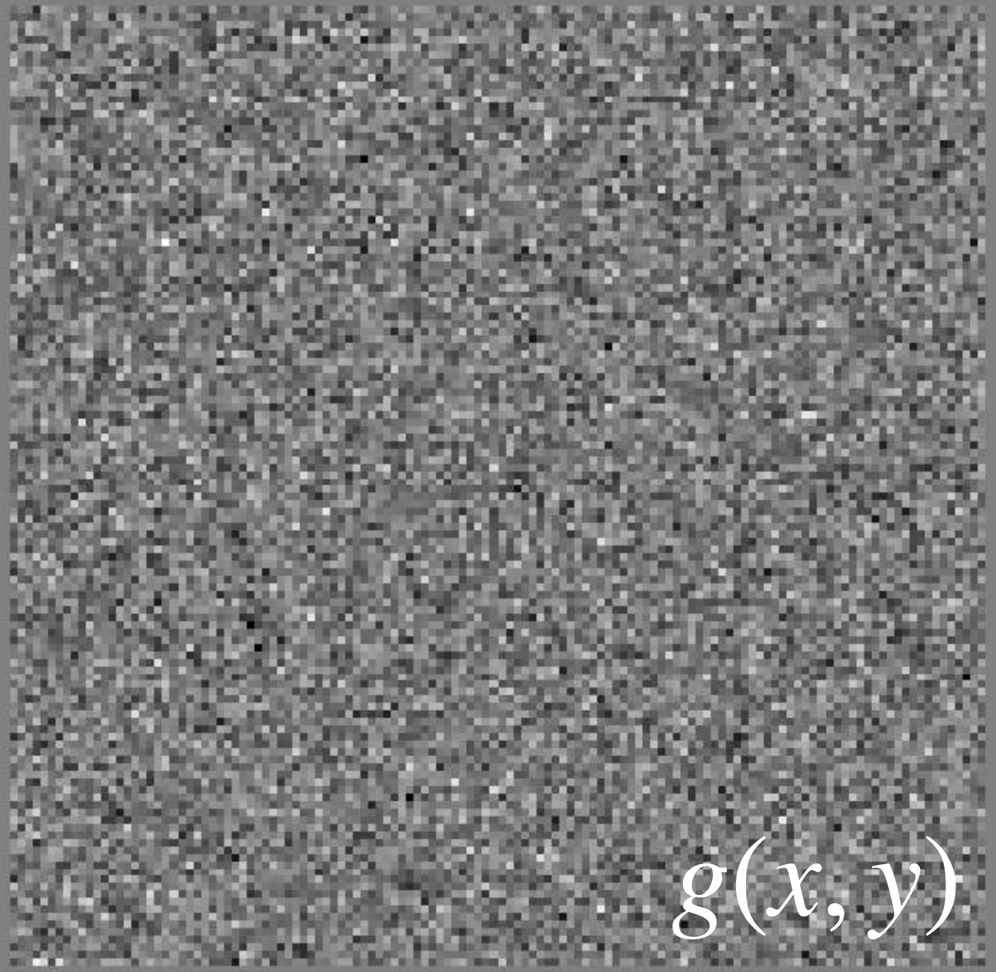
$$g * h = \iint g(x - s, y - t)\, h(s, t)\, ds\, dt$$

Convolution with a Gaussian

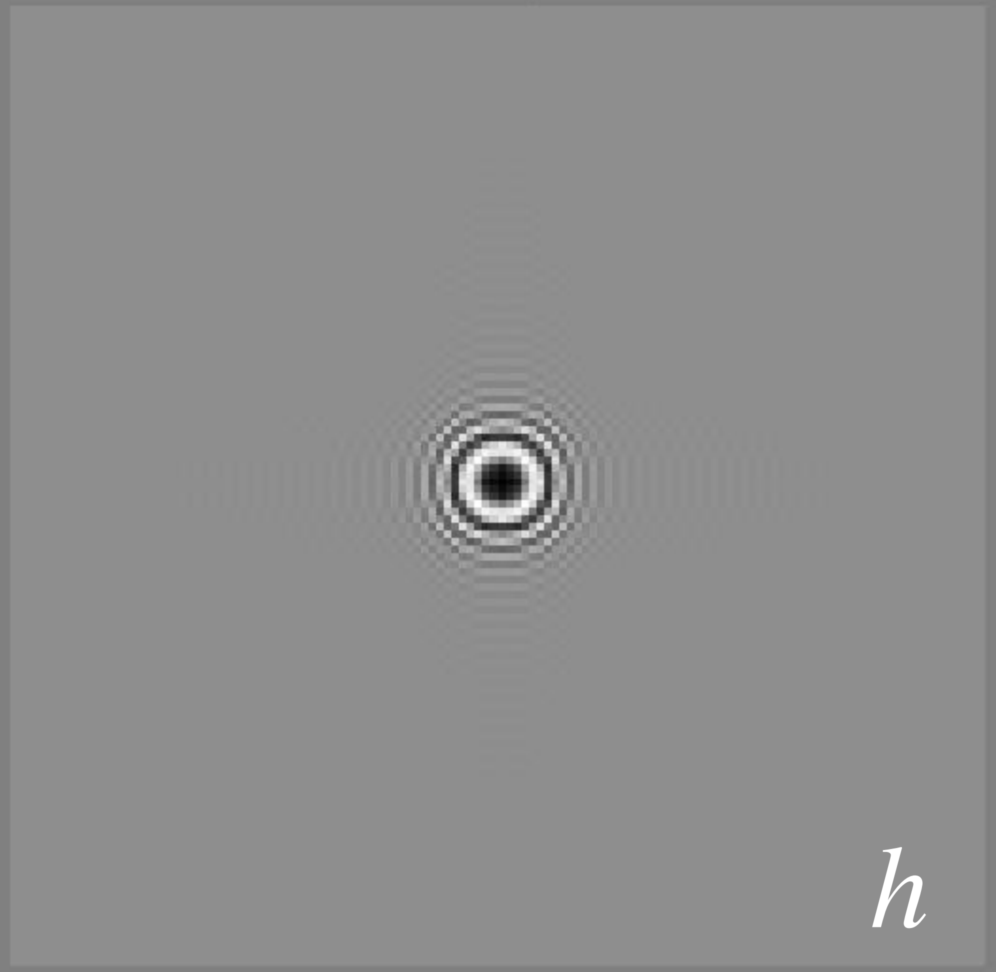


Visualizing the contrast transfer function

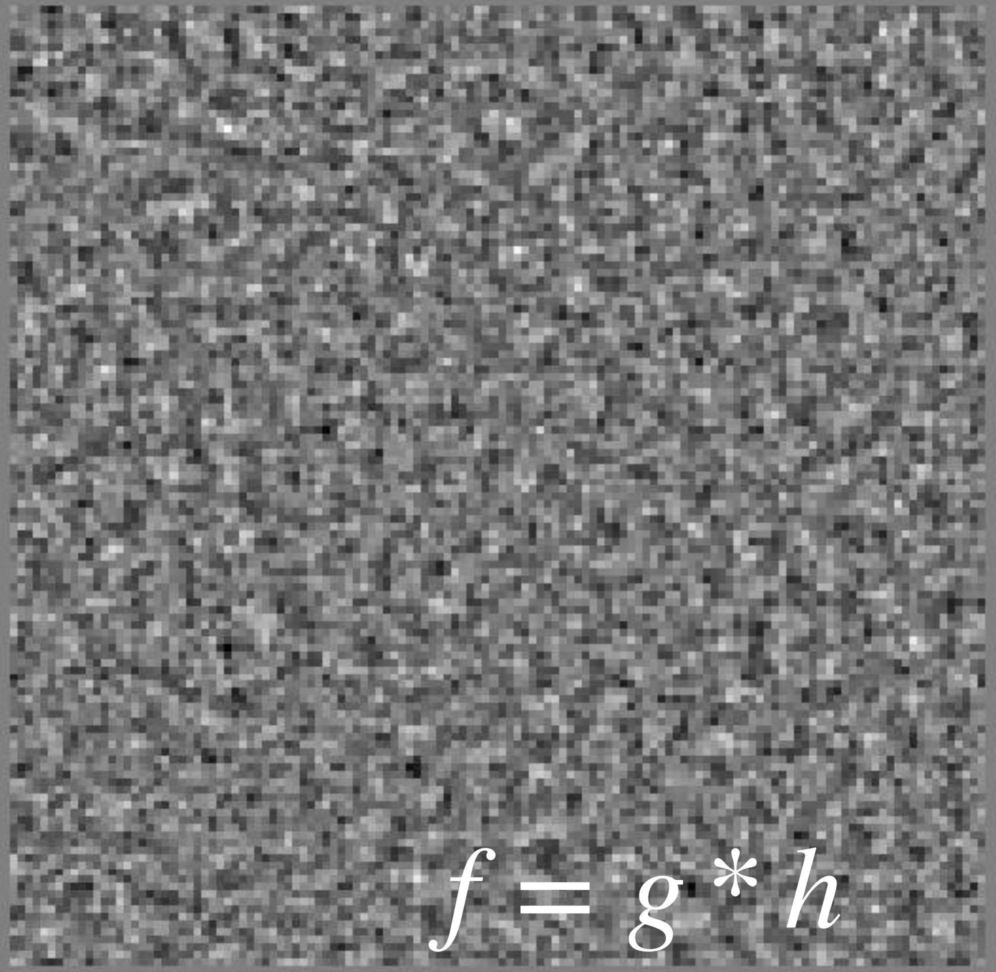
Random object



Point-spread

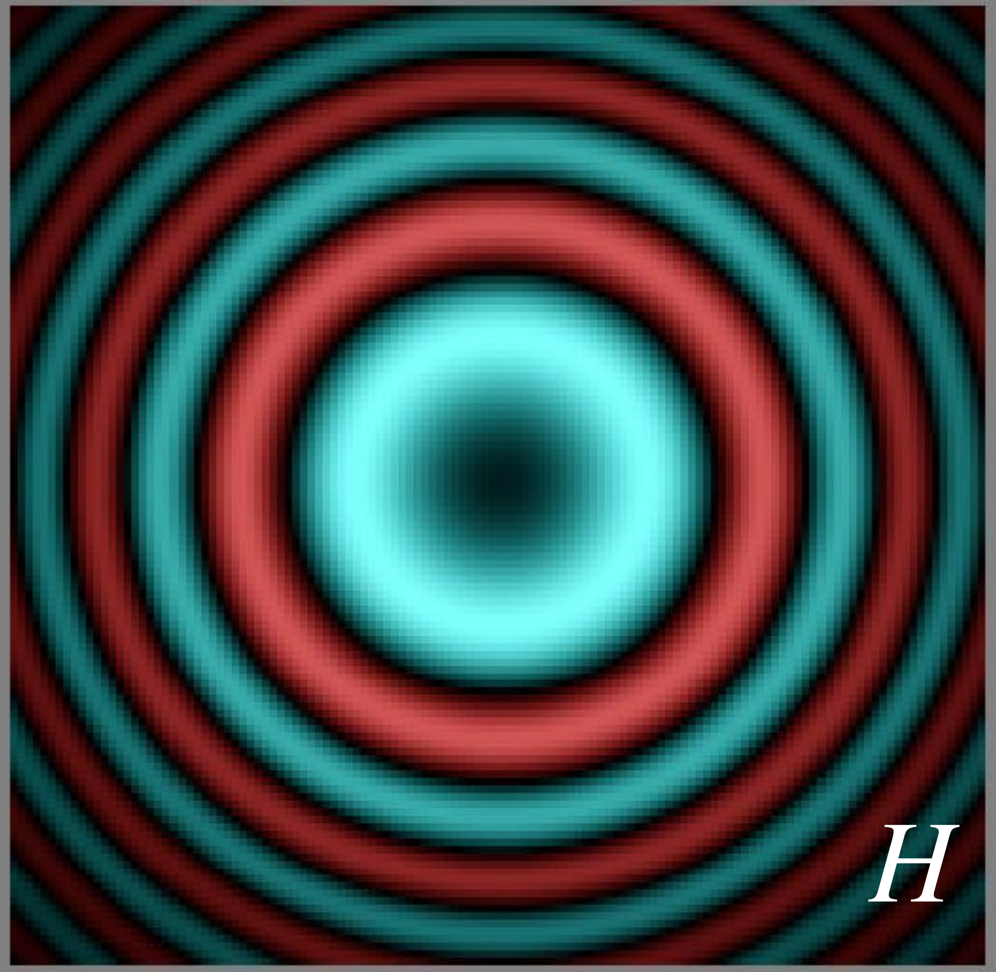


Image

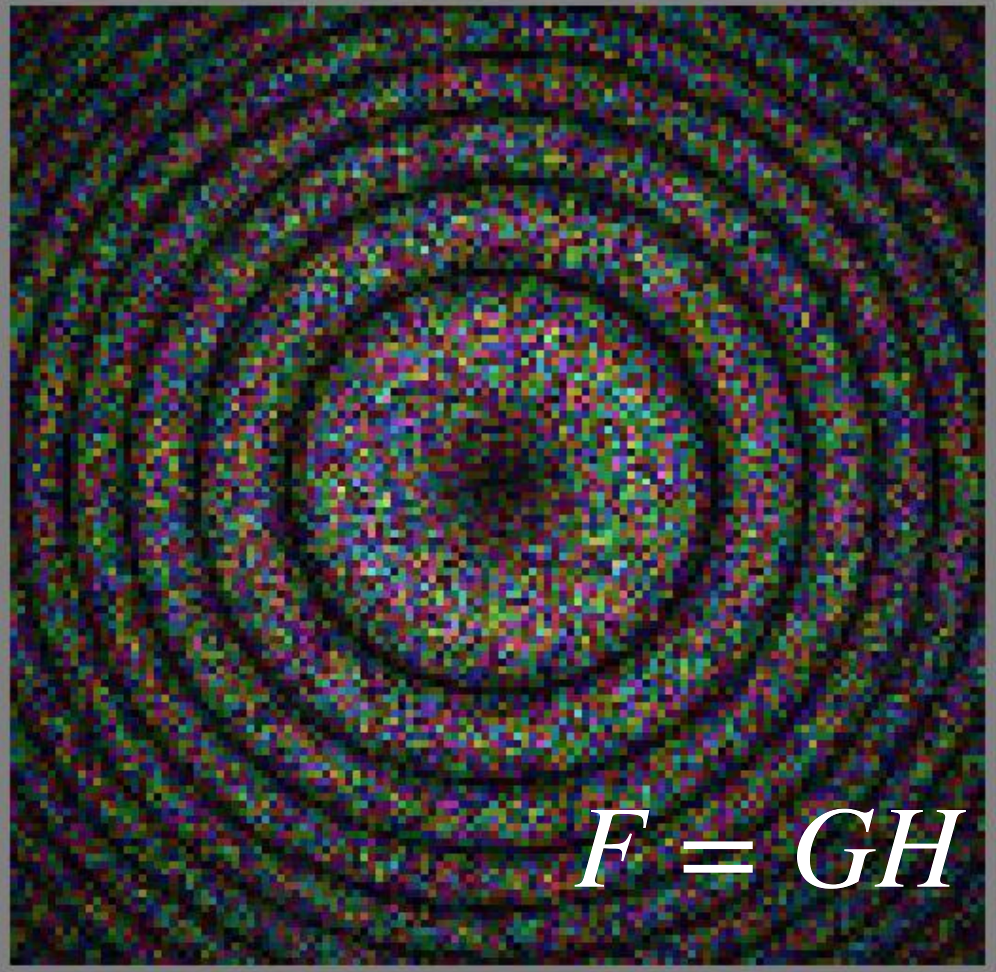


FT of object

CTF

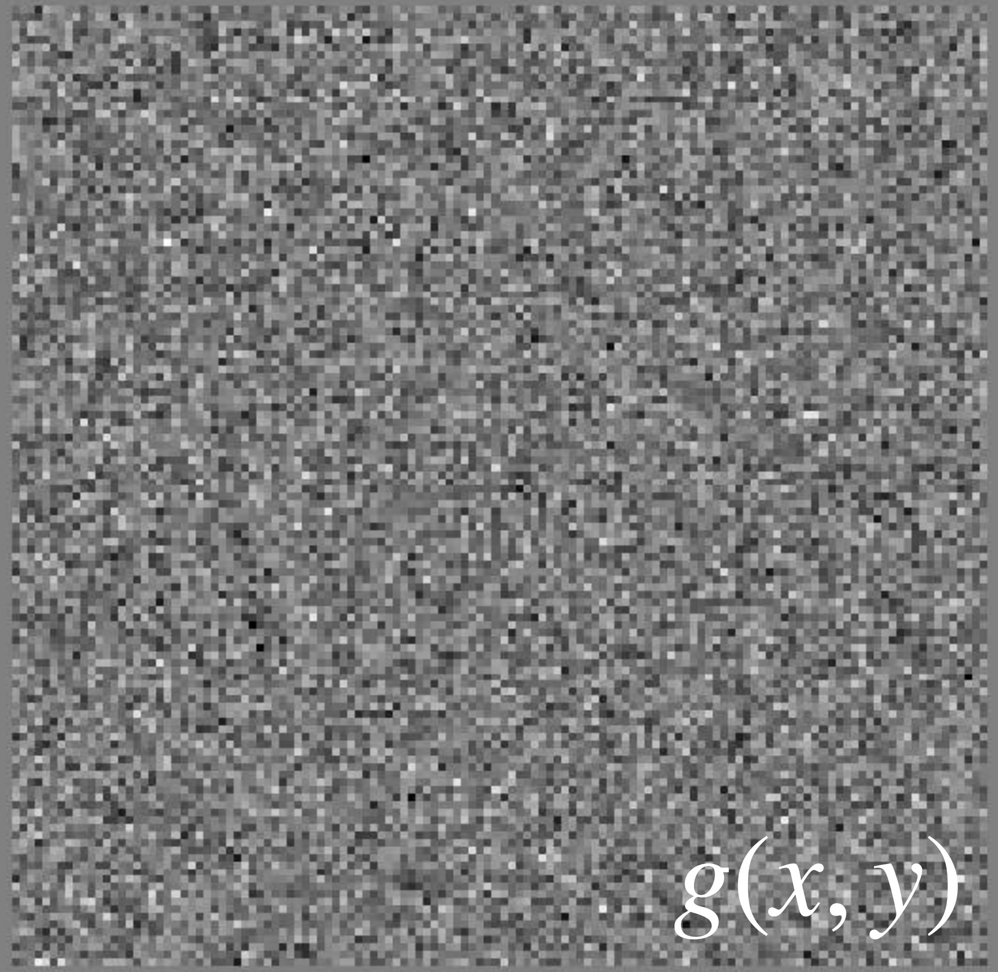


FT of image

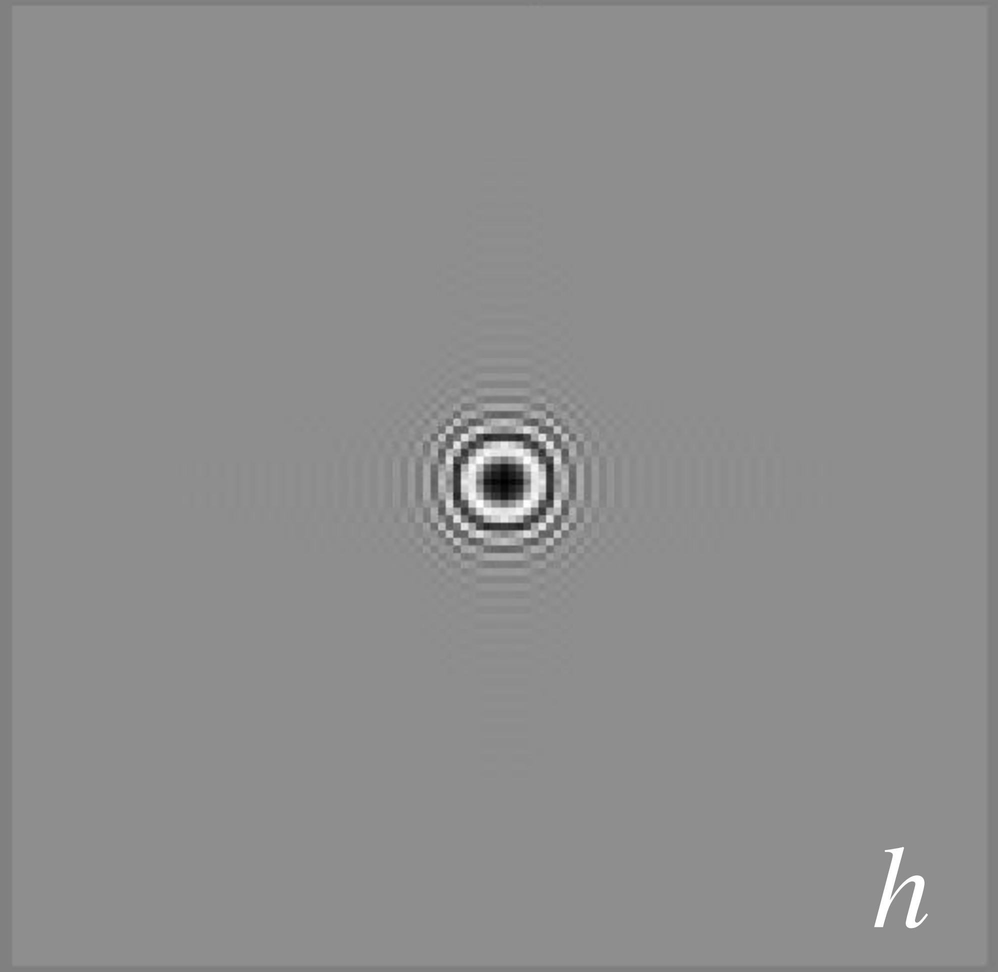


Visualizing the contrast transfer function

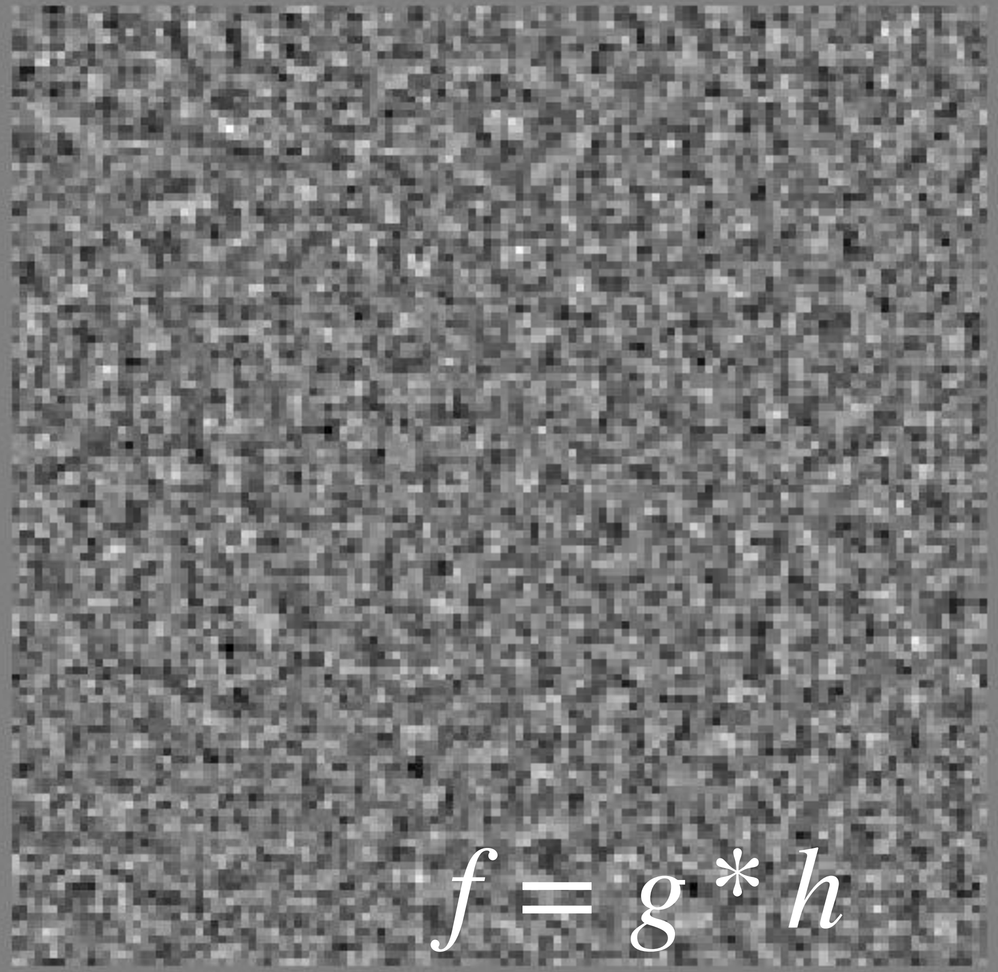
Random object



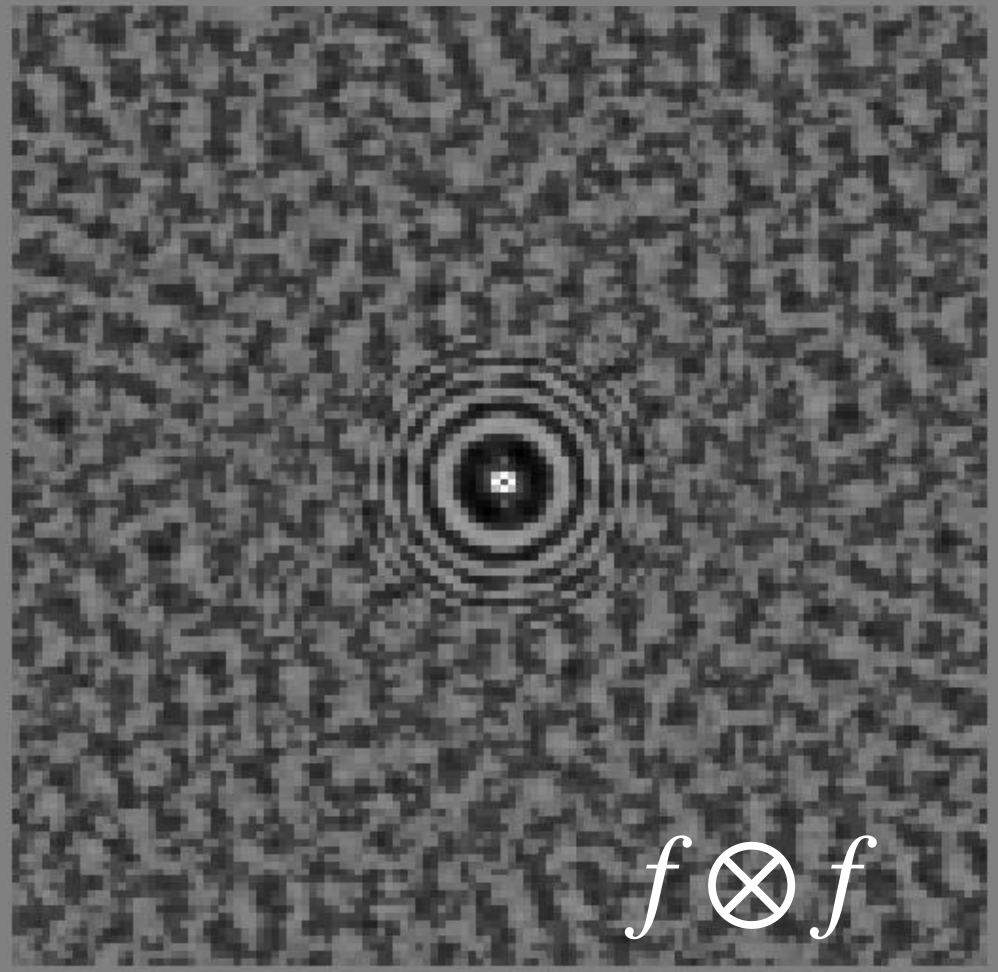
Point-spread



Image

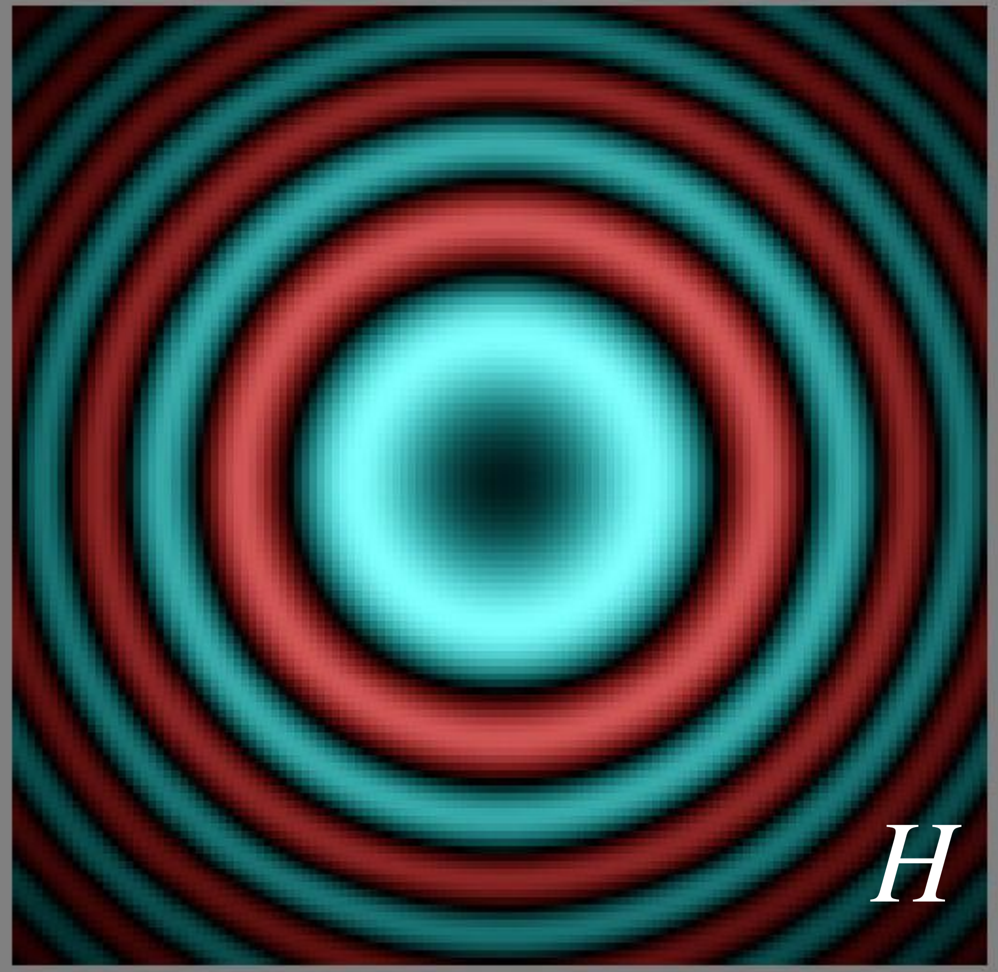


Autocorrelation

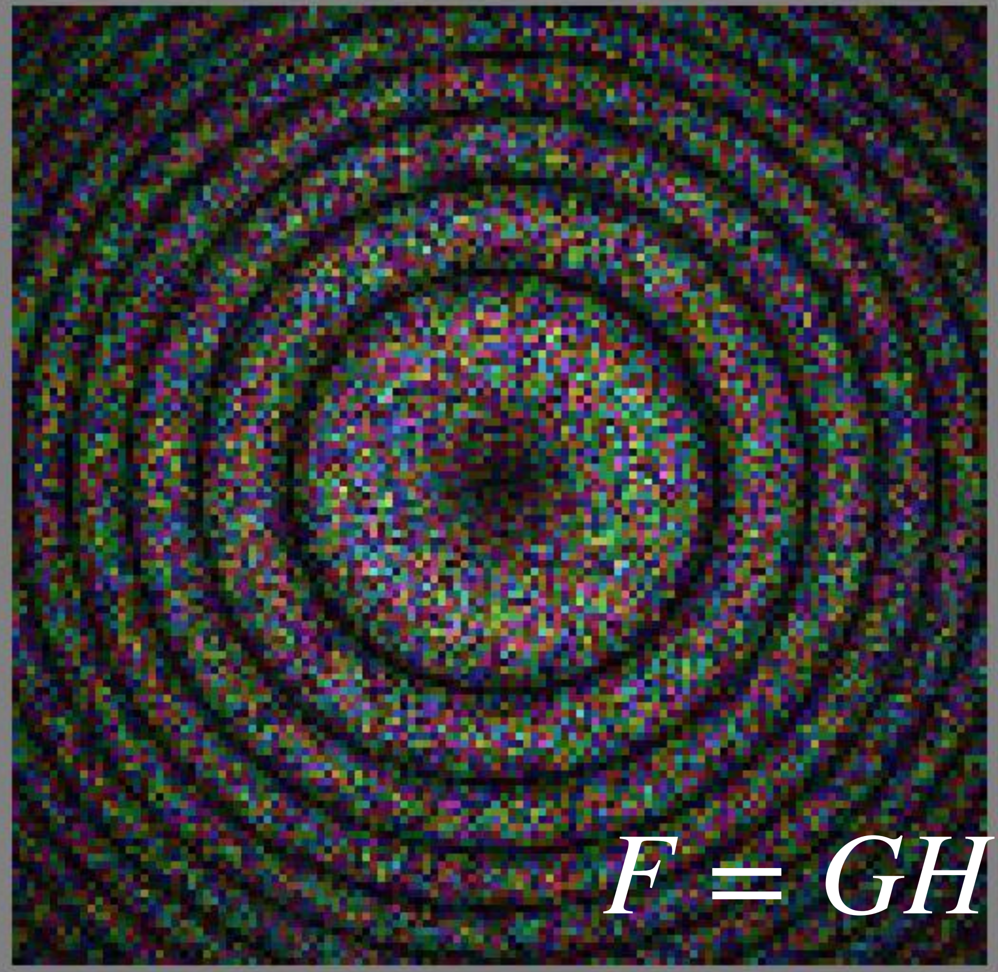


FT of object

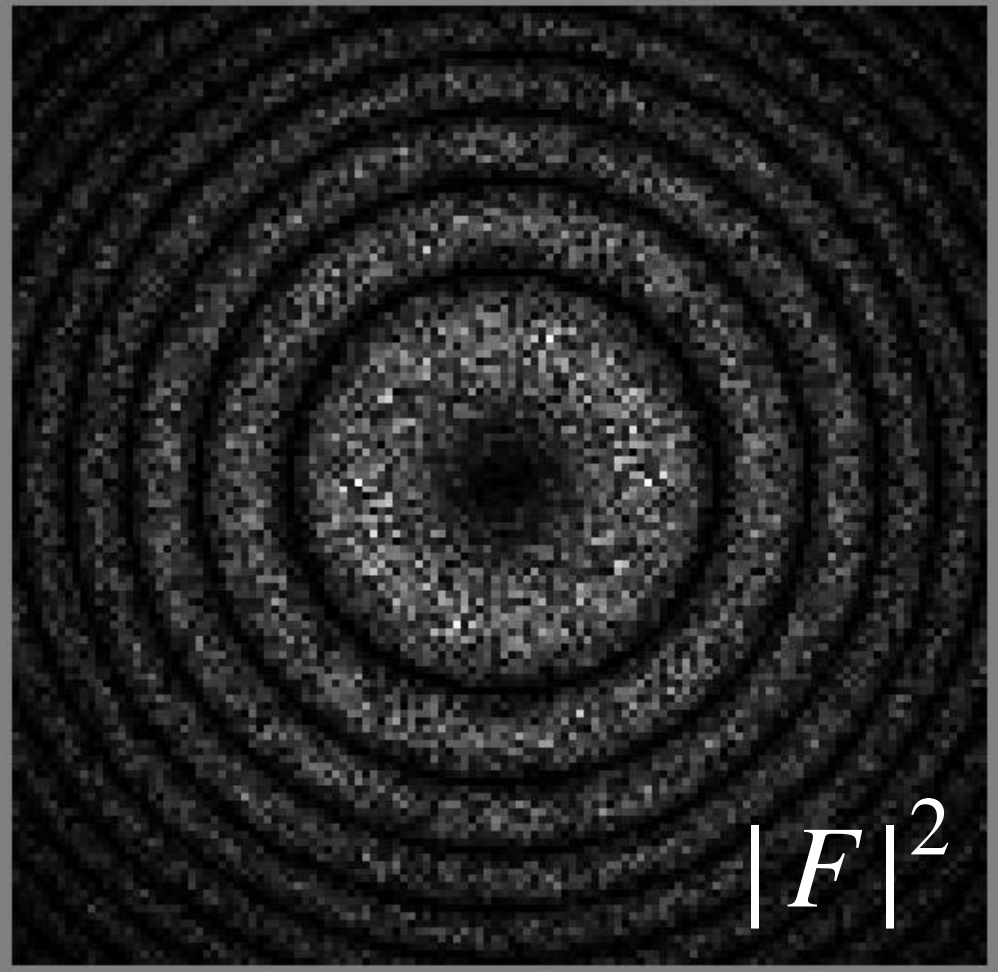
CTF



FT of image



Power spectrum



The rotation property

2D Fourier Transform

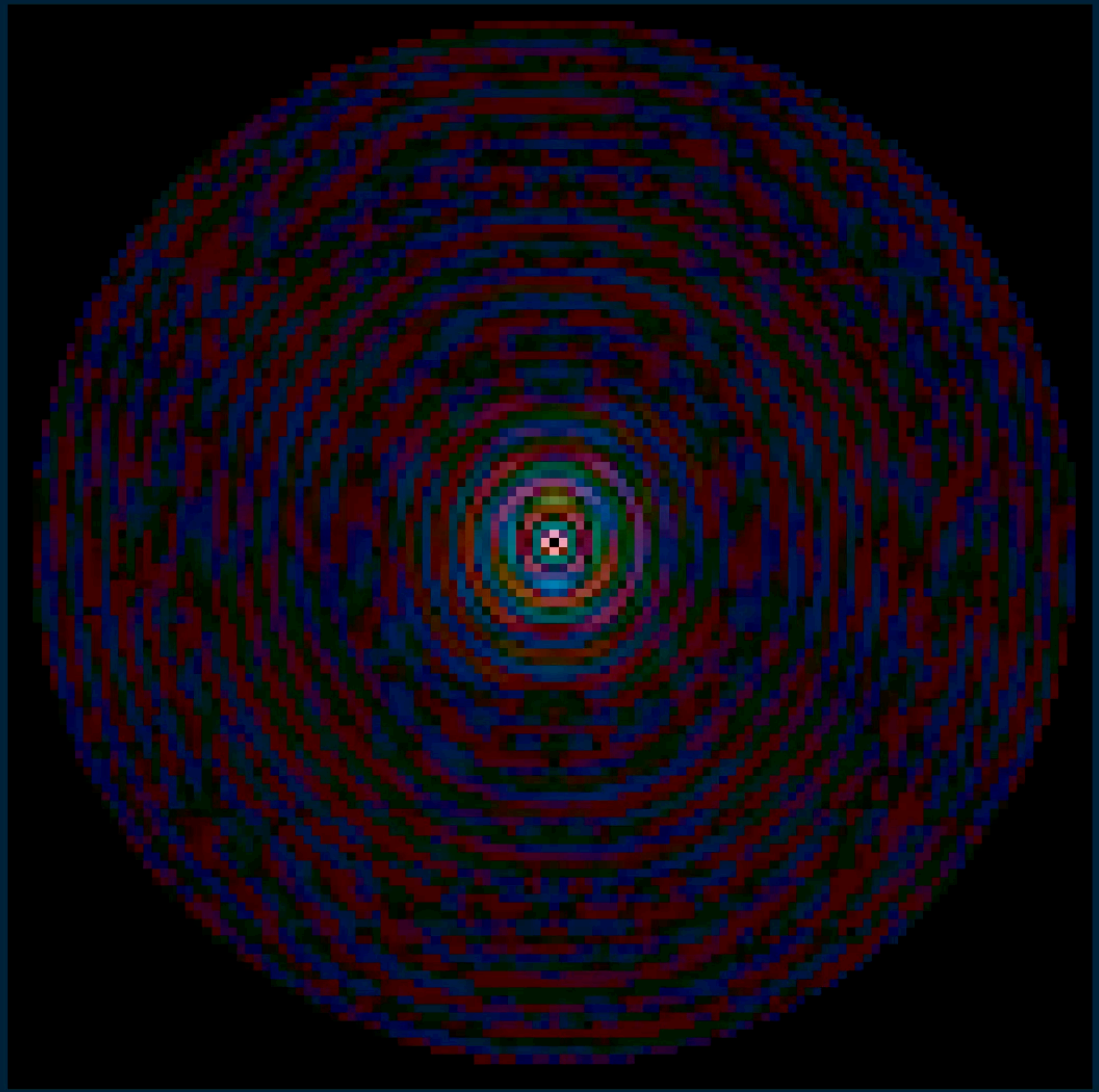
$$\boxed{G(u,v) = \iint g(x,y) e^{-i2\pi(ux+vy)} dxdy}$$

FT using 2D vectors

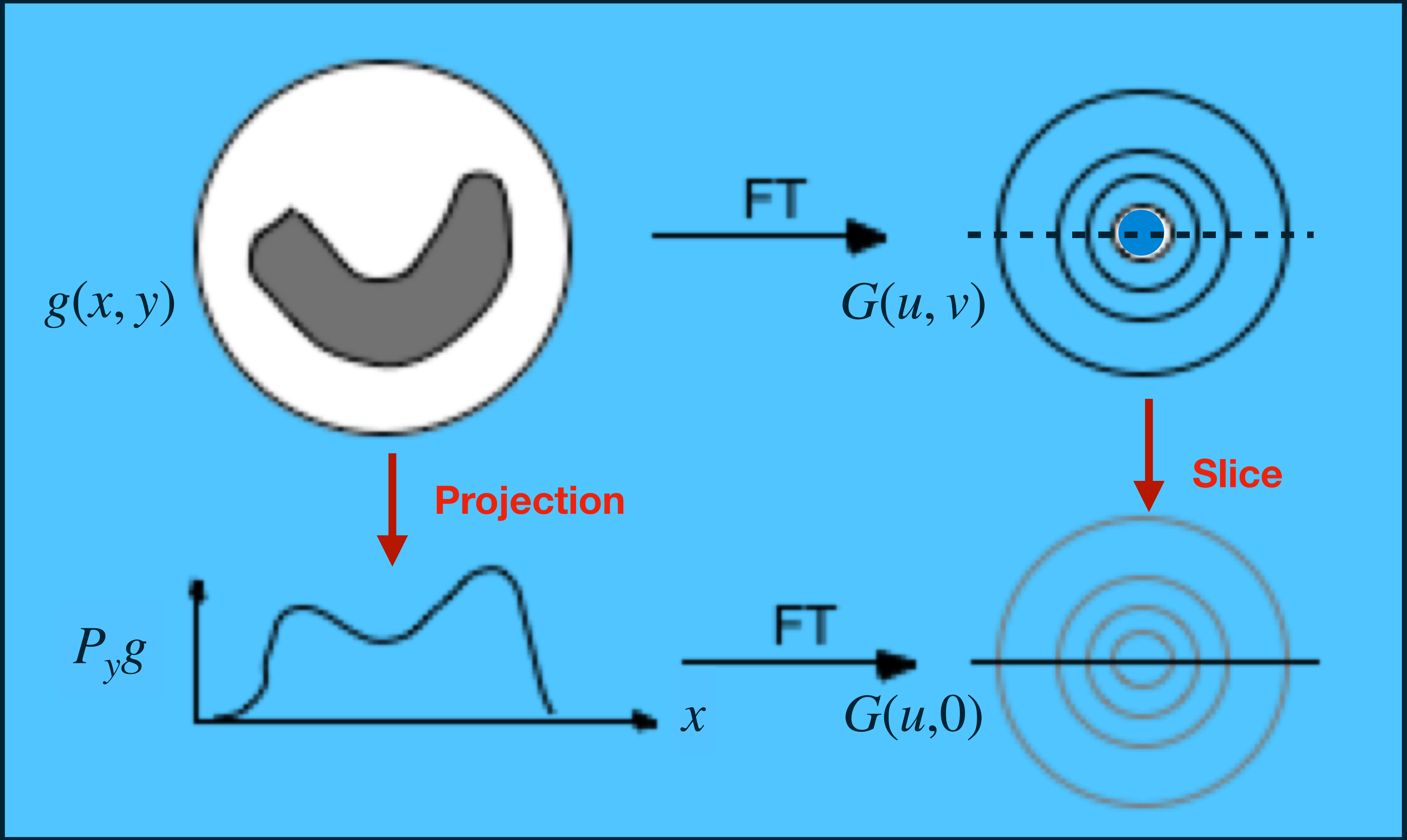
$$\boxed{G(\mathbf{u}) = \int\int g(\mathbf{x}) e^{-i2\pi(\mathbf{u}\cdot\mathbf{x})} d^2\mathbf{x}}$$

The dot-product is invariant under rotations!

FT



The Fourier Slice Theorem in 2D

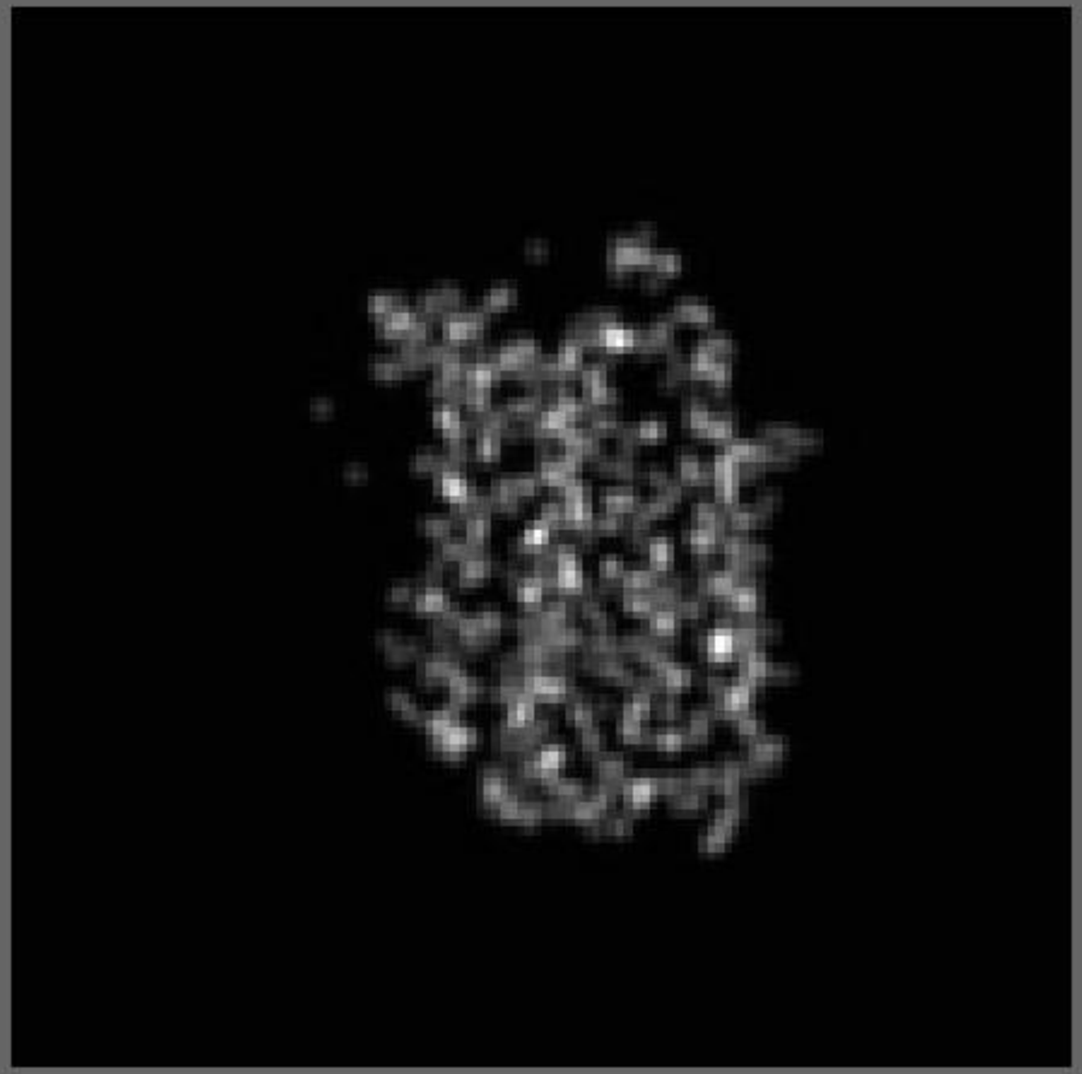


$$G(u, v) = \iint g(x, y) e^{-i2\pi(ux+vy)} dxdy$$

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

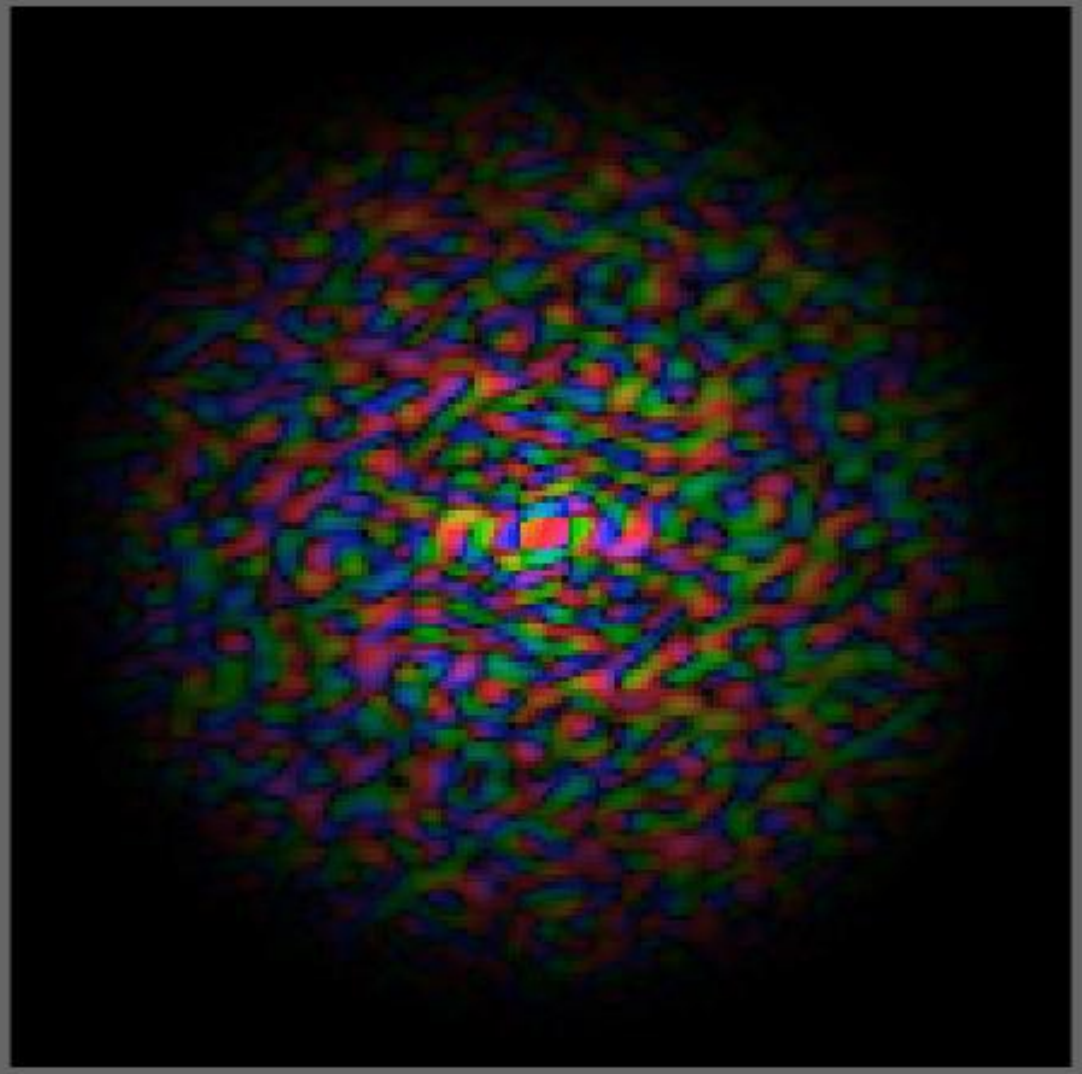
$$P_y g(x, y) = \int g(x, y) dy$$

Tomographic reconstruction using the Fourier slice theorem

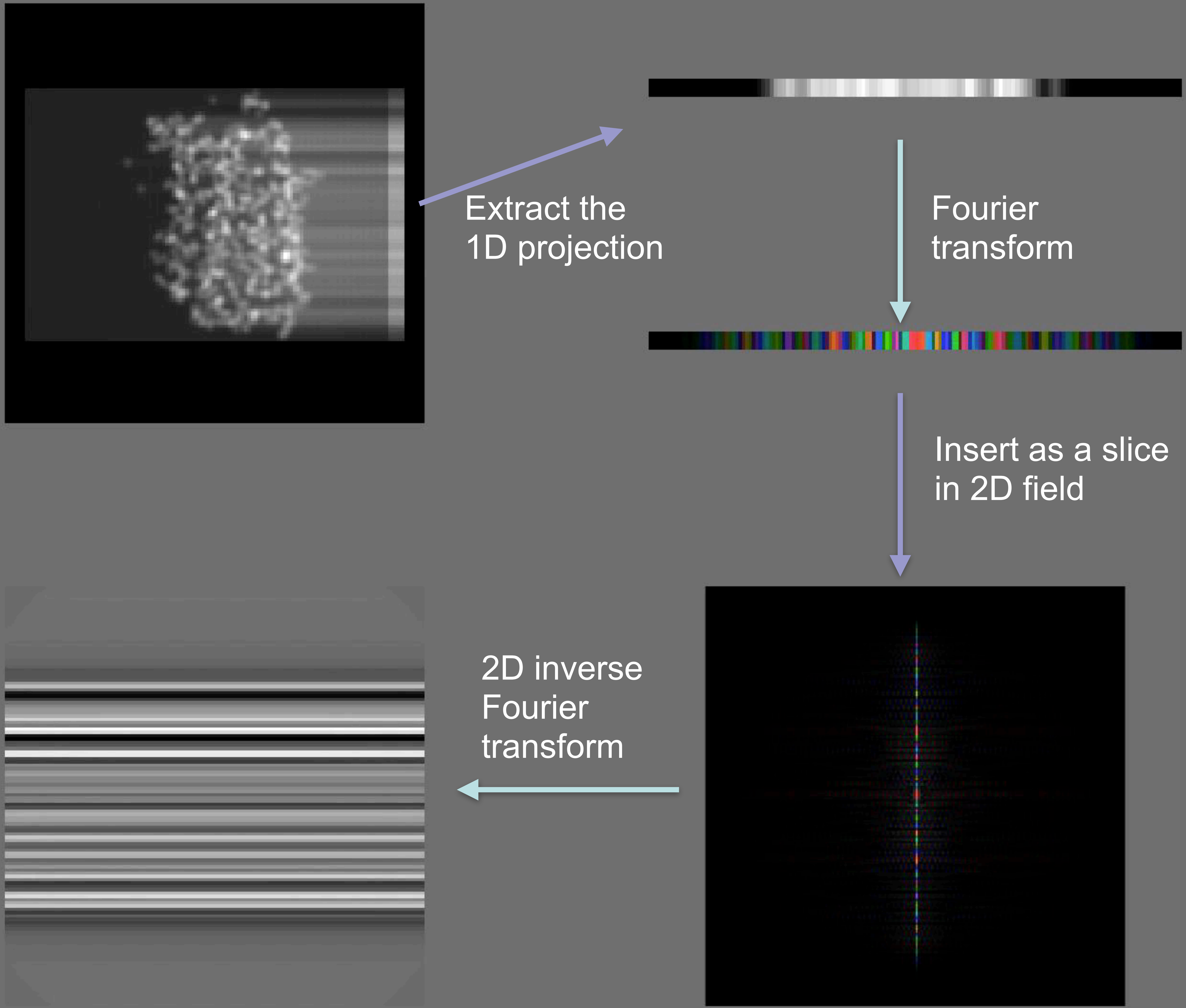


Here we'll demonstrate building up a 2D structure from 1D projections.

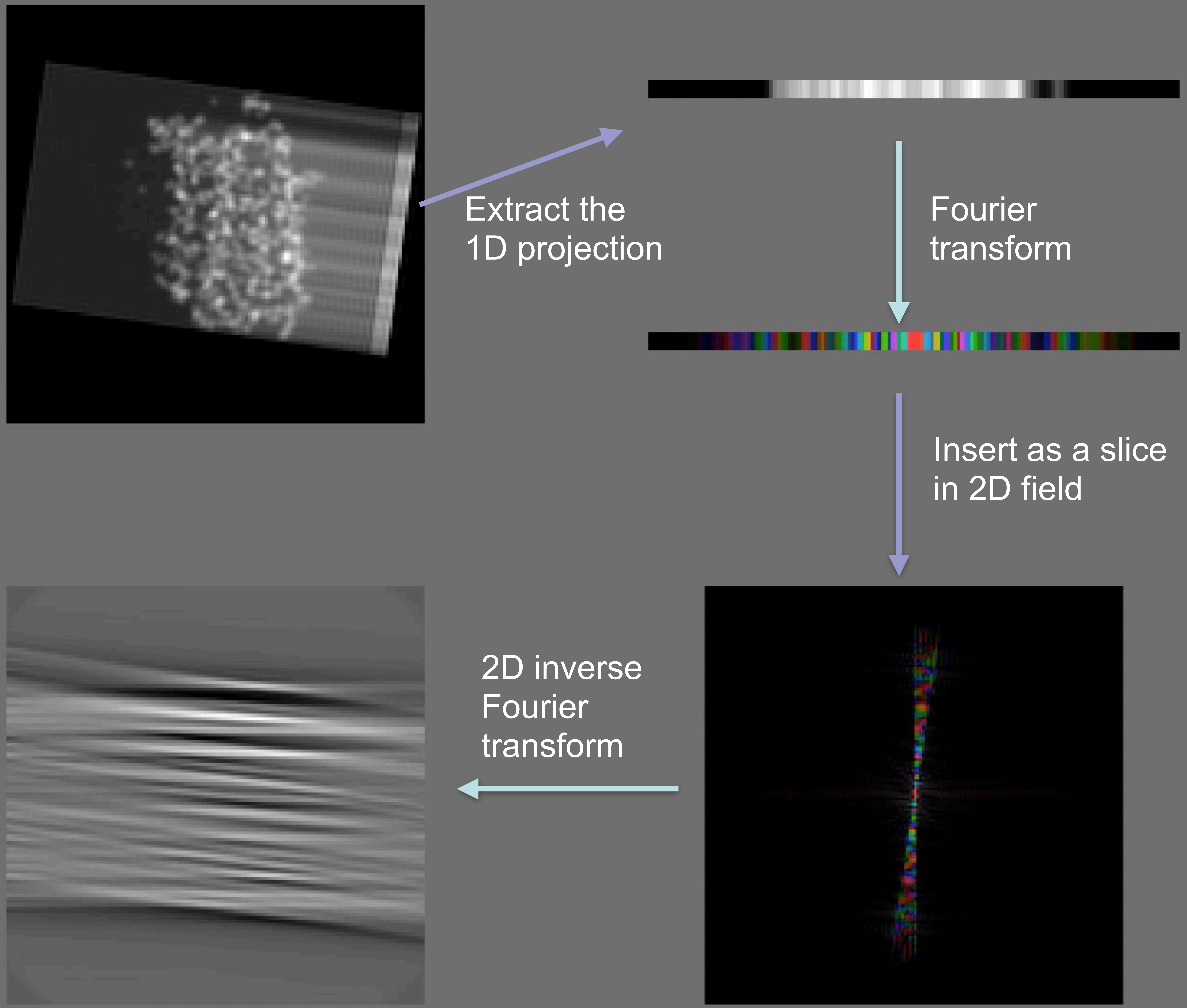
Fourier transform



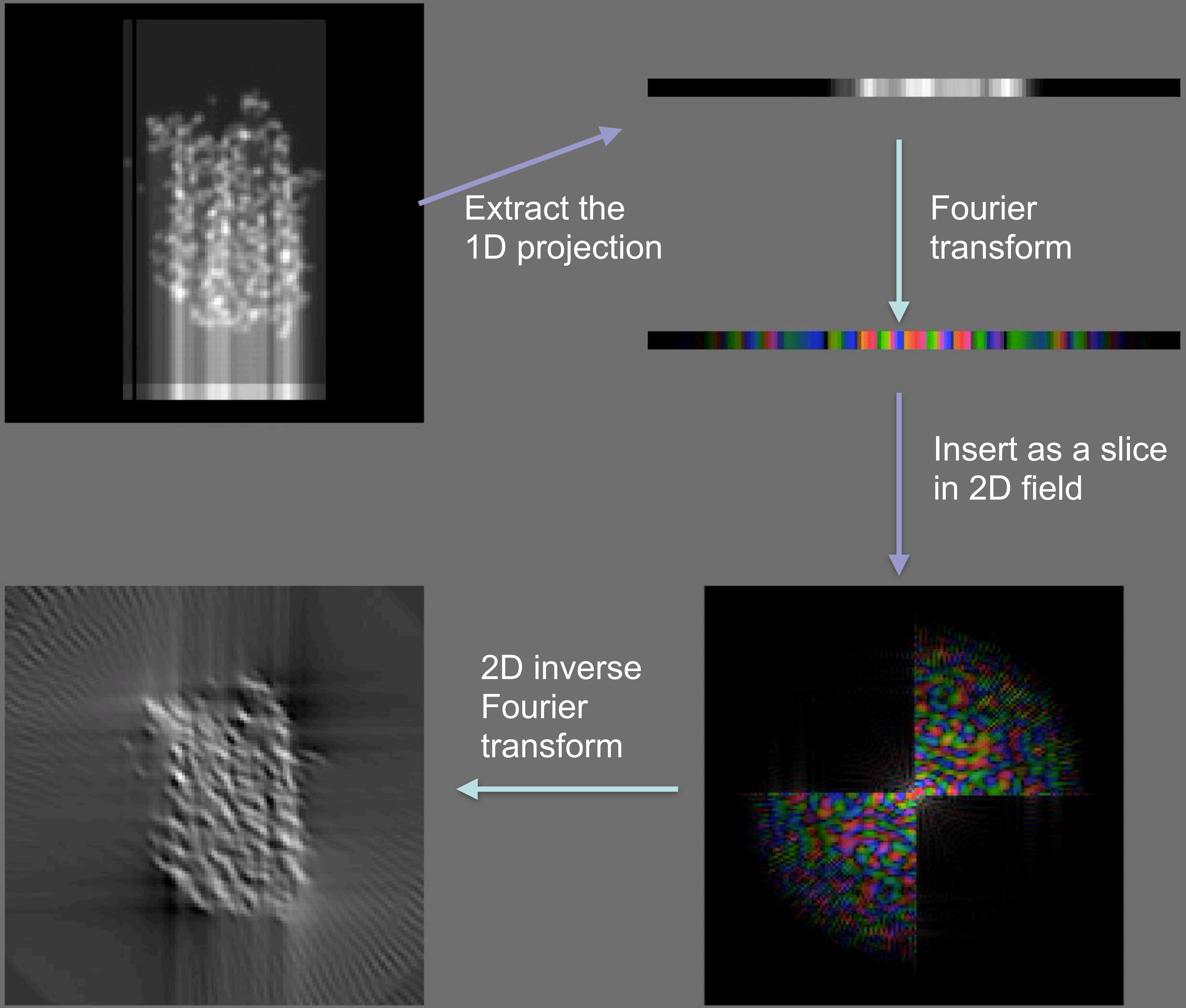
Tomographic reconstruction: 2D image from 1D projections



Tomographic reconstruction: 2D image from 1D projections



Tomographic reconstruction: 2D image from 1D projections



The discrete FT is what is calculated on a computer

2D Fourier transform

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

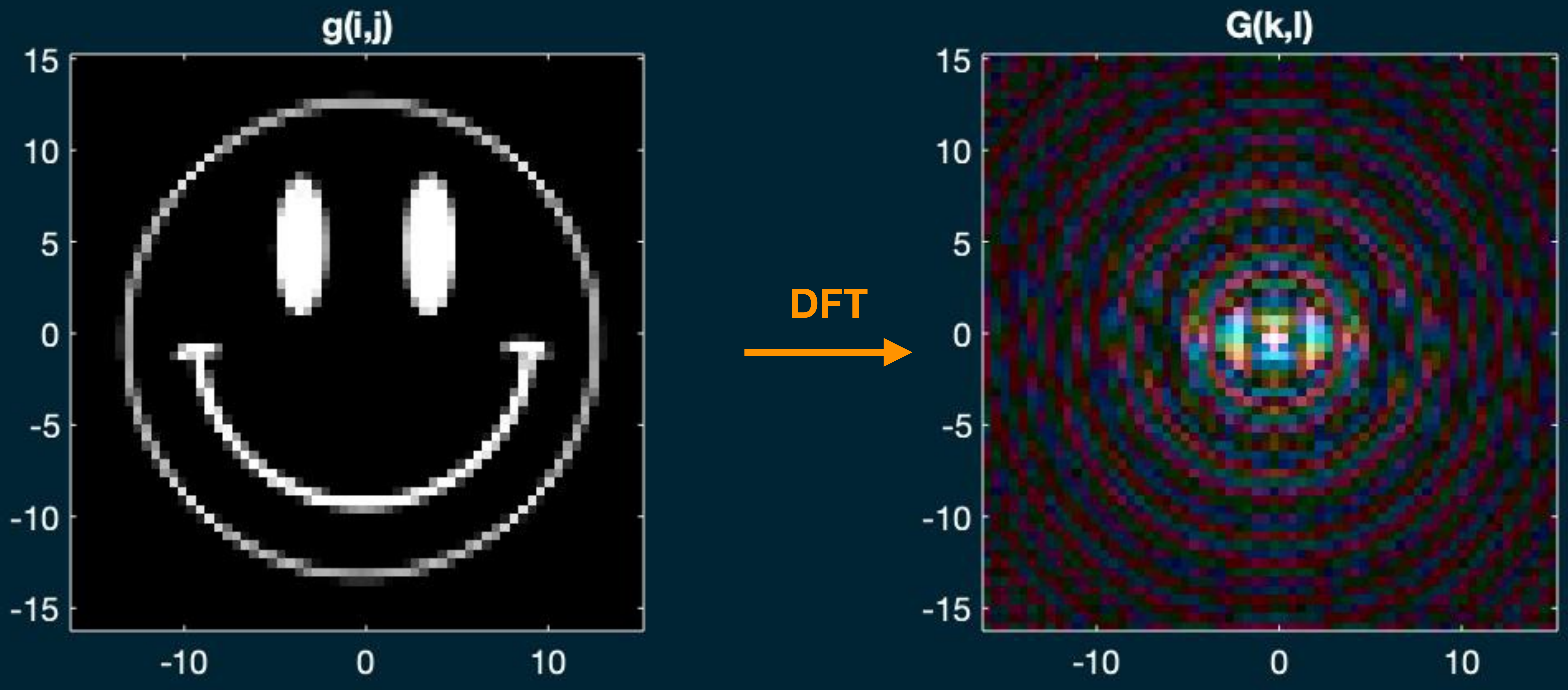
u, v are real numbers

2D discrete Fourier transform

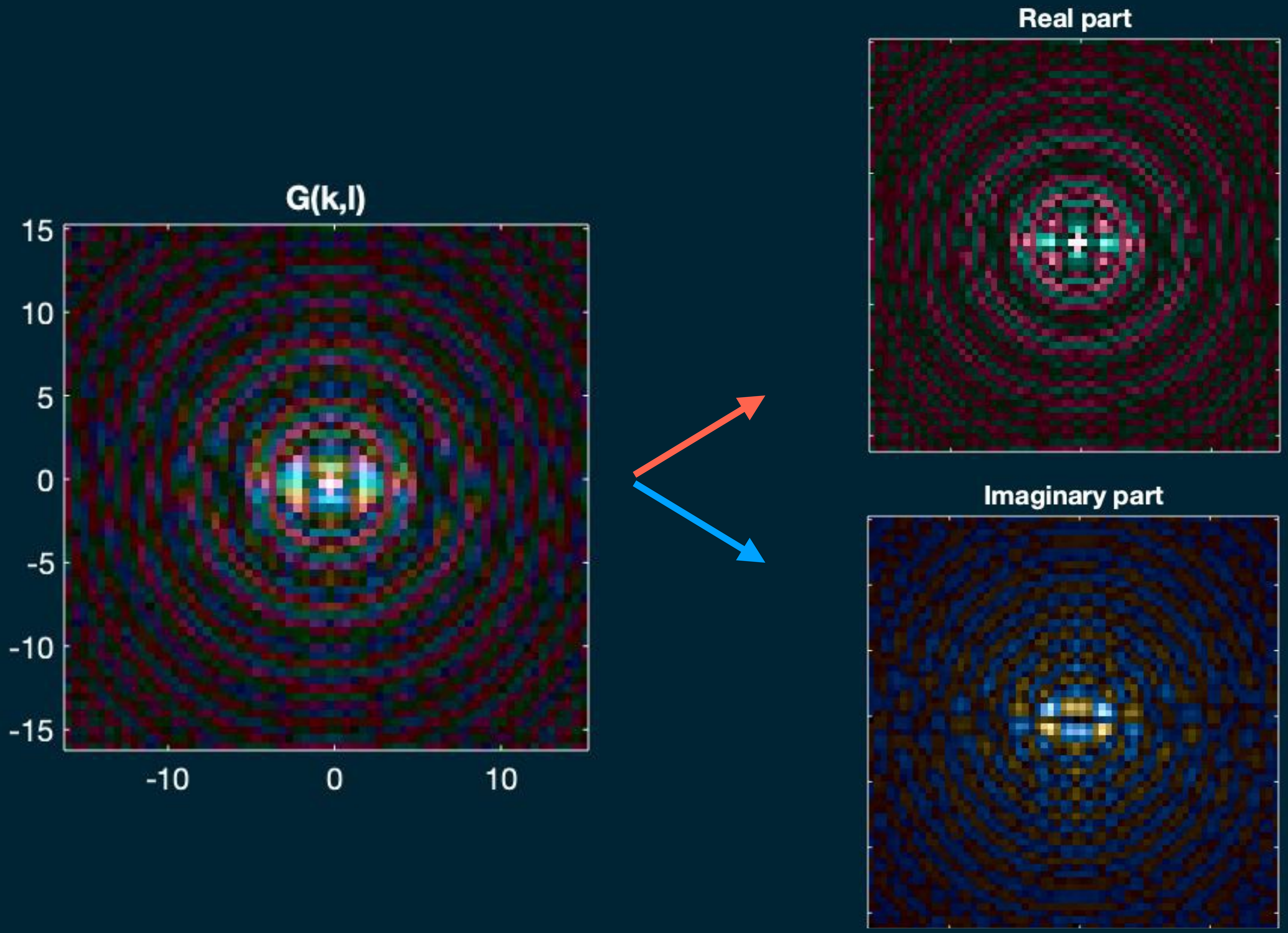
$$G(k, l) = \frac{1}{N} \sum_{i,j=-N/2}^{N/2-1} g(i, j)\, e^{-i2\pi(ik+jl)/N}$$

k, l are integers

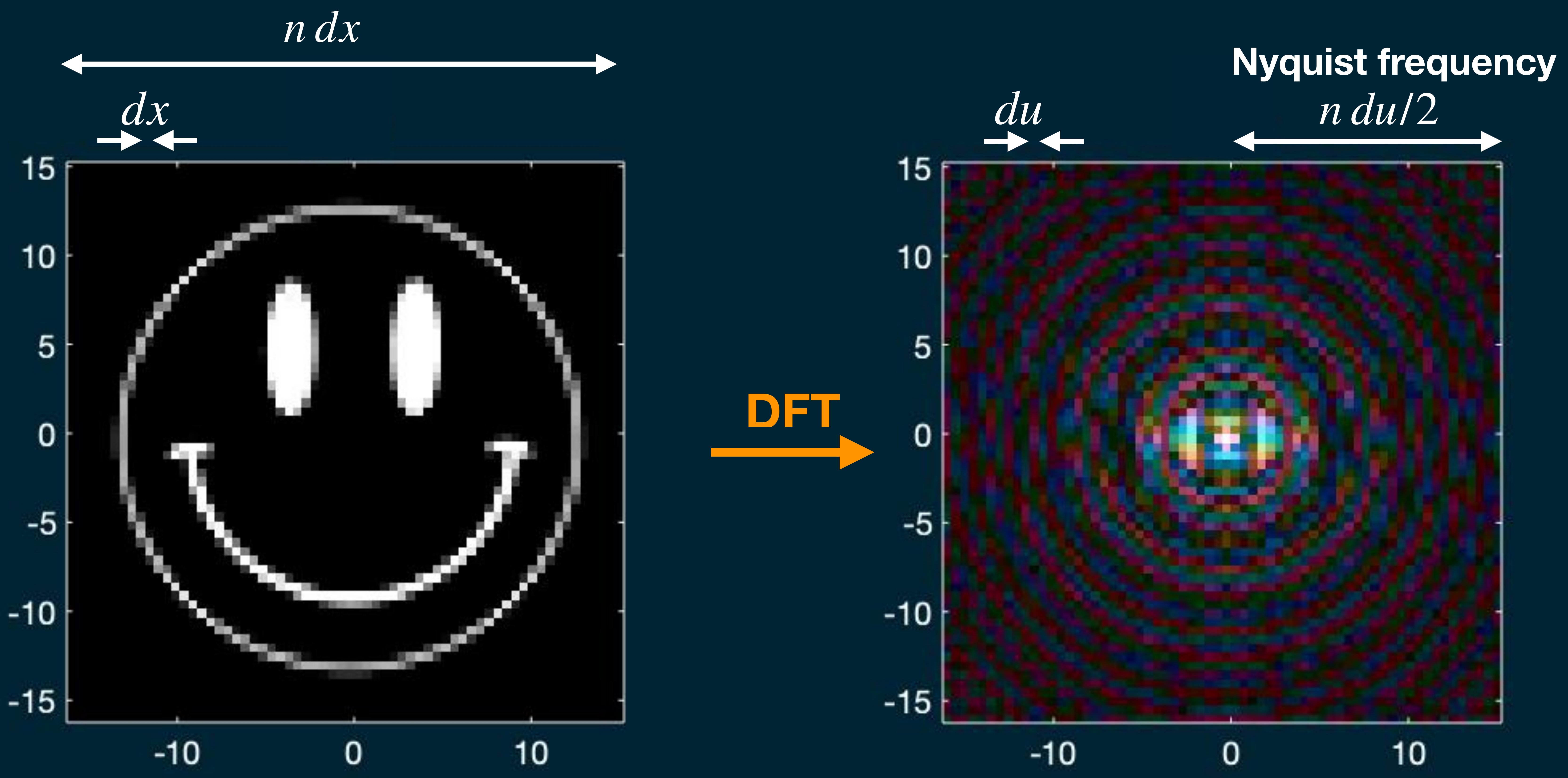
The DFT of a 32 x 32 pixel image has 32 x 32 complex pixel values



But the DFT of a real image has twofold redundancy



What is the pixel size of the transformed image?



$$dx = \frac{1}{n\,du} \qquad du = \frac{1}{n\,dx}$$

Note that the maximum accessible frequency $n\,du/2$ (the Nyquist frequency) corresponds to $2dx$, twice the sampling period.

The Fourier transform in three dimensions

The 3D transform

3D Fourier transform

$$G(u, v, w) = \iiint g(x, y, z) e^{-i2\pi(ux+vy+wz)} dx\, dy\, dz$$

3D Inverse Fourier transform

$$g(x, y, z) = \iiint G(u, v, w) e^{+i2\pi(ux+vy+wz)} du\, dv\, dw$$

3D Fourier transform properties

$$abc\, g(ax, by, cz) \rightarrow G(u/a, v/b, w/c)$$ Scale

$$g(x-a, y-b, z-c) \rightarrow G(u, v, w)e^{-i2\pi(au+bv+cw)}$$ Shift

$$g * h \rightarrow GH$$ Convolution

$$g(x', y', z') \rightarrow G(u', v', w')$$ Rotation

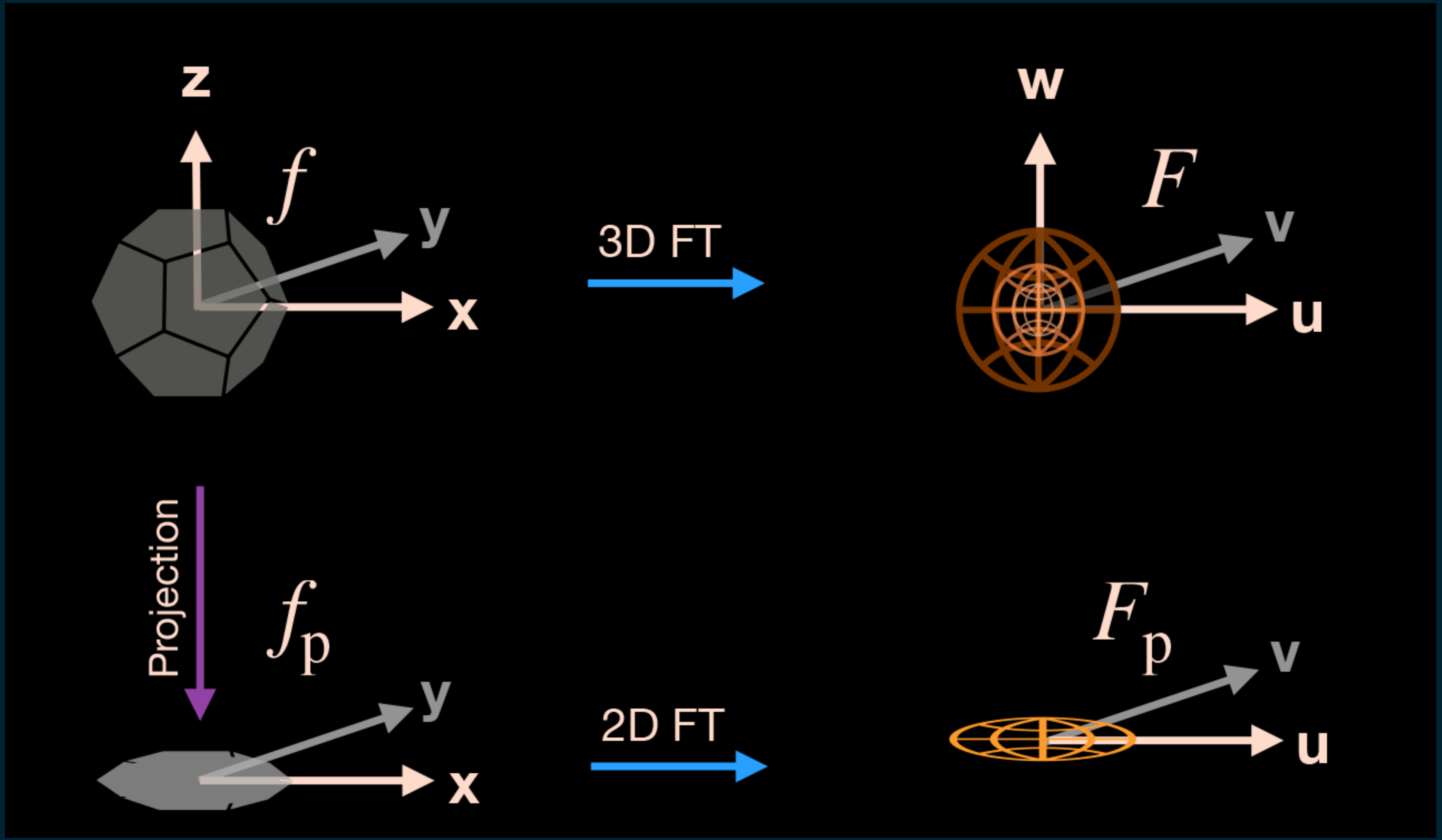
$$P_z\, g(x, y, z) \rightarrow G(u, v, 0)$$ Projection

The 3D Fourier slice theorem

Suppose we have a 3D density $f(x, y, z)$. The projection along z is a 2D image we'll call $f_p(x, y)$:

$$f_p(x, y) = \int f(x, y, z)\, dz$$

Let its Fourier transform be $F_p(u, v)$.

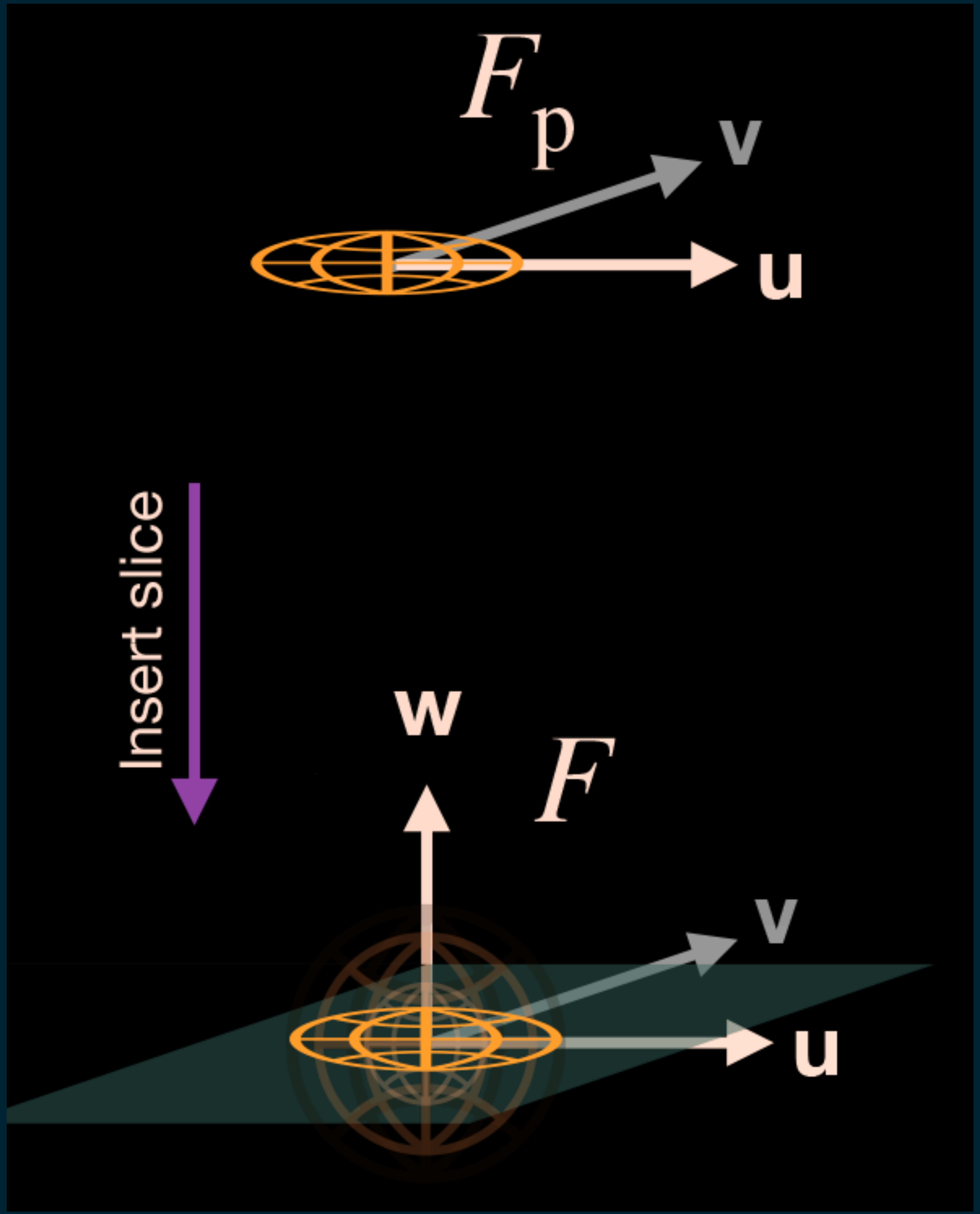


The 3D Fourier slice theorem

$F_p(u, v)$ is actually the same as the FT of the 3D volume, when we evaluate it at $w = 0$. In other words,
$F_p(u, v) = F(u, v, 0)$.

That's because we can separate out the integral over z to get

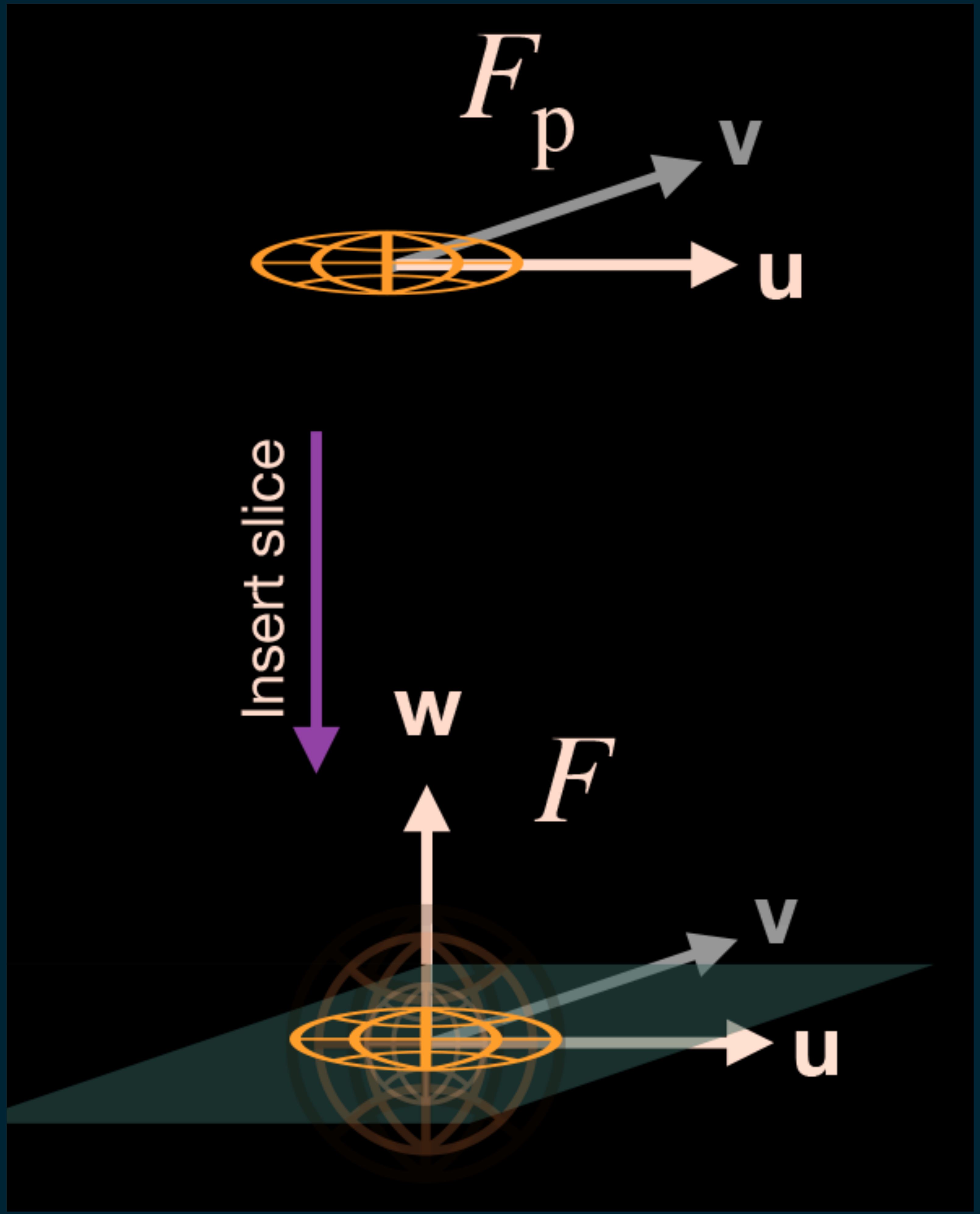
$$F(u, v, 0) = \iint \underbrace{\left(\int f(x, y, z)\, dz \right)}_{f_p(x, y)} e^{-i2\pi(ux+vy)} dx\, dy$$



Building up a 3D reconstruction

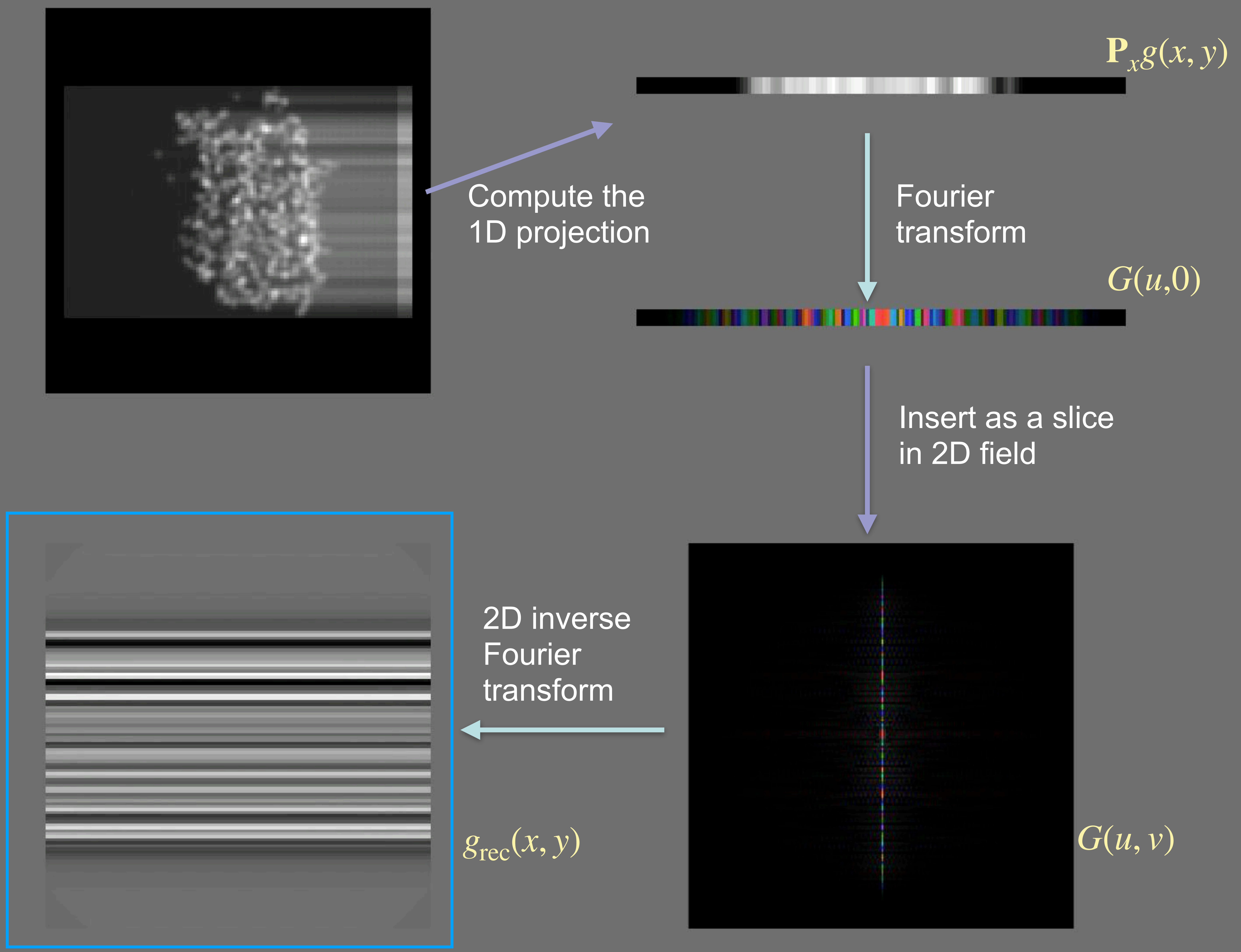
We can make use of the rotation property of the 3D FT to compute projections in many different directions, and insert the planes at the corresponding angles into the 3D Fourier volume.

If we've covered the volume completely then we can transform back to recover the complete 3D volume.



Part 3: Single-particle reconstruction

Reconstruction using the Fourier Slice Theorem

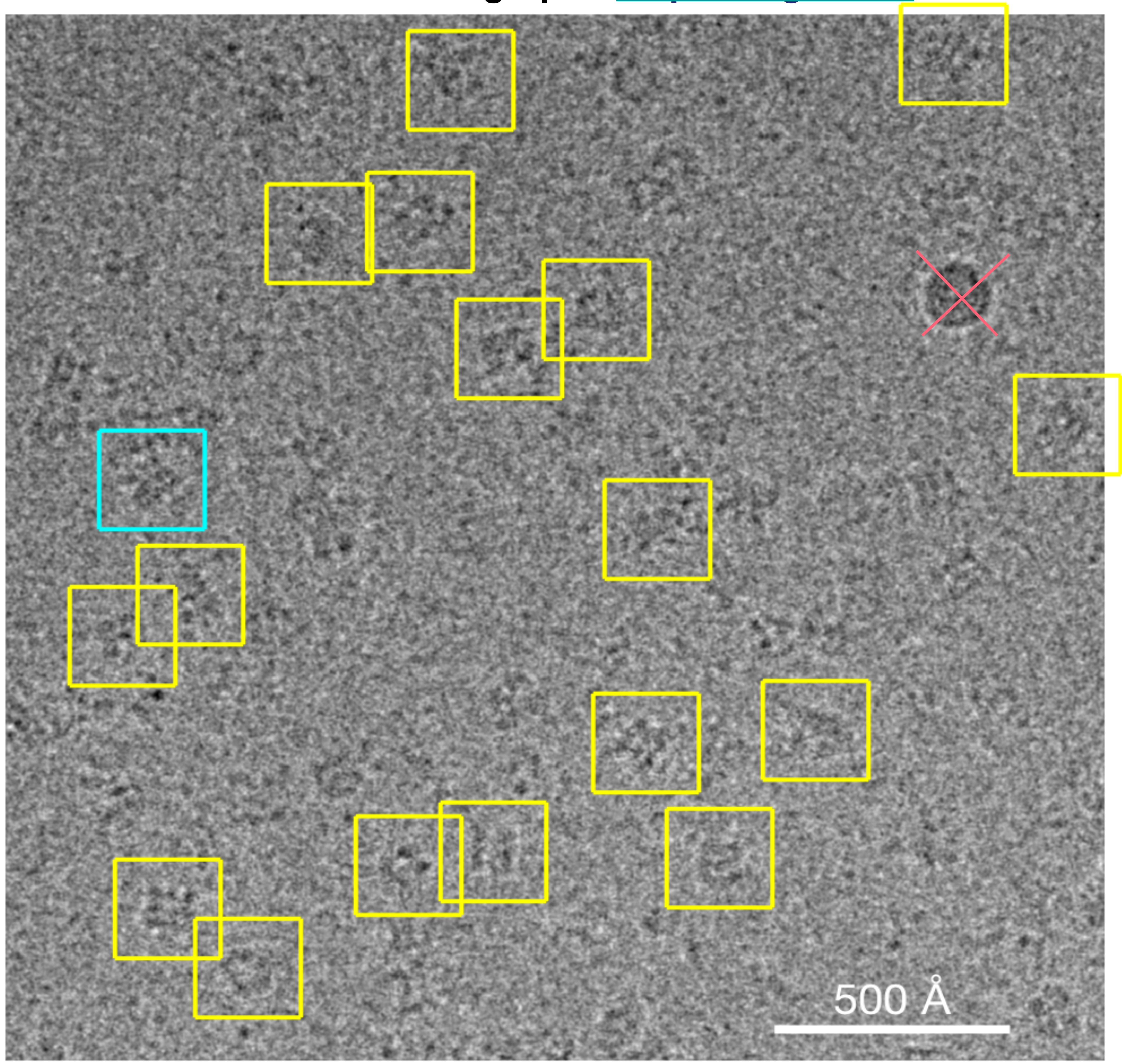


Determining the orientation angles: example from the TRPV1 dataset

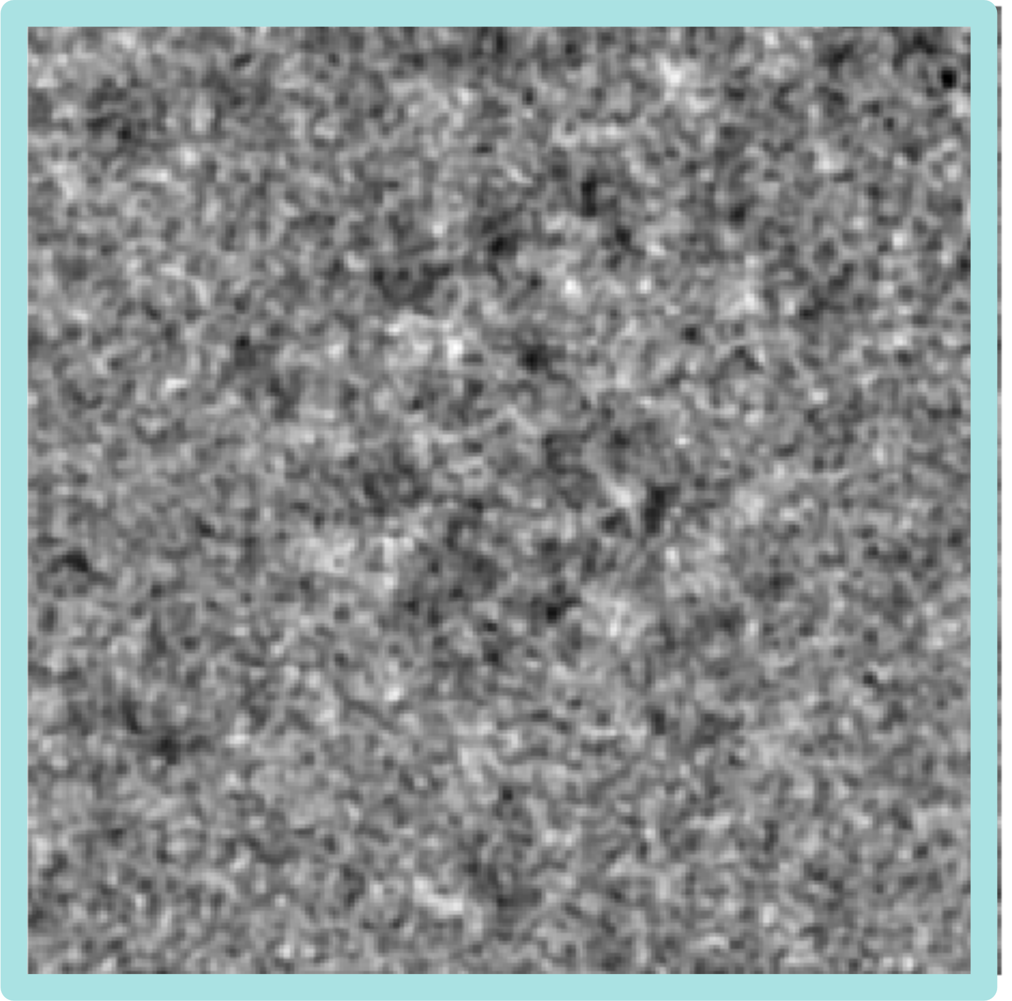
Structure of the TRPV1 ion channel determined by electron cryo-microscopy

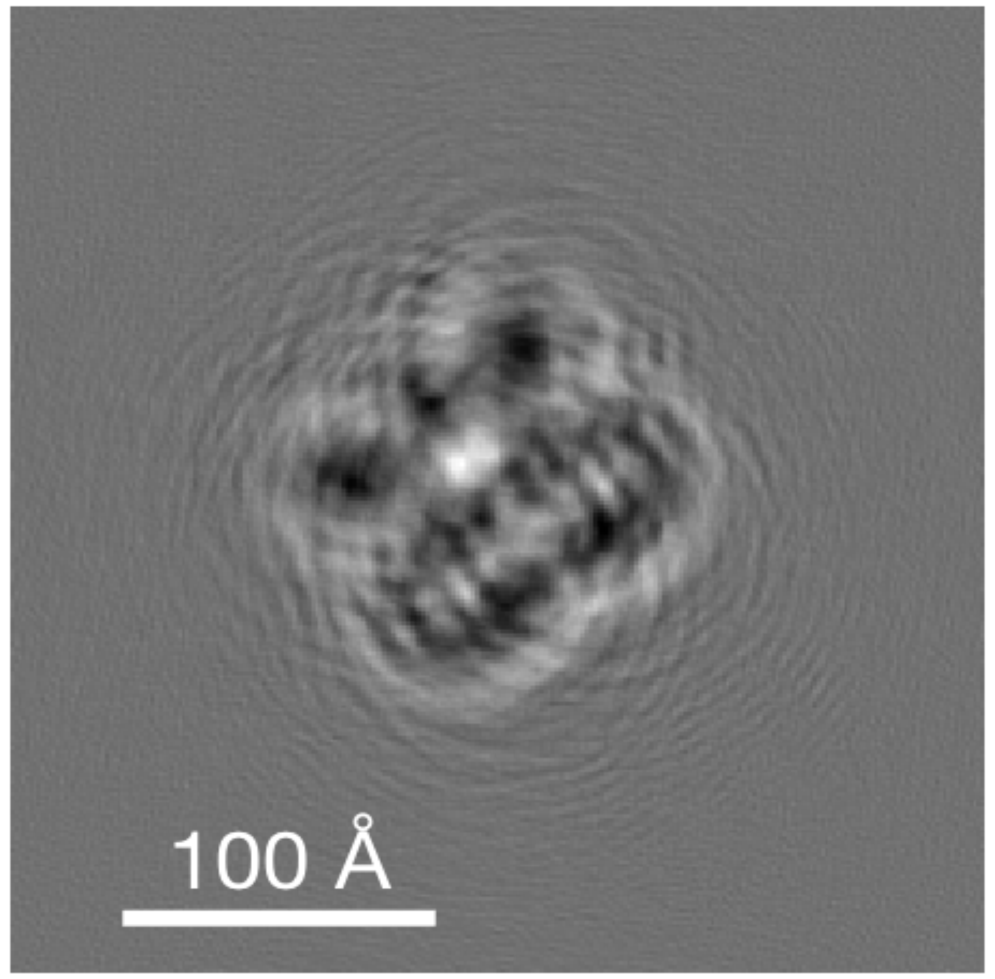
Maofu Liao[1]*, Erhu Cao[2]*, David Julius[2] & Yifan Cheng[1]

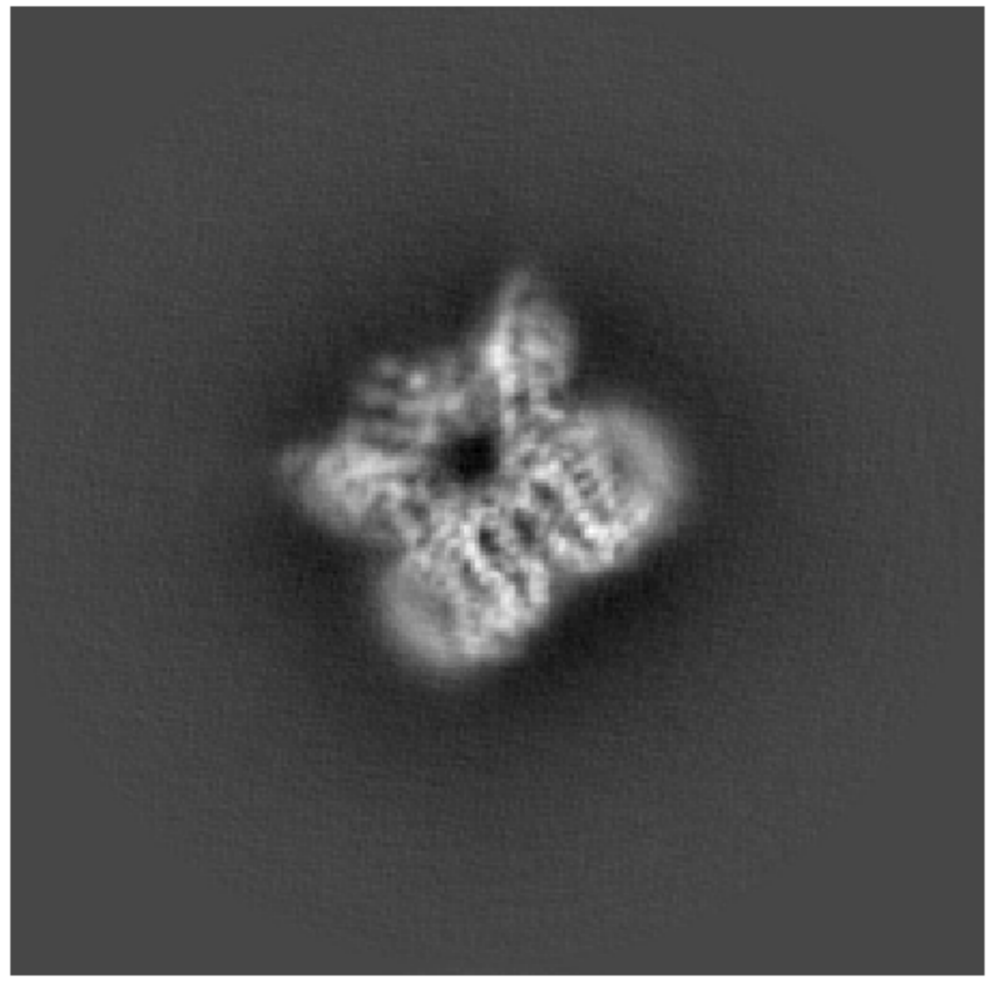
1/4 of a micrograph - empiar.org/10005

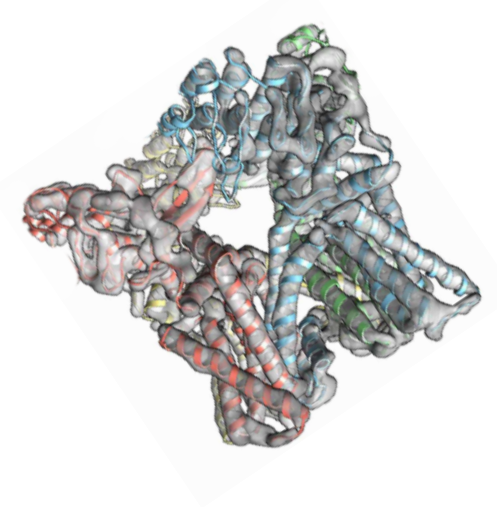


One particle image

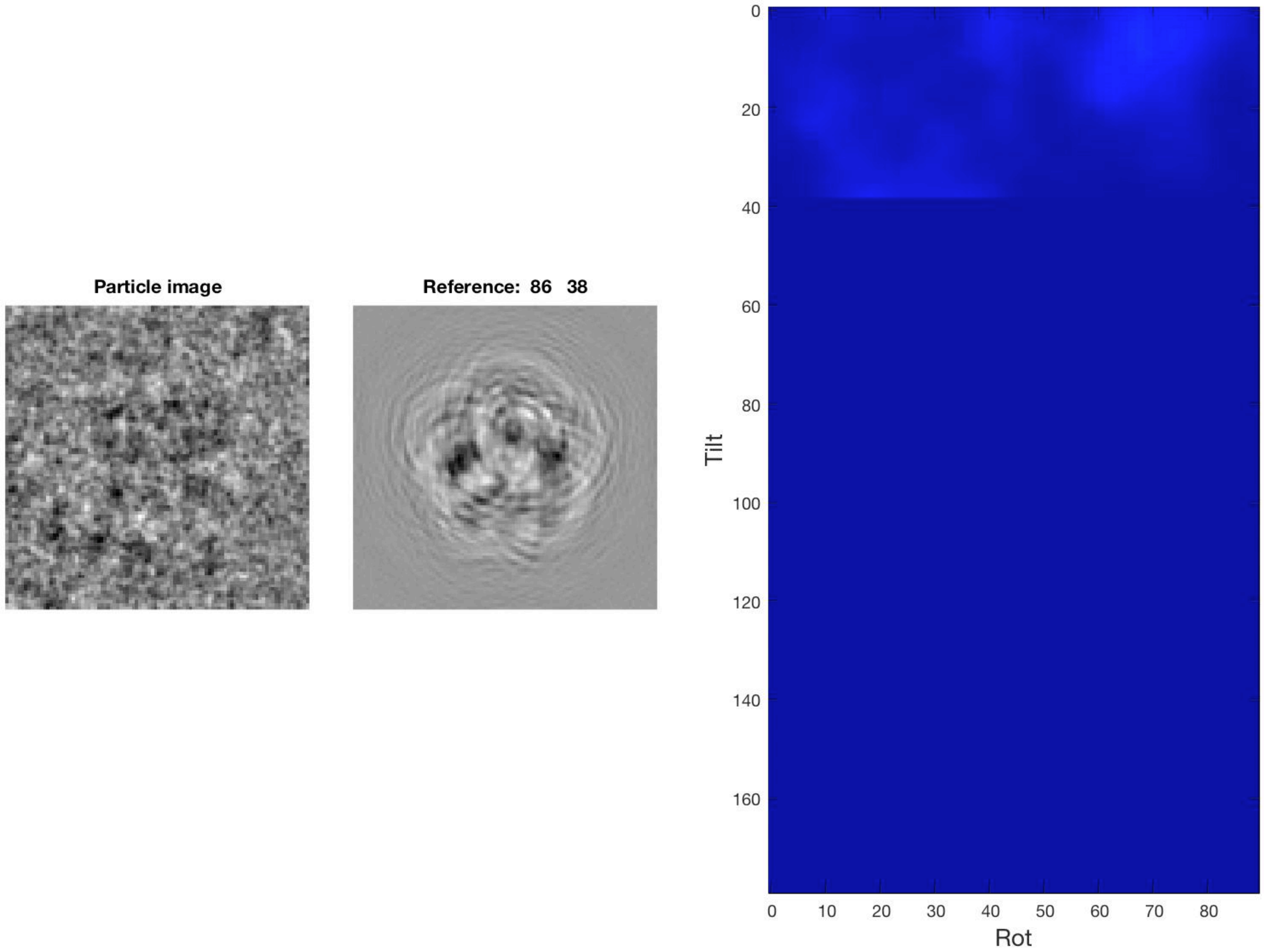






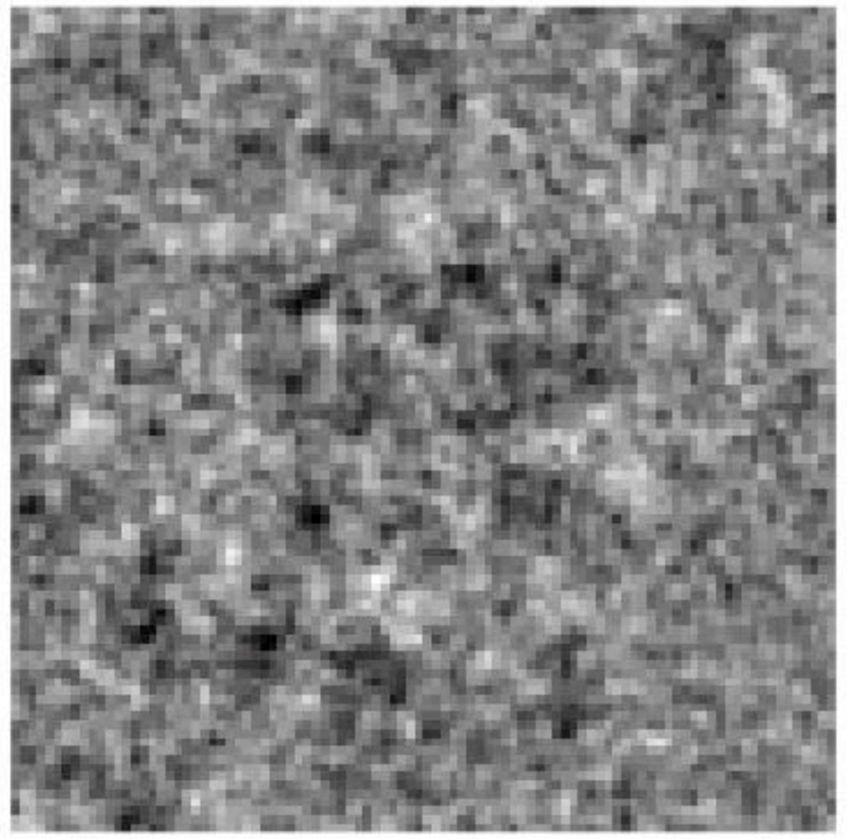


The probability of orientations $P(\phi \mid X, V)$ is remarkably sharp

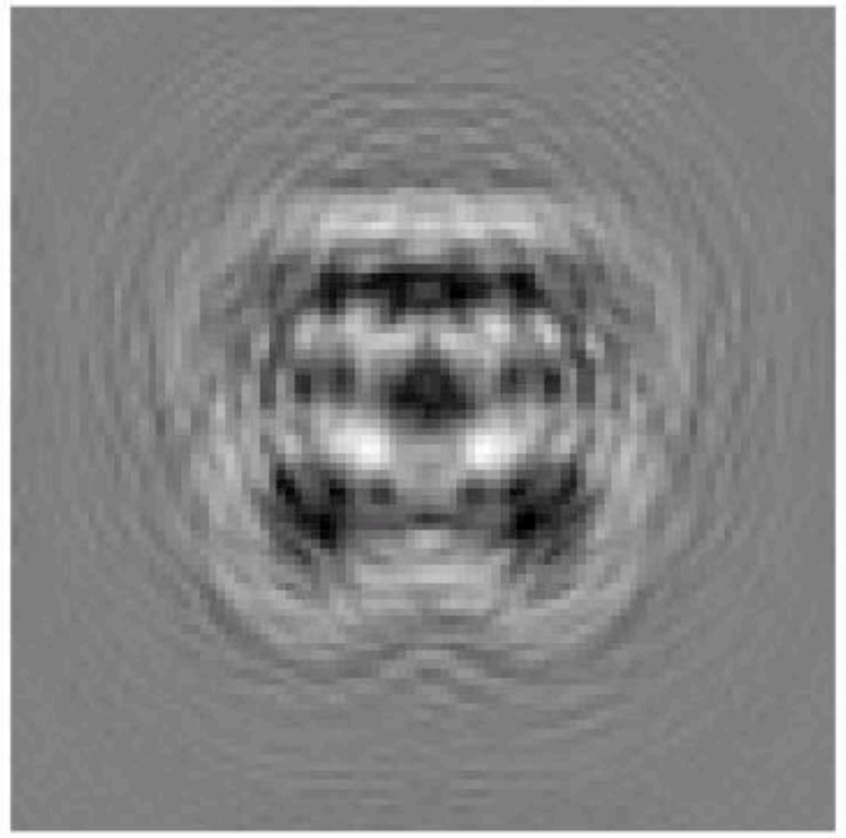


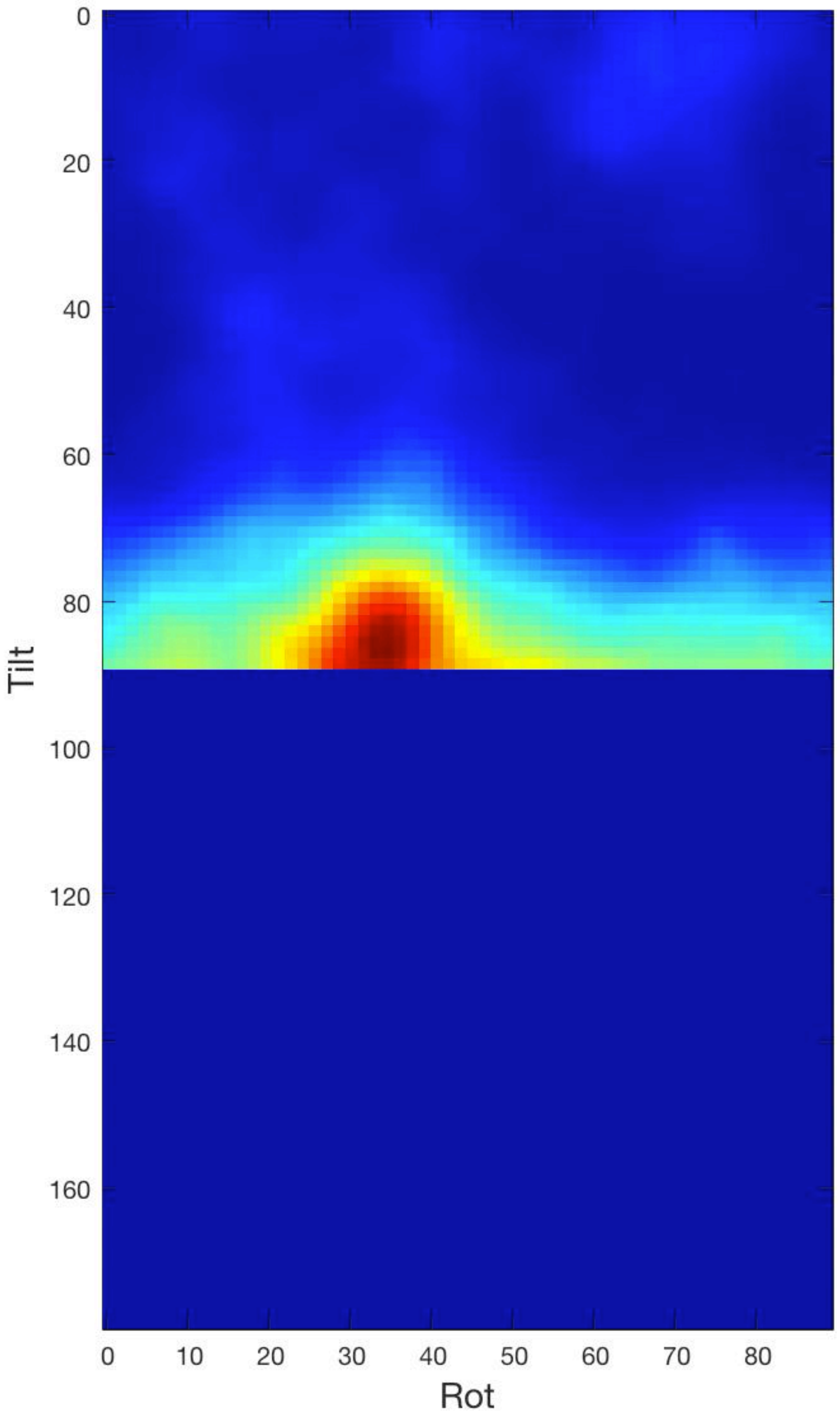
The probability of orientations $P(\phi | X, V)$ is remarkably sharp

Particle image



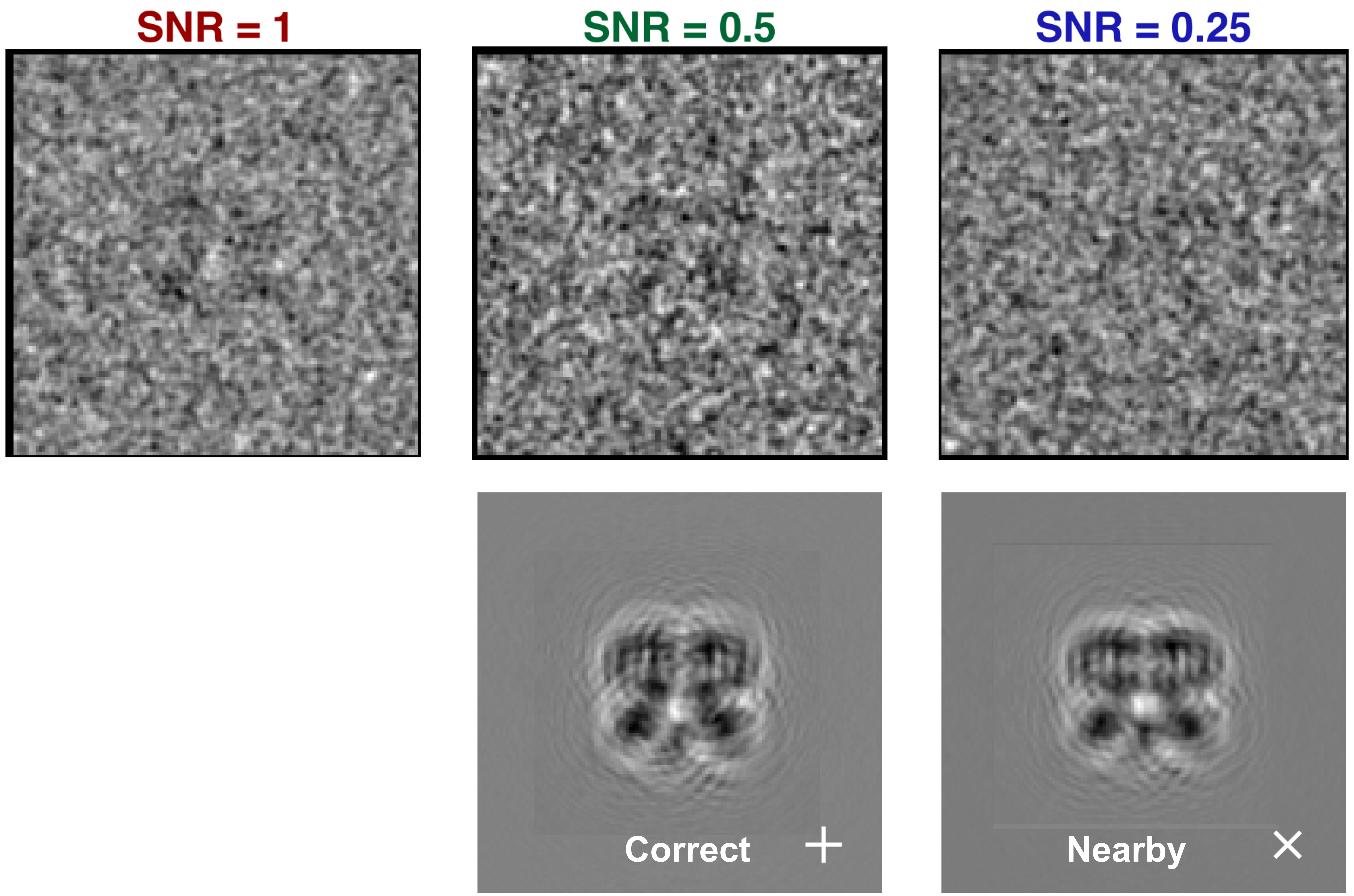
Reference: 89 89





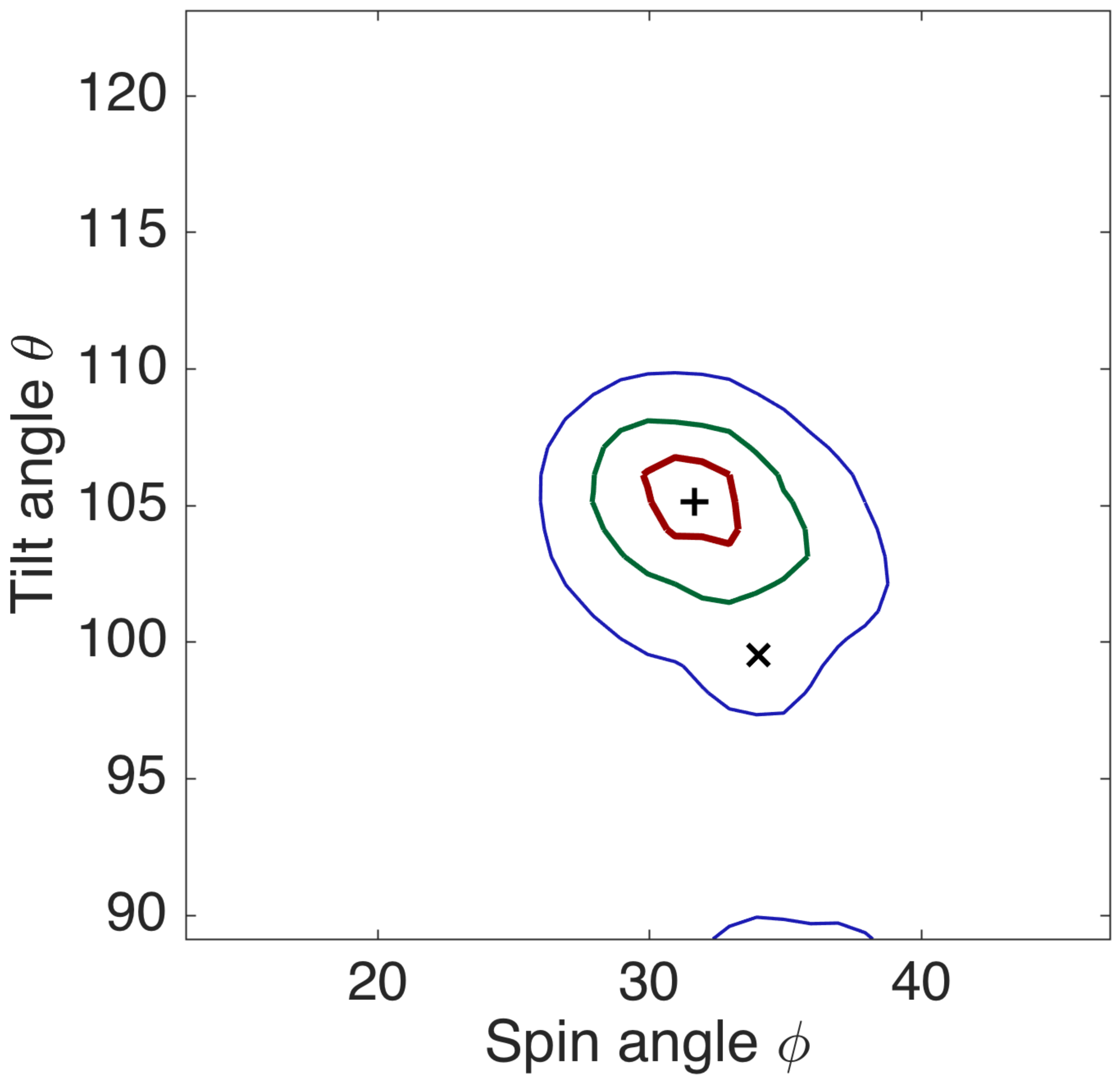
Precise angle determination by image matching

Single-particle images (simulated)

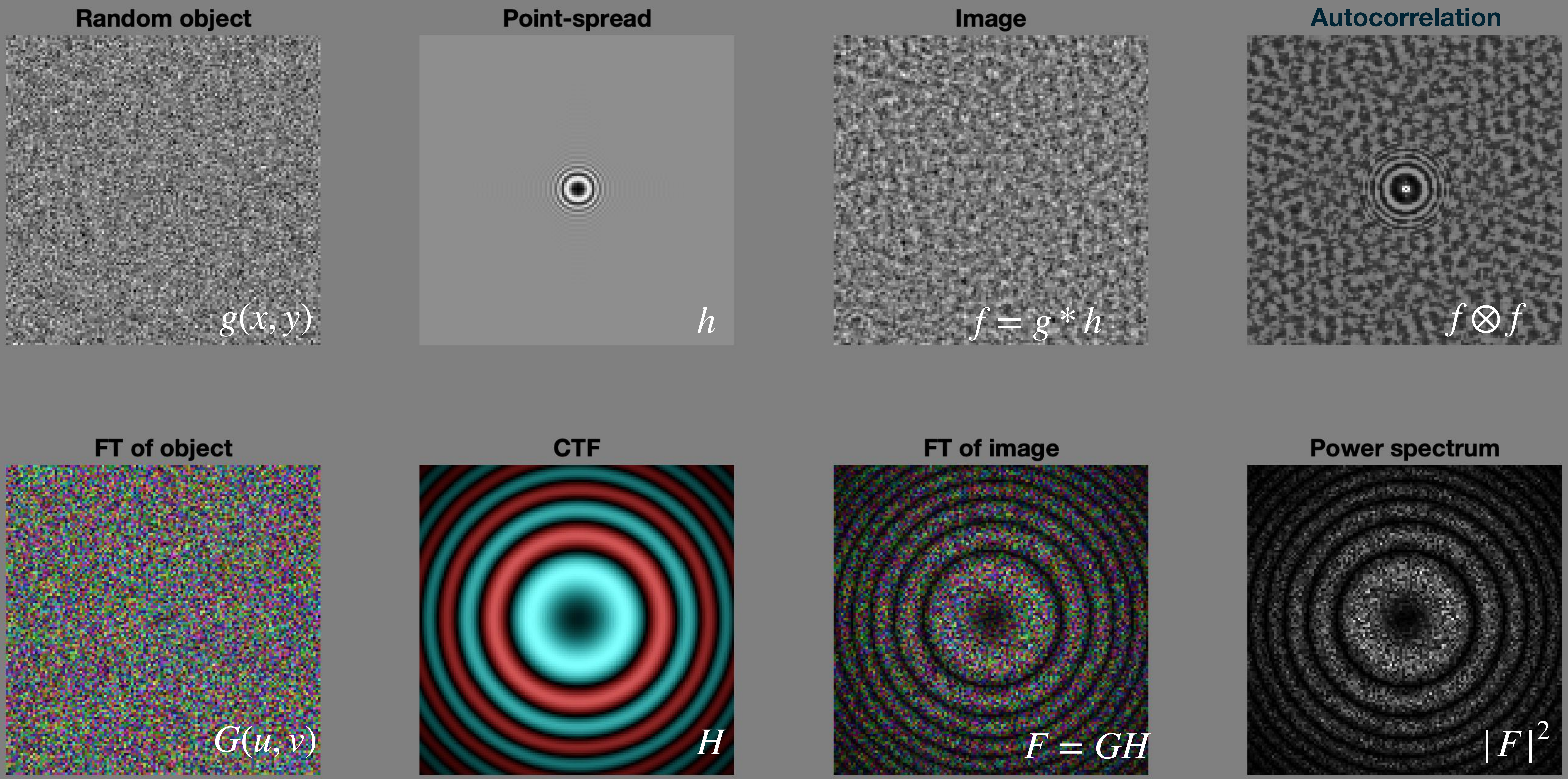


Noiseless projections

Angle accuracy (50th percentile)



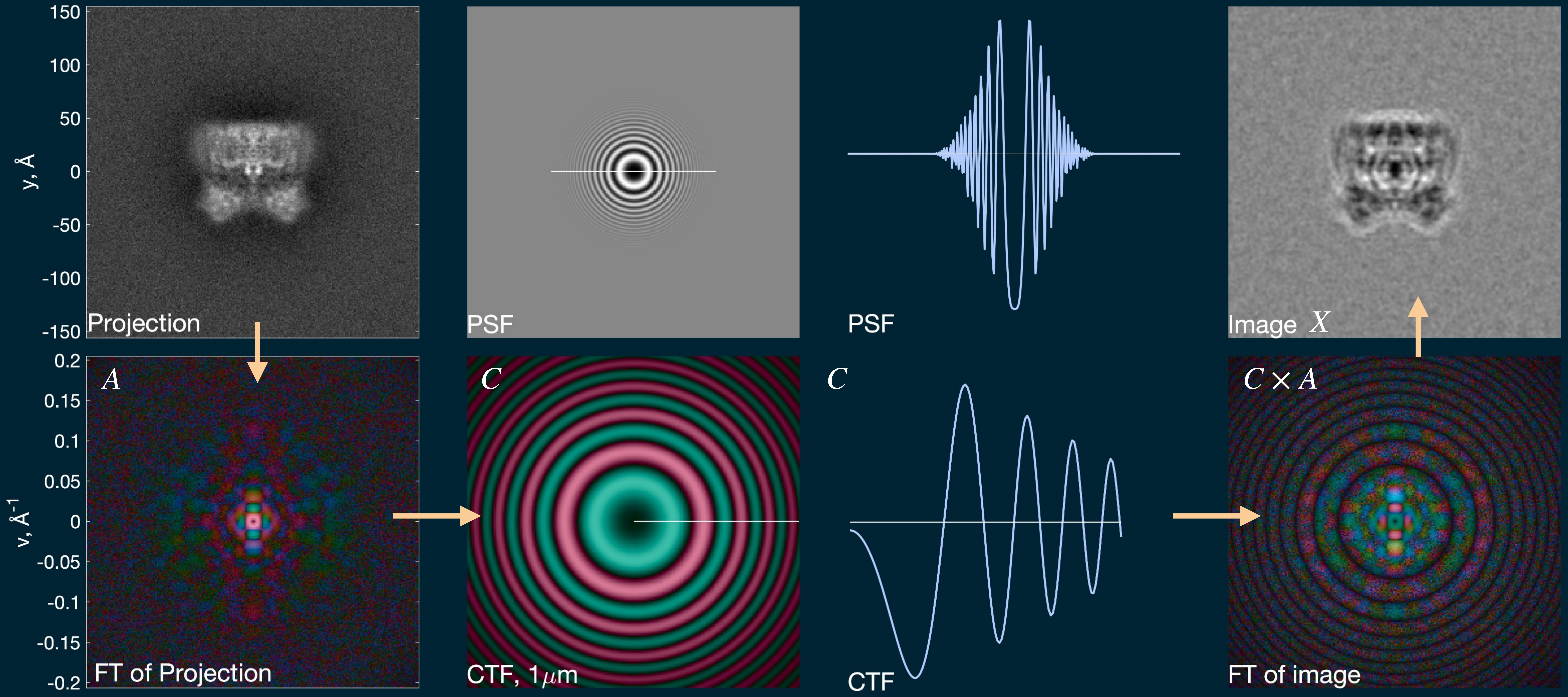
Visualizing the contrast transfer function



Modeling the CTF effect on an image

$$X = CA + N$$

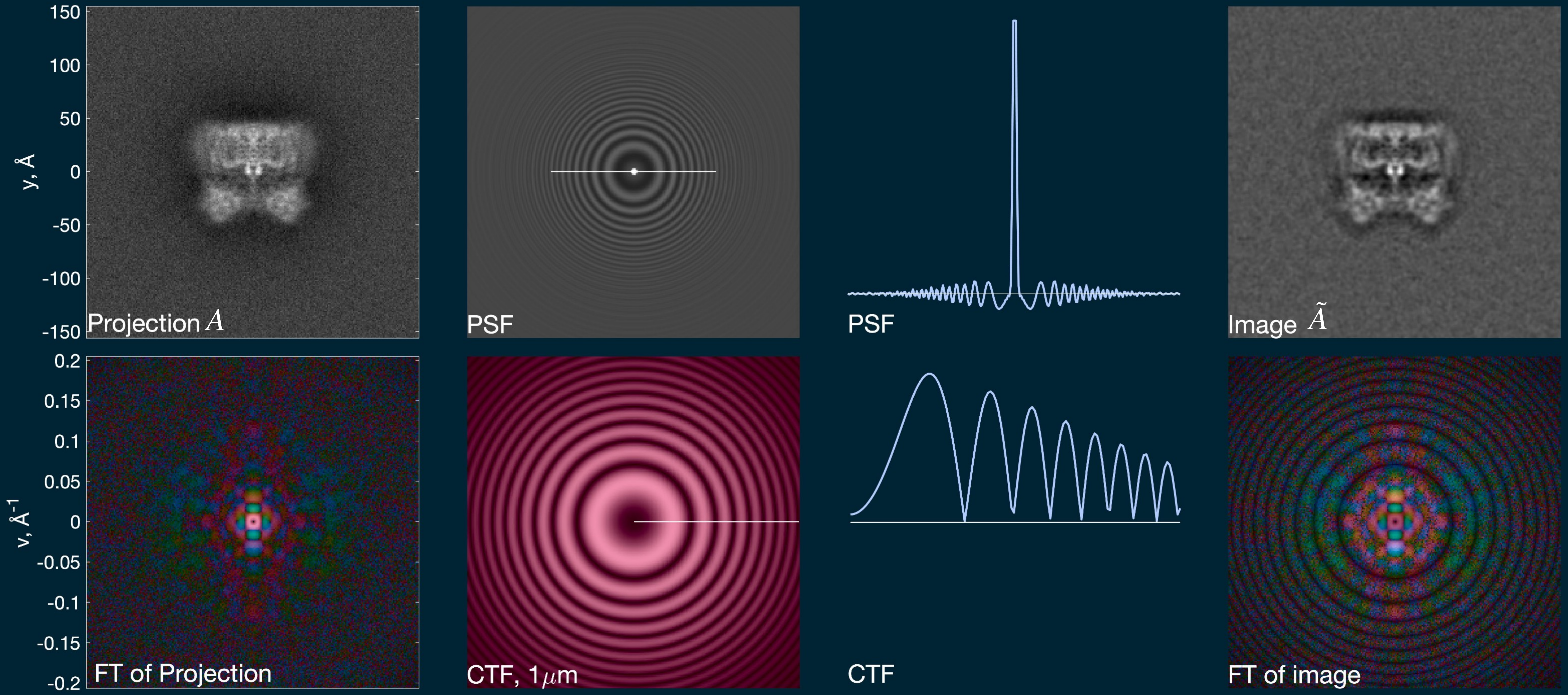
Can we do the deconvolution:
$\tilde{A} = X/C$??



How to undo the CTF effects?

1. **Phase flipping**

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



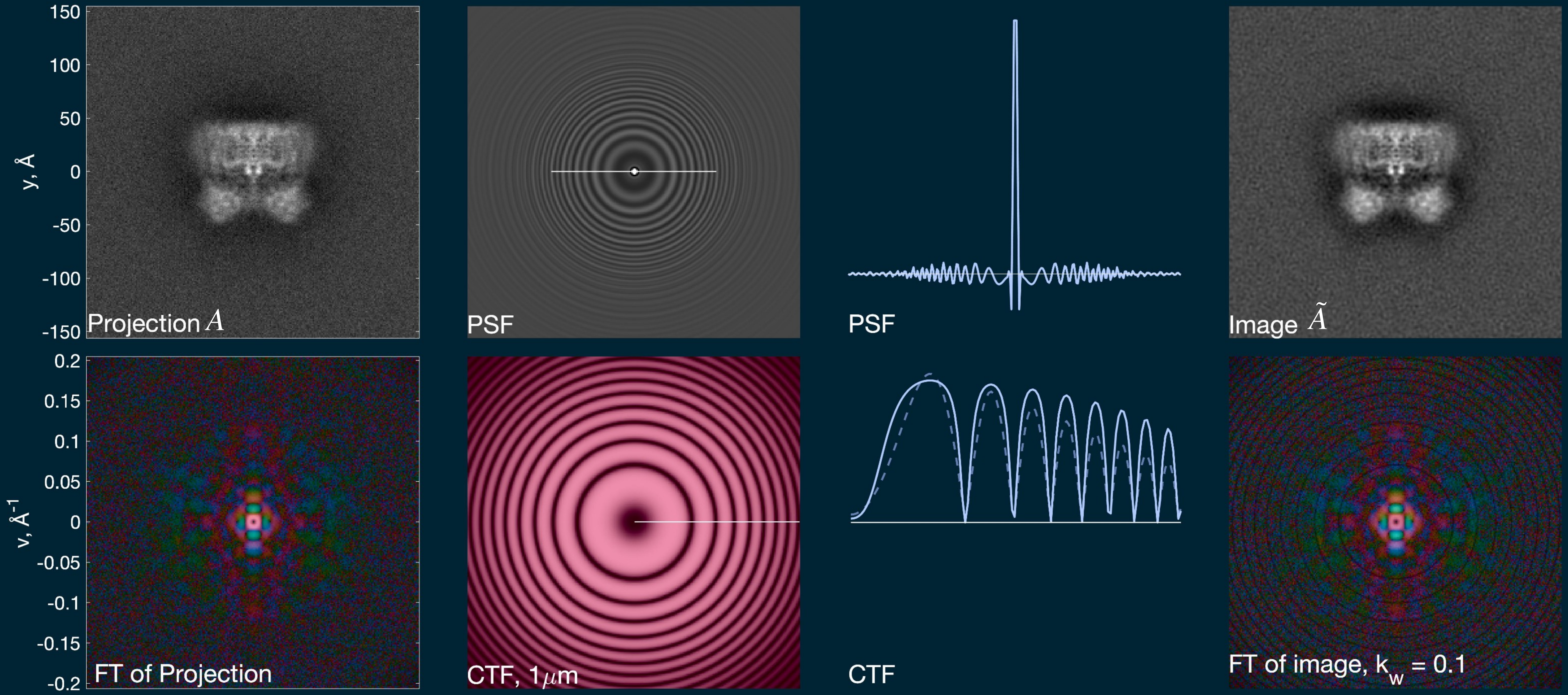
How to undo the CTF effects?

1. Phase flipping

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

2. Wiener filter

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



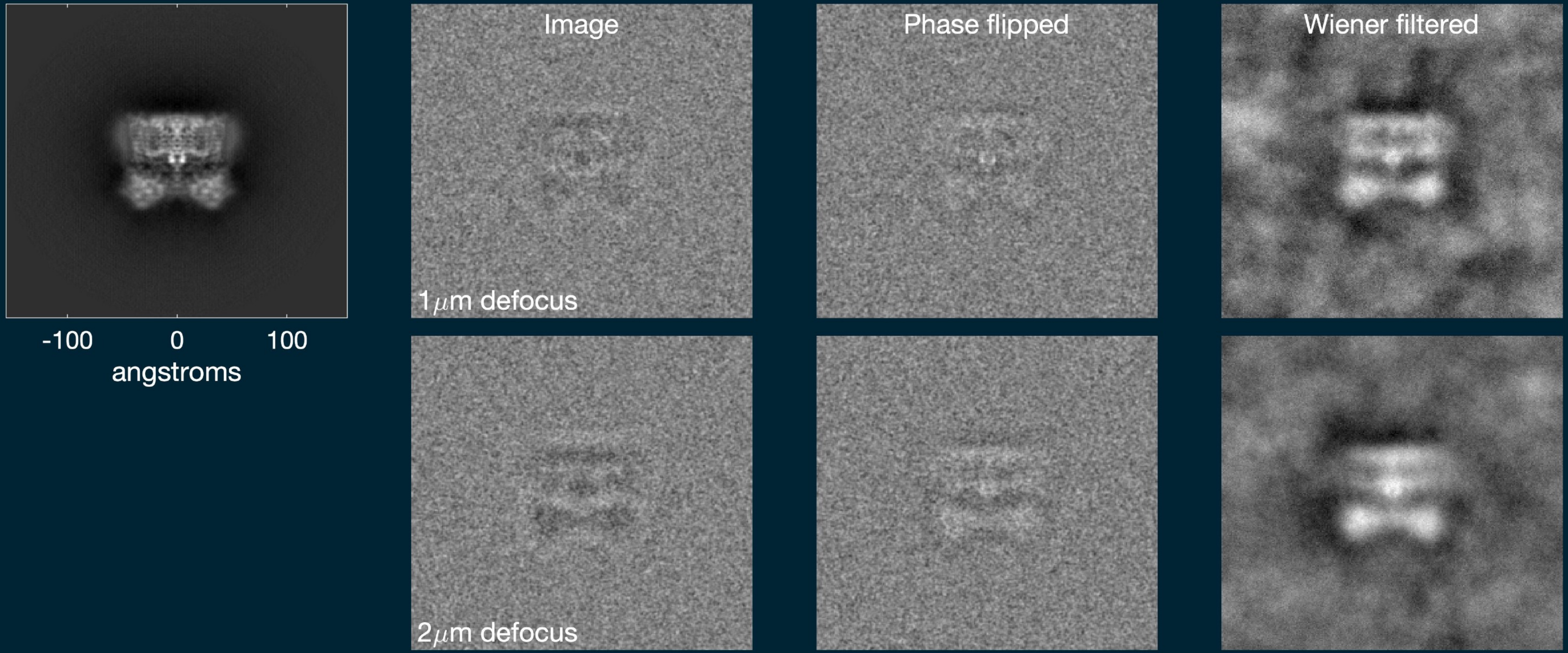
How to undo the CTF effects in noisy images?

1. **Phase flipping**

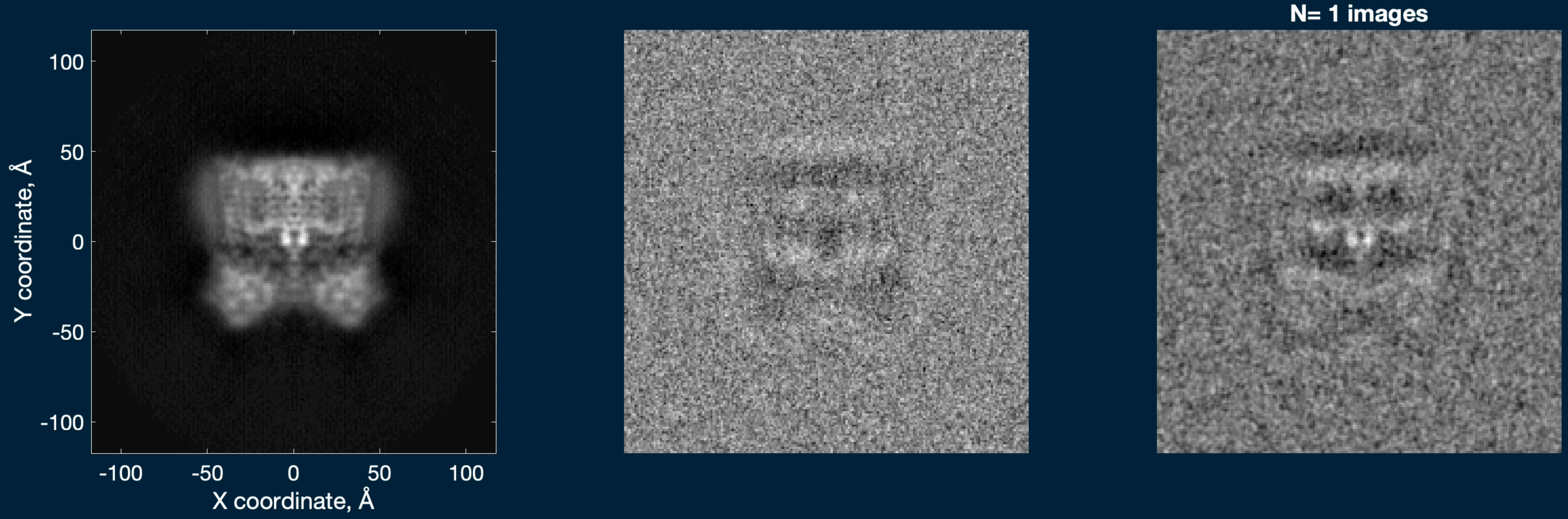
$$\tilde{A} = \mathrm{sgn}(C)X$$

2. **Wiener filter**

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



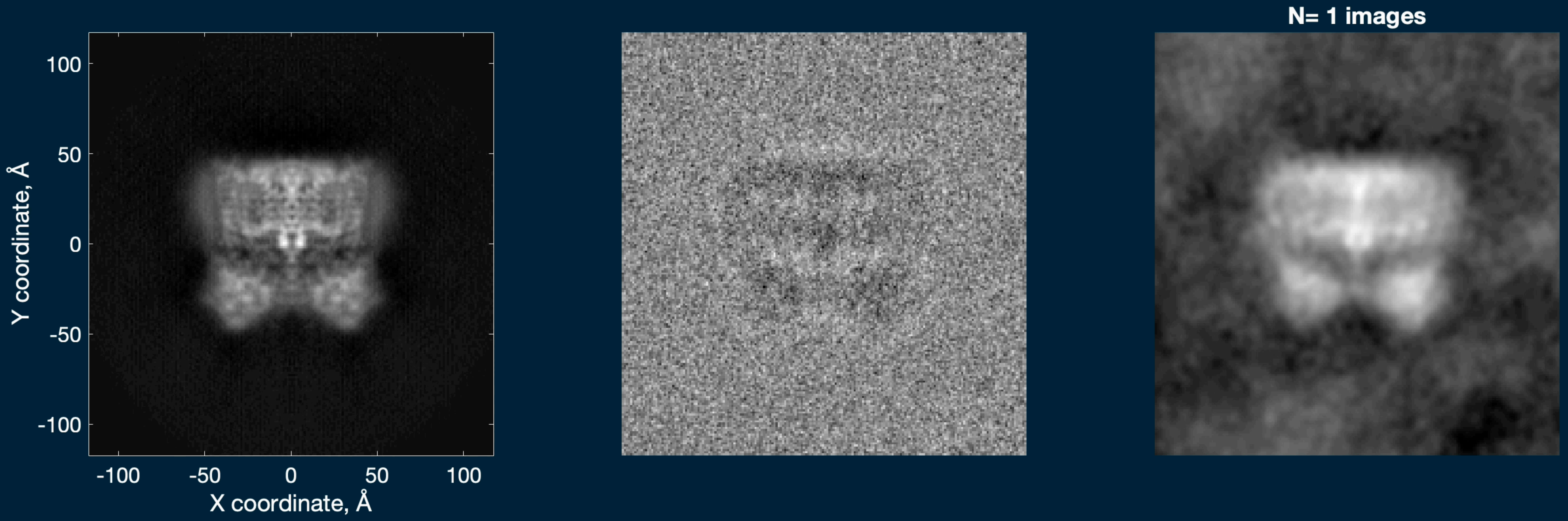
How to undo the CTF effects in noisy images?



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

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

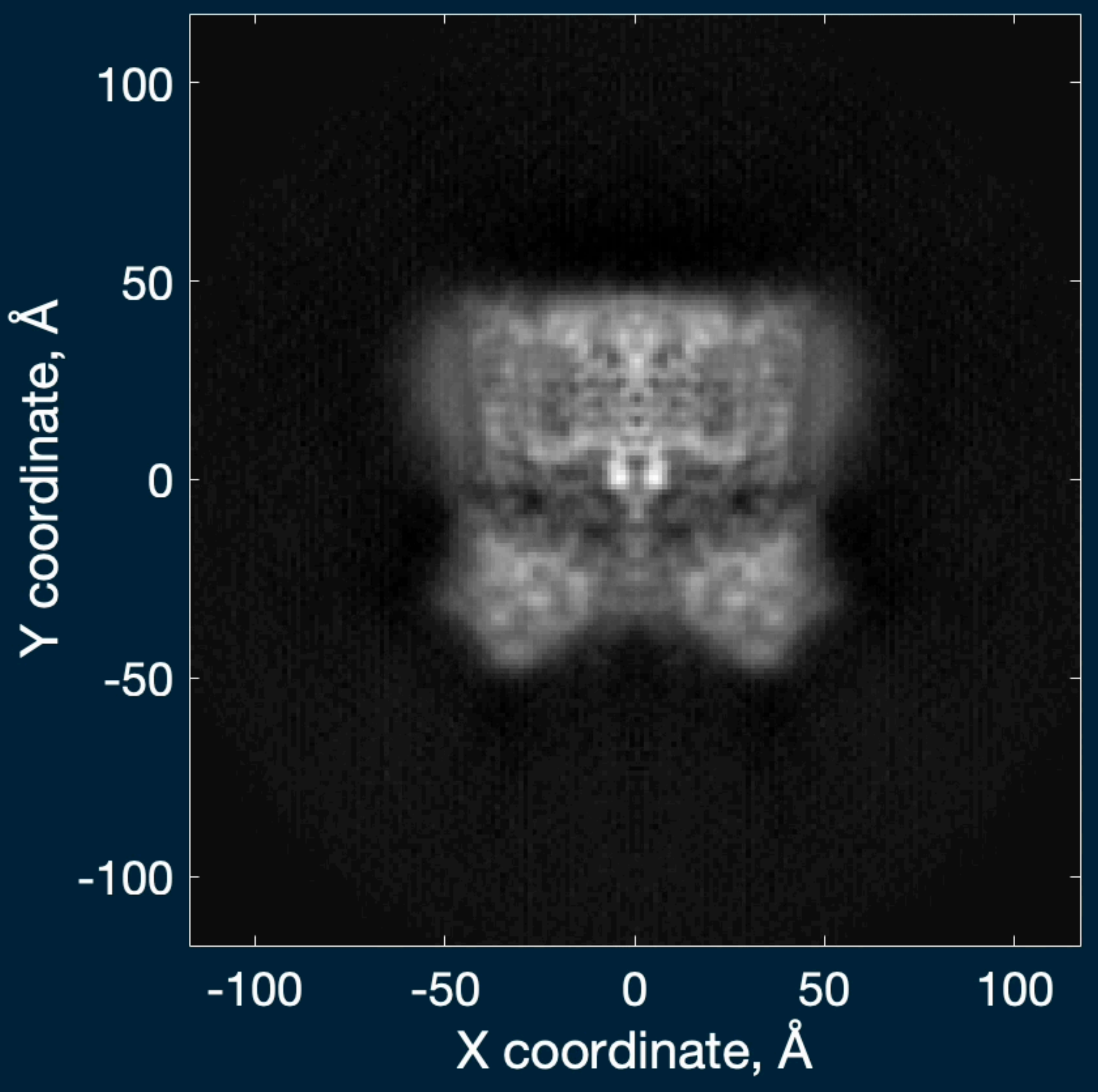
$$k(s) = 1/\mathrm{SNR} = \frac{|N|^2}{|A|^2}$$

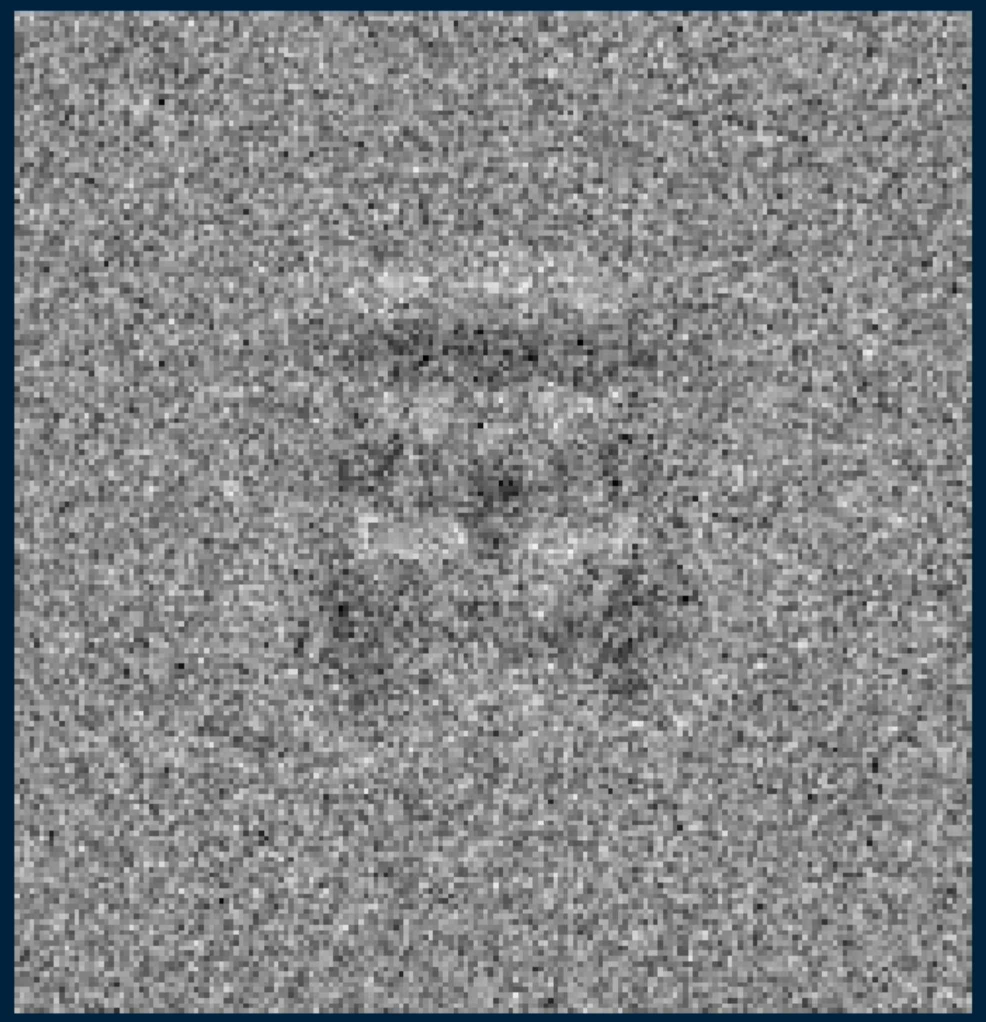
Image restoration when spectral SNR is known

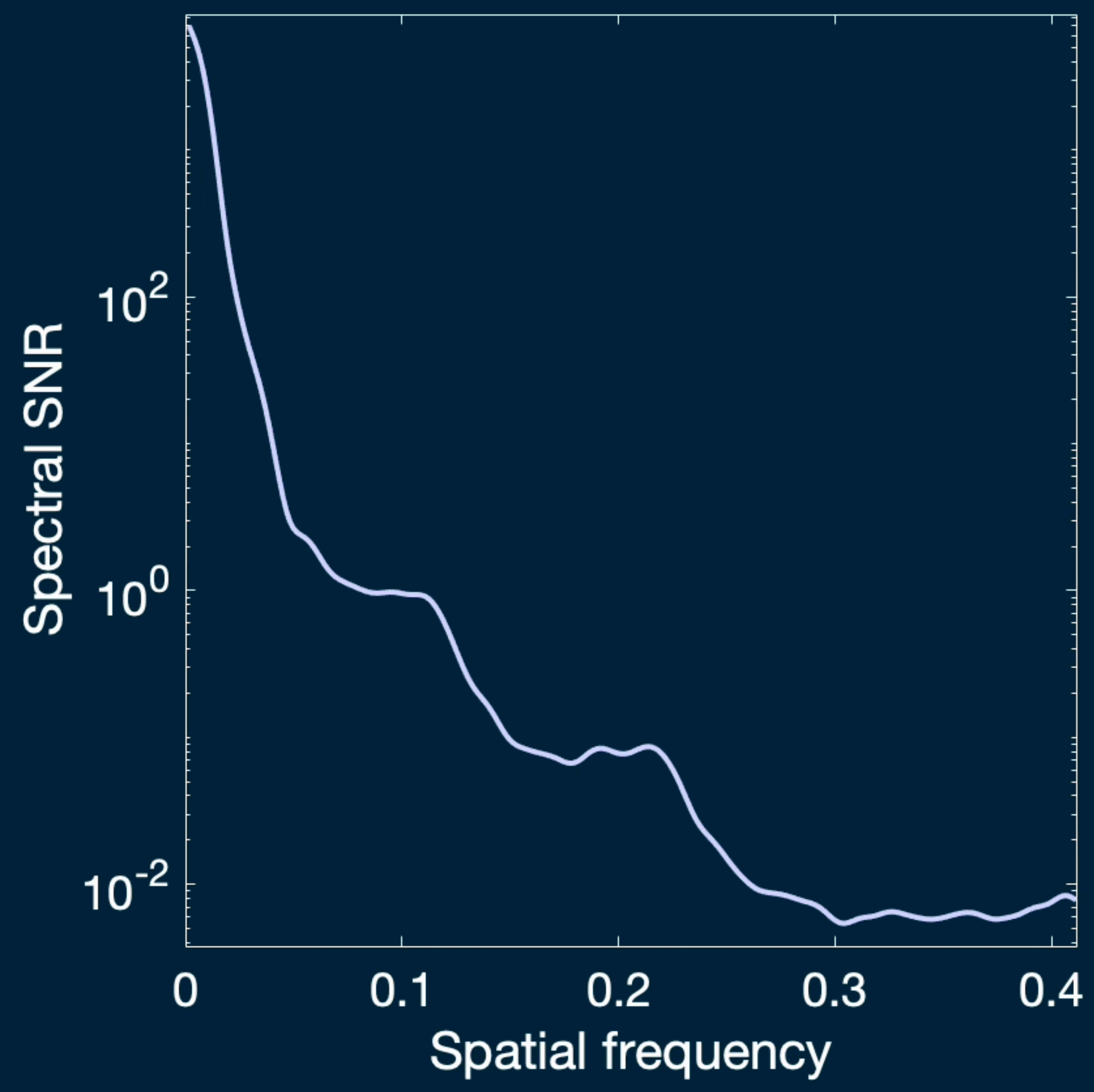
Restoration from multiple images

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

The defocus varies to fill in CTF zeros







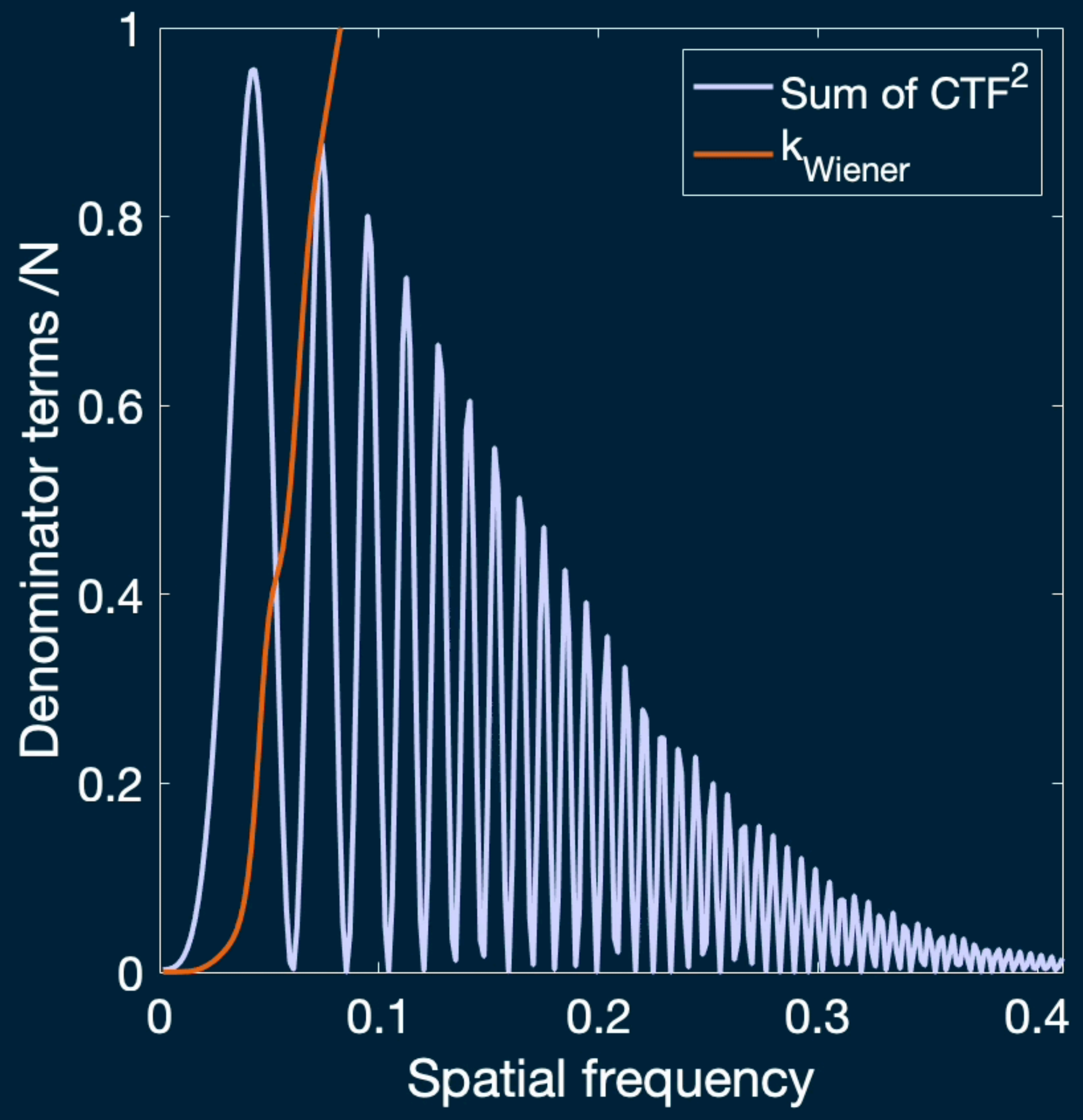
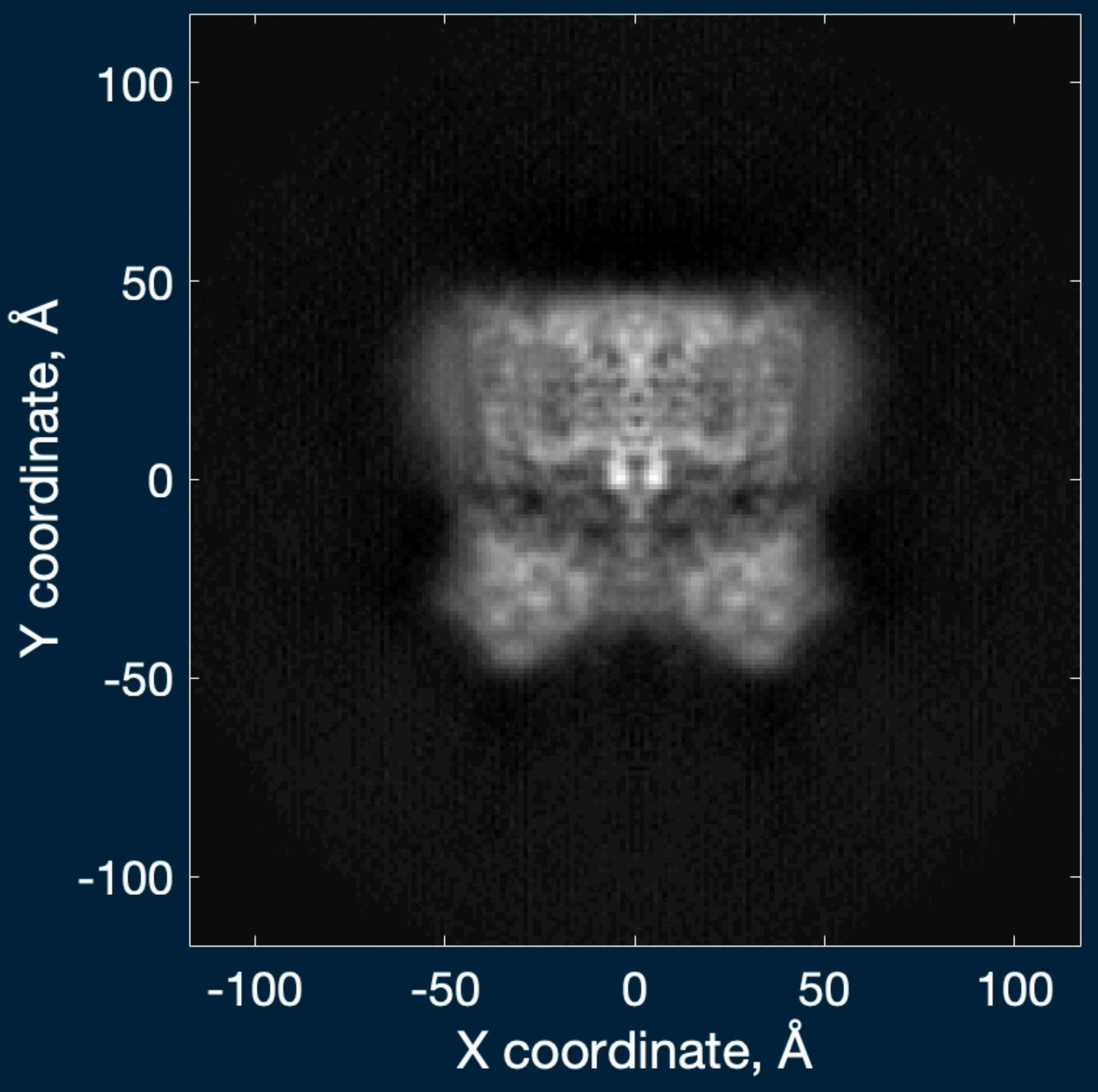


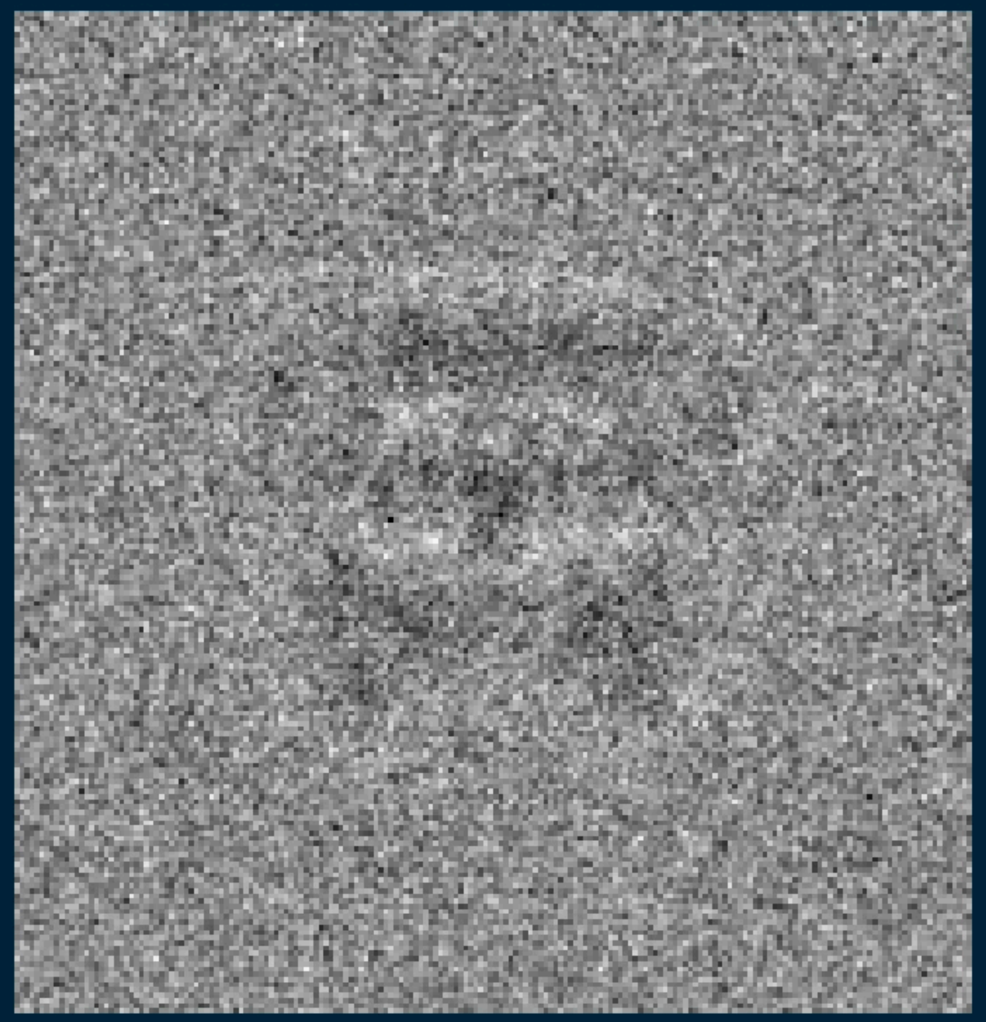
Image restoration when spectral SNR is known

Restoration from multiple images

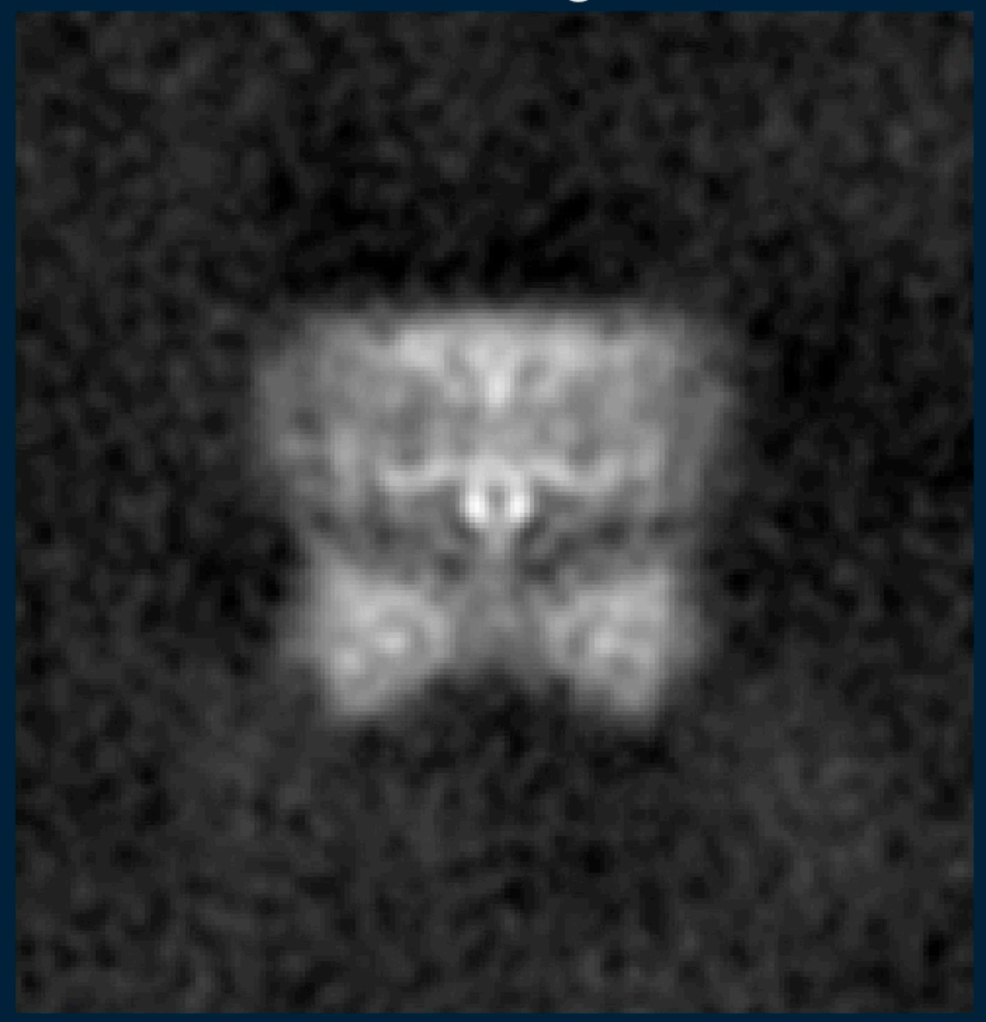
$$\tilde{A} = \frac{\sum_i^N C_i X_i}{\frac{1}{\mathrm{SSNR}} + \sum_i^N C_i^2}$$

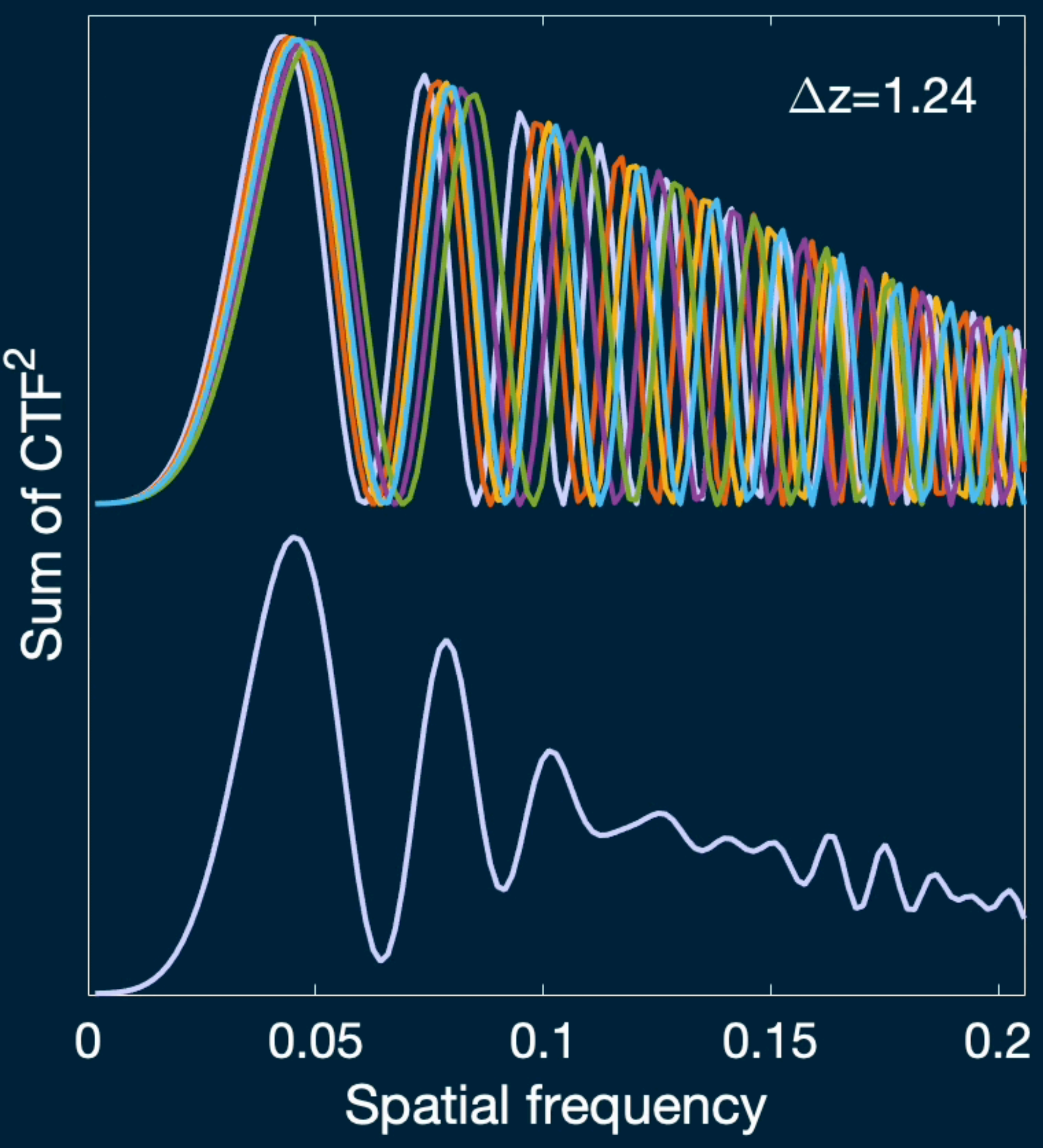
The defocus varies to fill in CTF zeros





N= 6 images





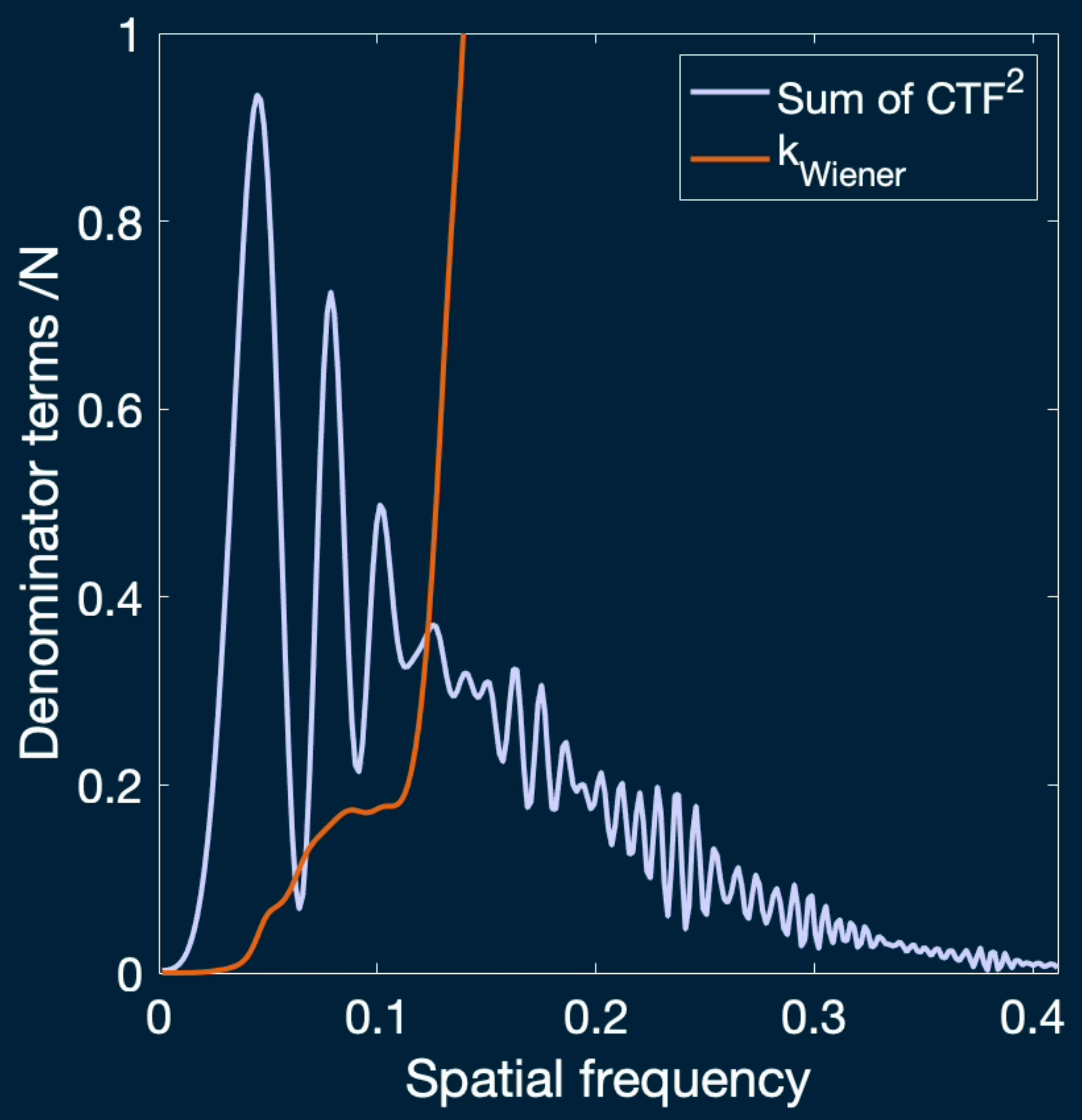
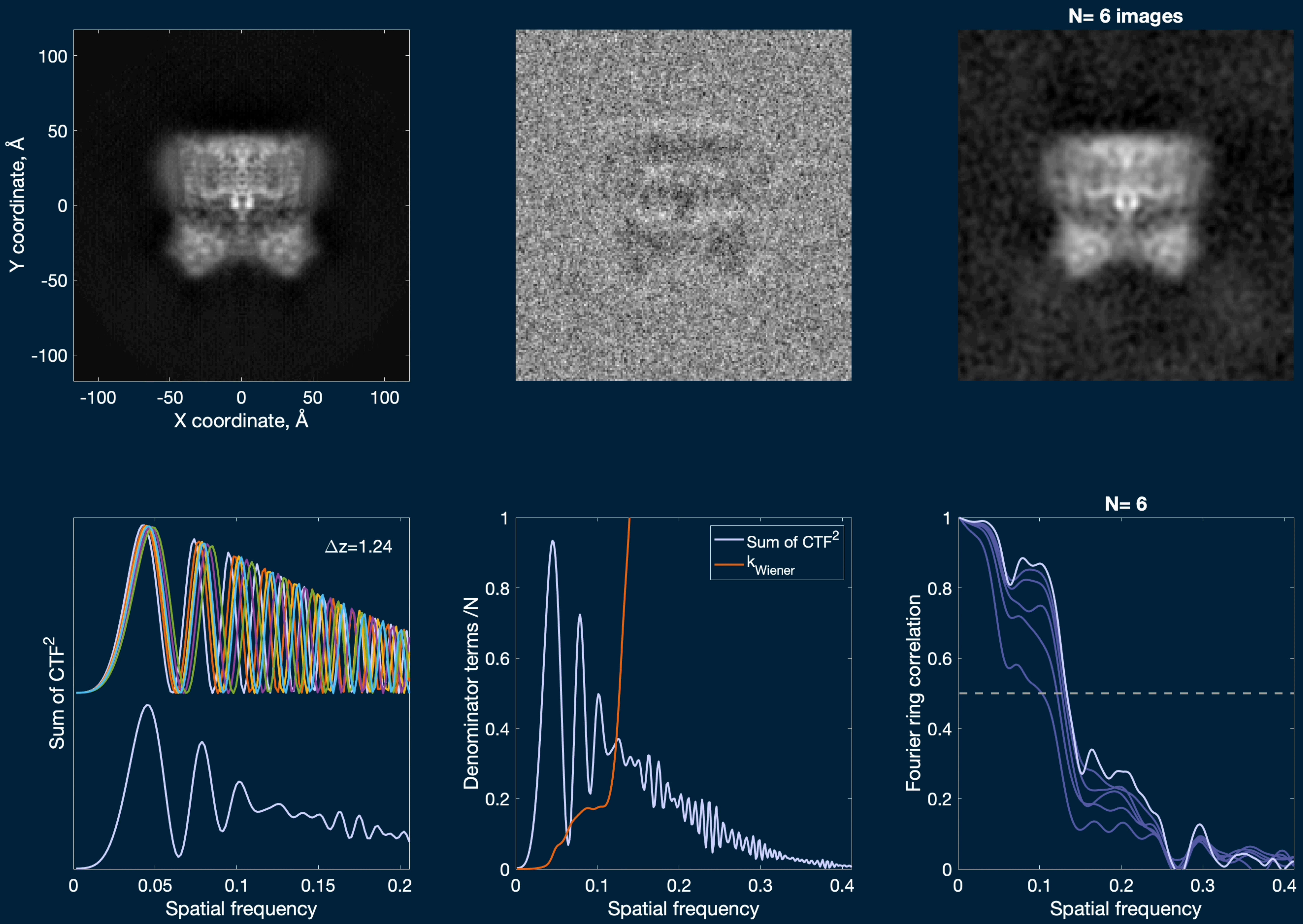


Image restoration when spectral SNR is known

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



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:

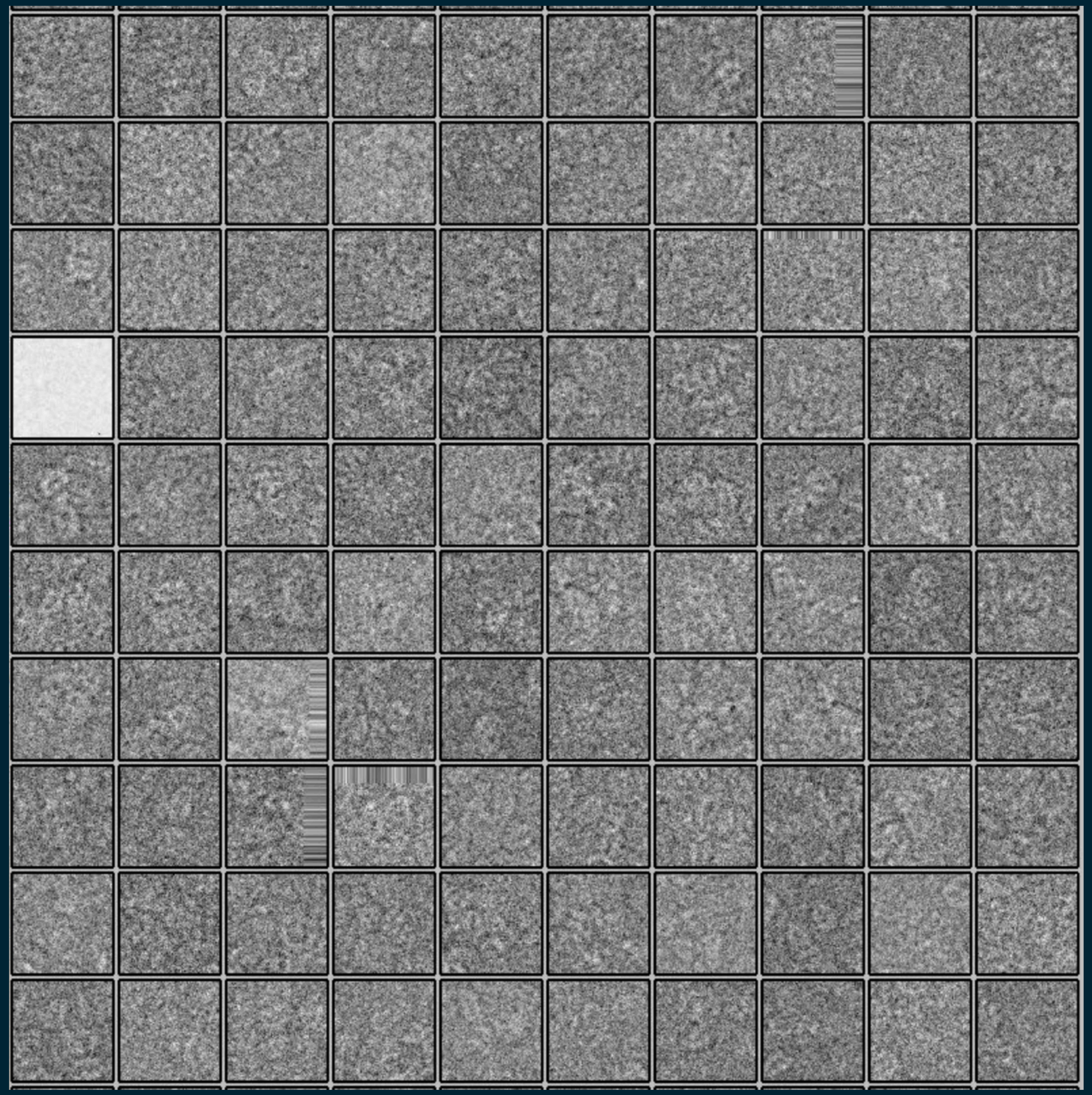
X_j —the j^{th} pixel (out of J pixels total)

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

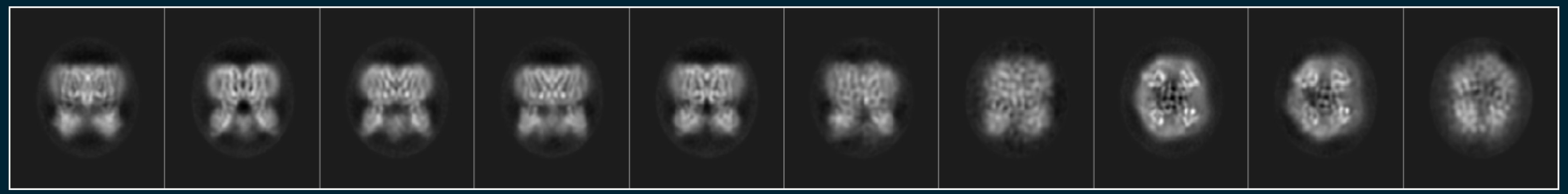
X_i

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 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

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

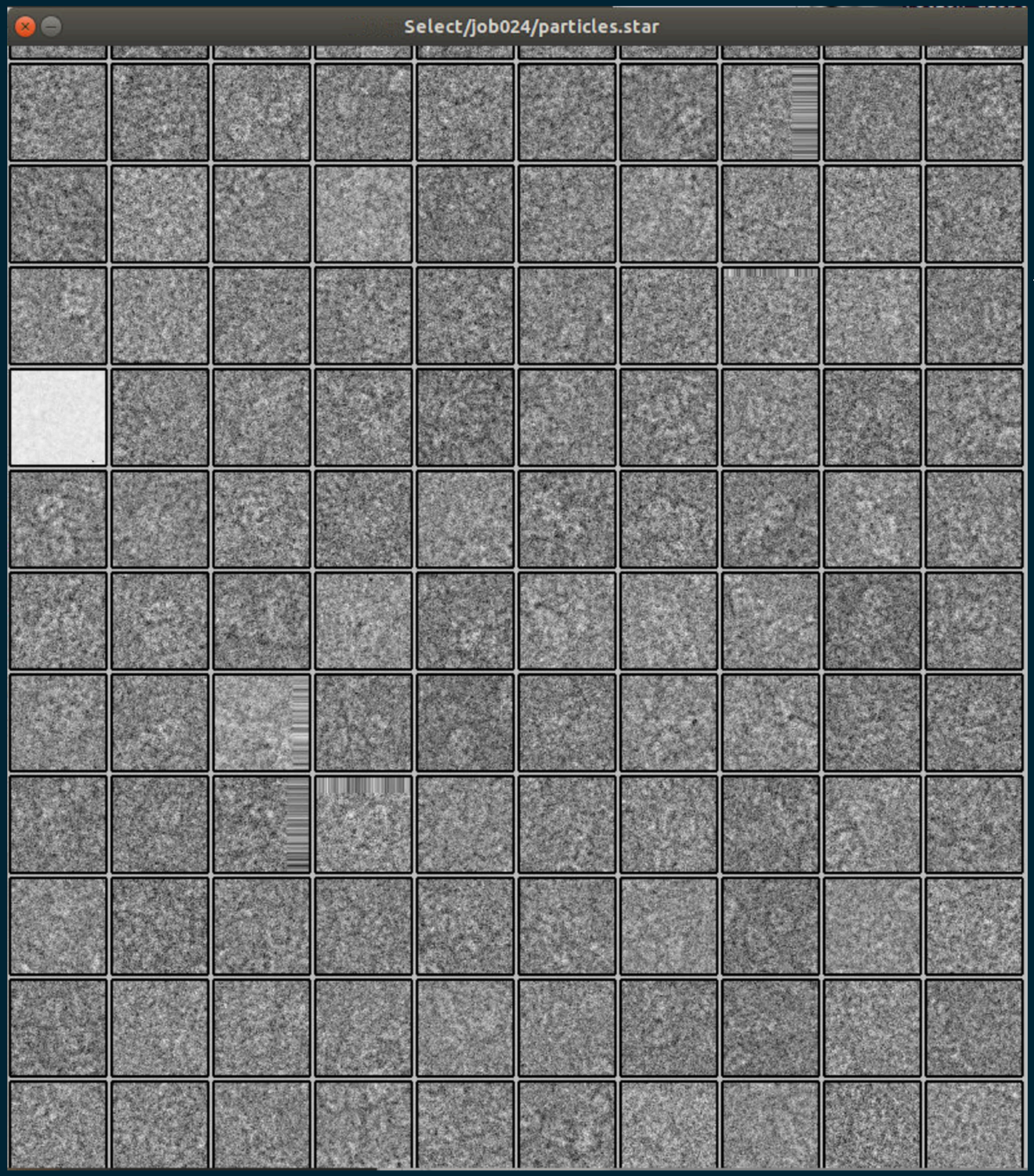
where $\{i'\}$ is the set of i values for which $k_i = k$, and $\mathbf{P}_{\phi_i}^{\mathbf{T}}$ is the inverse of the transformation $\mathbf{P}_{\phi_i}$.

Single-particle reconstruction

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

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

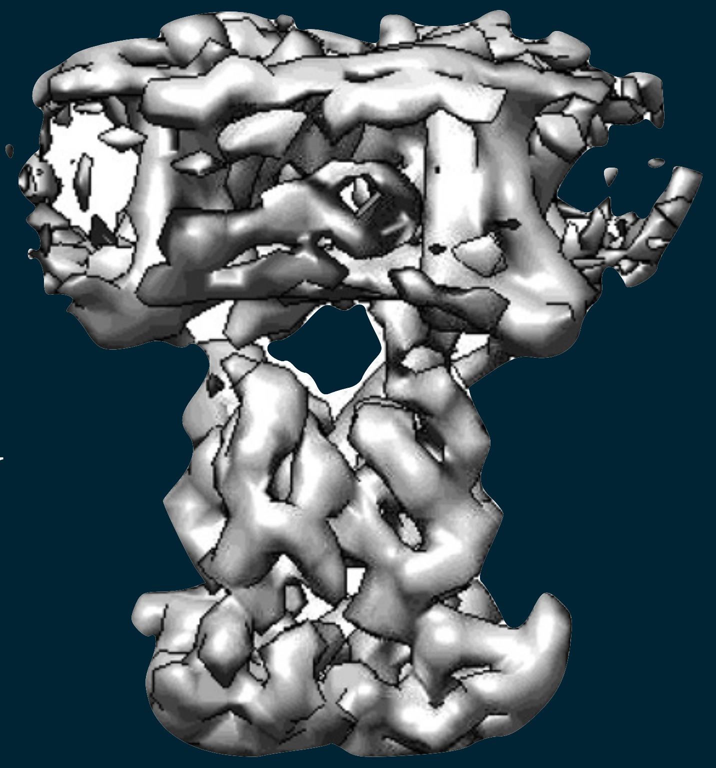
The goal is to discover the volume V



X_i

Project along ϕ_i

V



The first step is to compare images to determine orientations…

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

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

Probabilities, another way to compare images

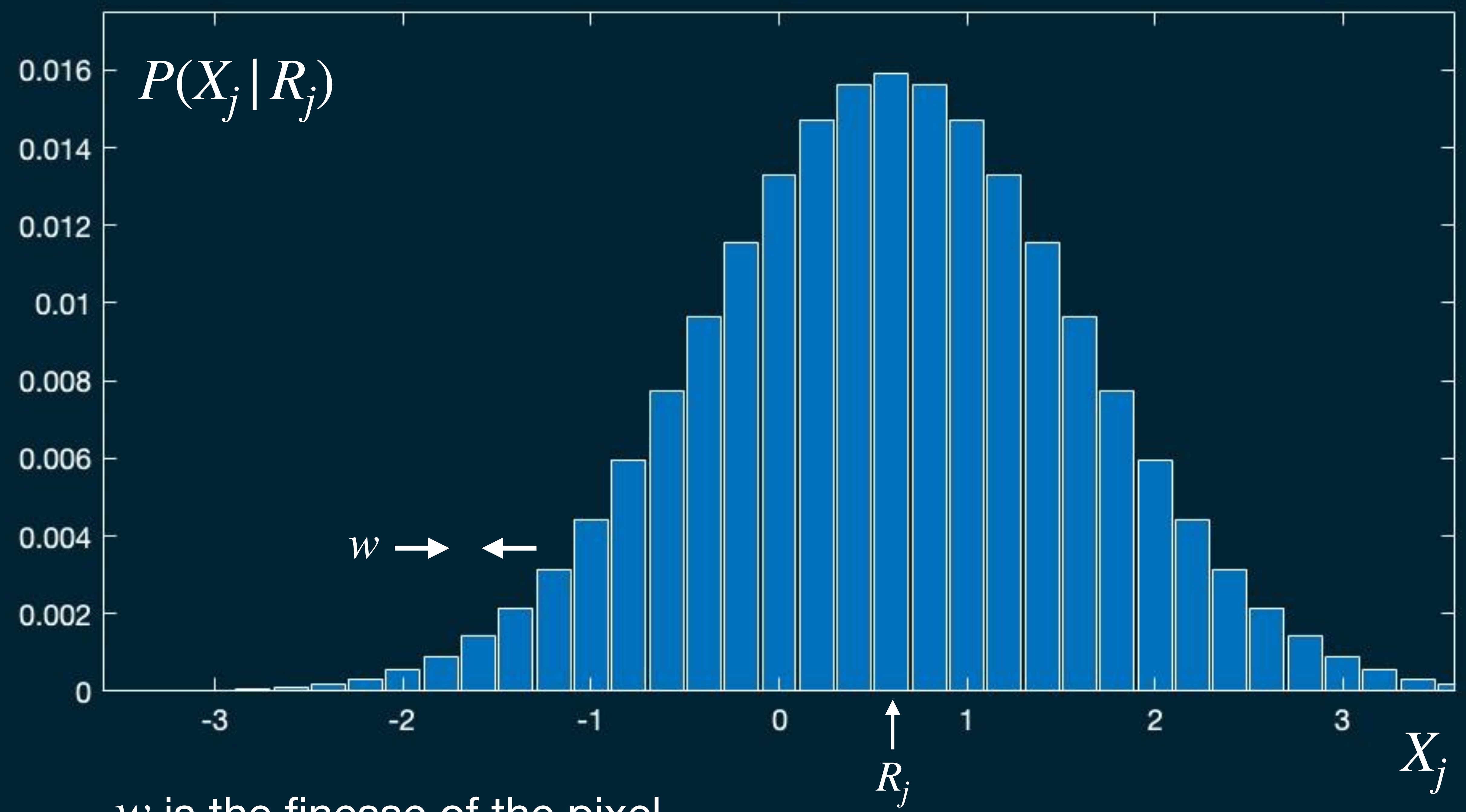
Image model: $X = R + N$

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:

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



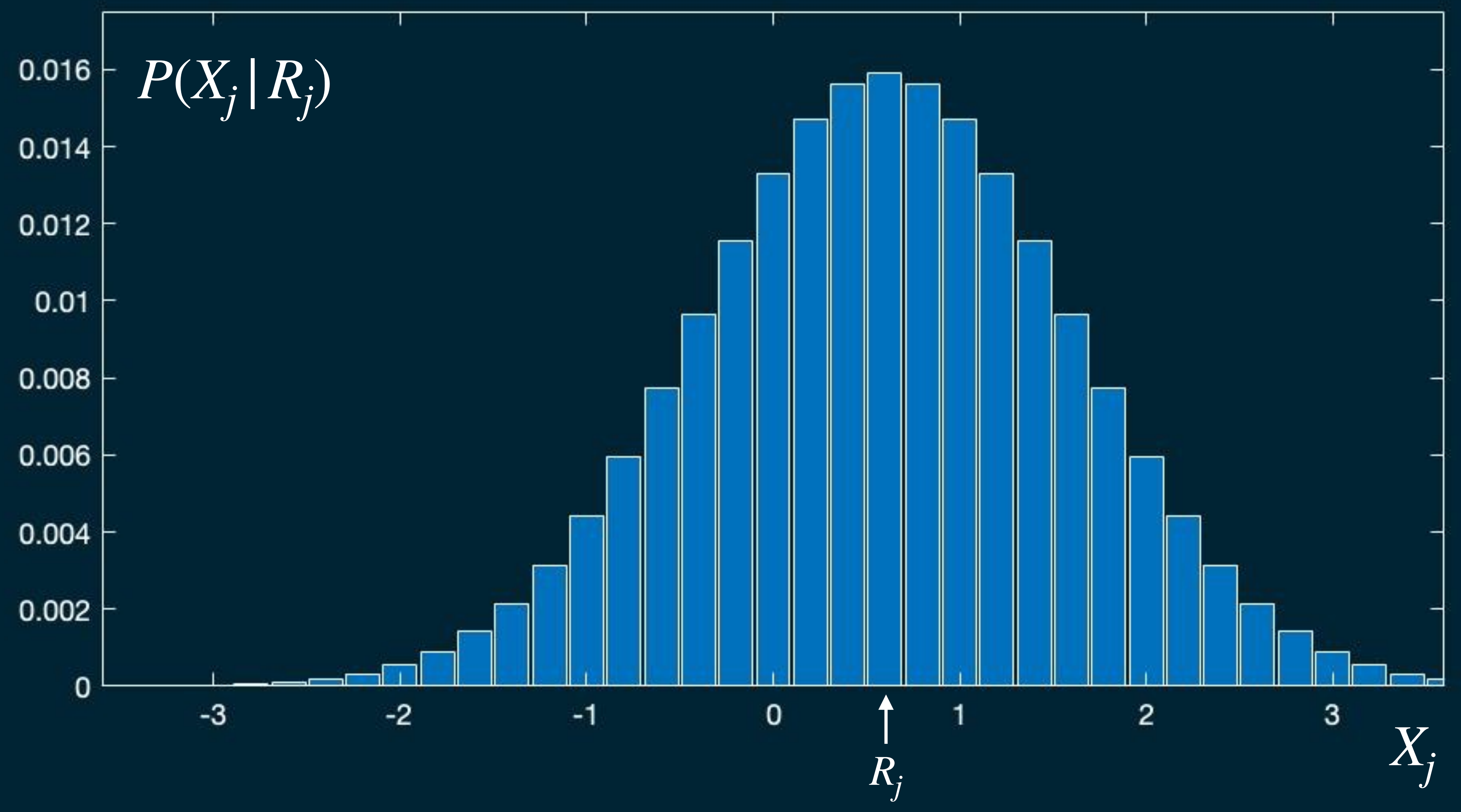
w is the finesse of the pixel intensity measurements. We'll ignore it (set it to 1).

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 and ignore it.)

The Likelihood

Let $\mathbf{X} = \{X_1 \,..\, X_N\}$ 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 $\rightarrow$ **Experiment** $\rightarrow$ ***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

We know that

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

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

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

assuming $P(\phi)$ is uniform.

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.

2D classification by maximum likelihood

Given estimates of $A_1 \ldots A_K$ 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 $\mathbf{P}_{\phi_i}$.

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:

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

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, based on the current estimate $V^{(n)}$ of the volume.

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:

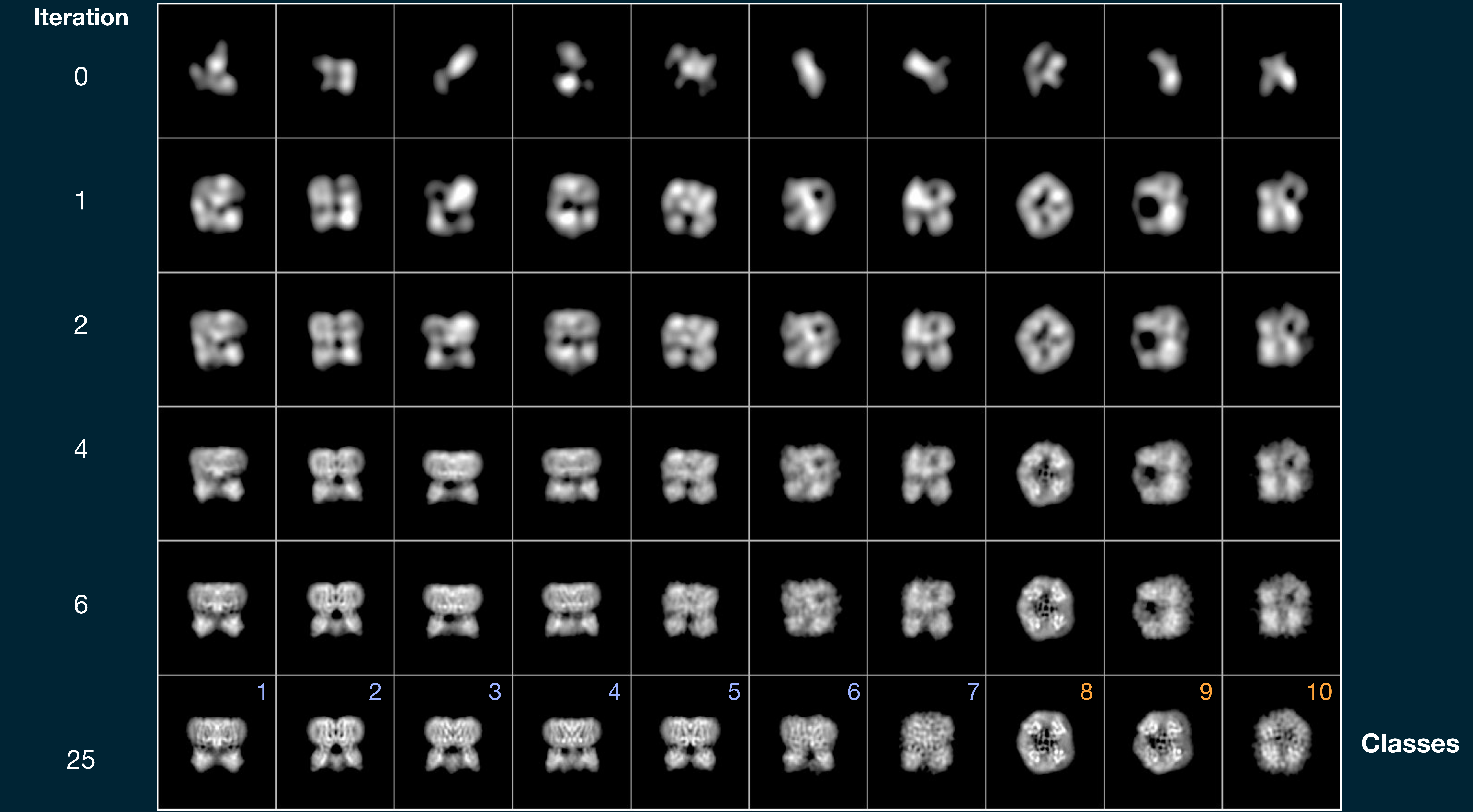
1. Find the best orientation ϕ_i for each particle image X_i
2. Update the volume according to

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

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

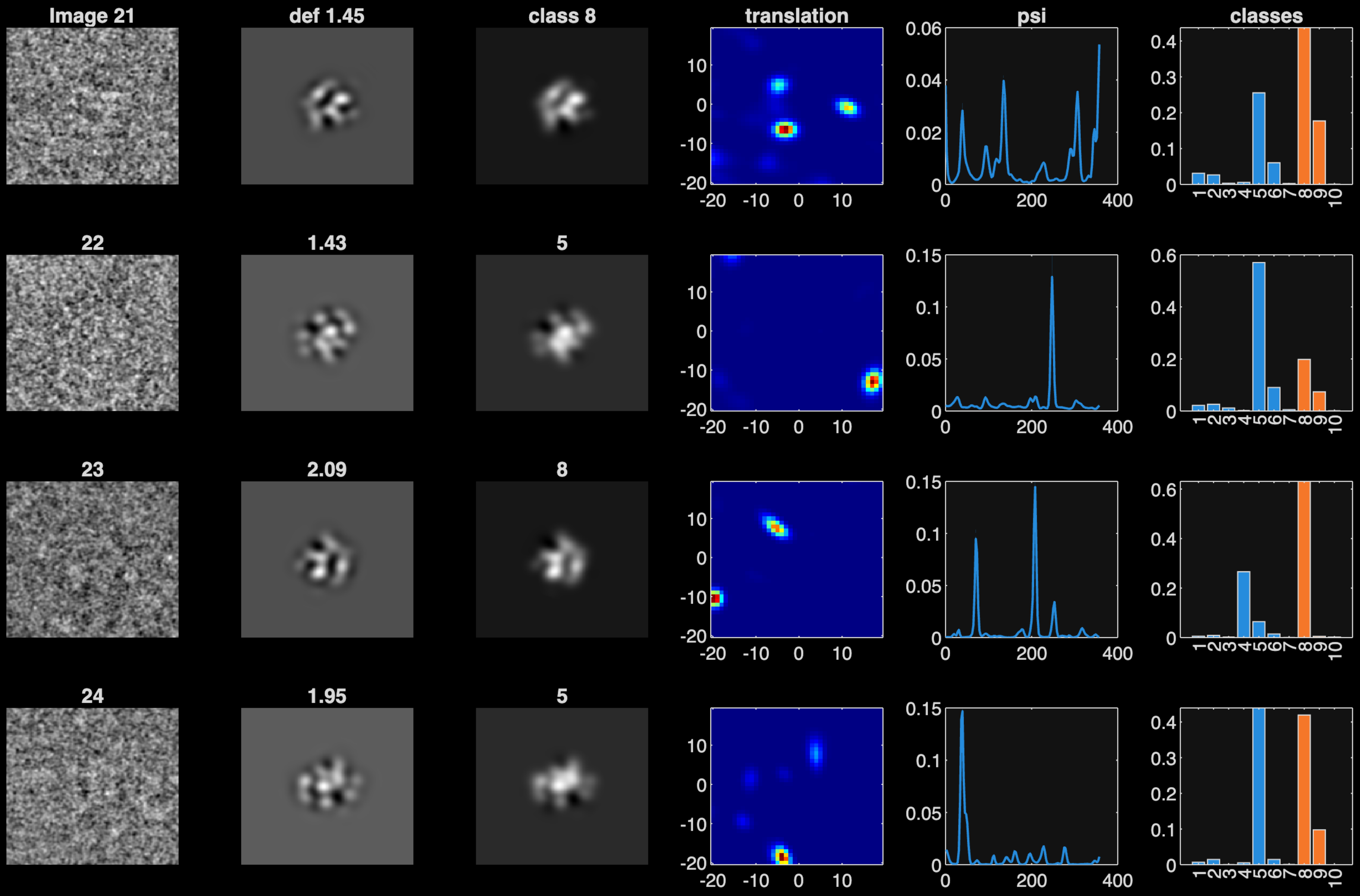
Quantity	Meaning in 3D classification	Meaning in 2D classification
V_k	Class volume	Class average image
ϕ	3 Euler angles of orientation + 2 translations	1 angle of rotation + 2 translations
$\mathbf{P}_\phi$	Projection operator 3D $\longrightarrow$ 2D	Image rotation and shift
$\mathbf{P}_\phi^\mathbf{T}$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

2D classification by maximum likelihood

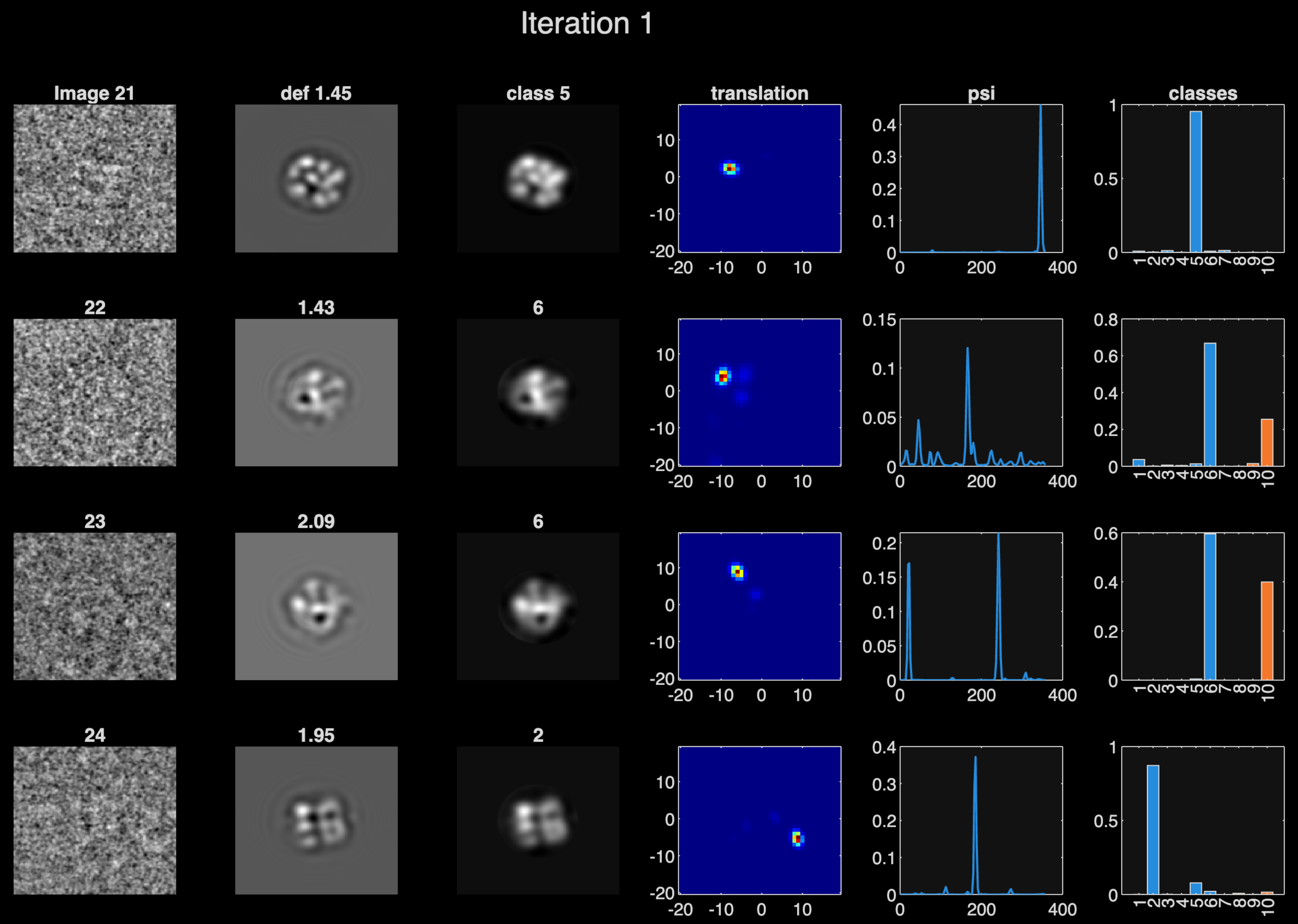


2D classification by maximum likelihood

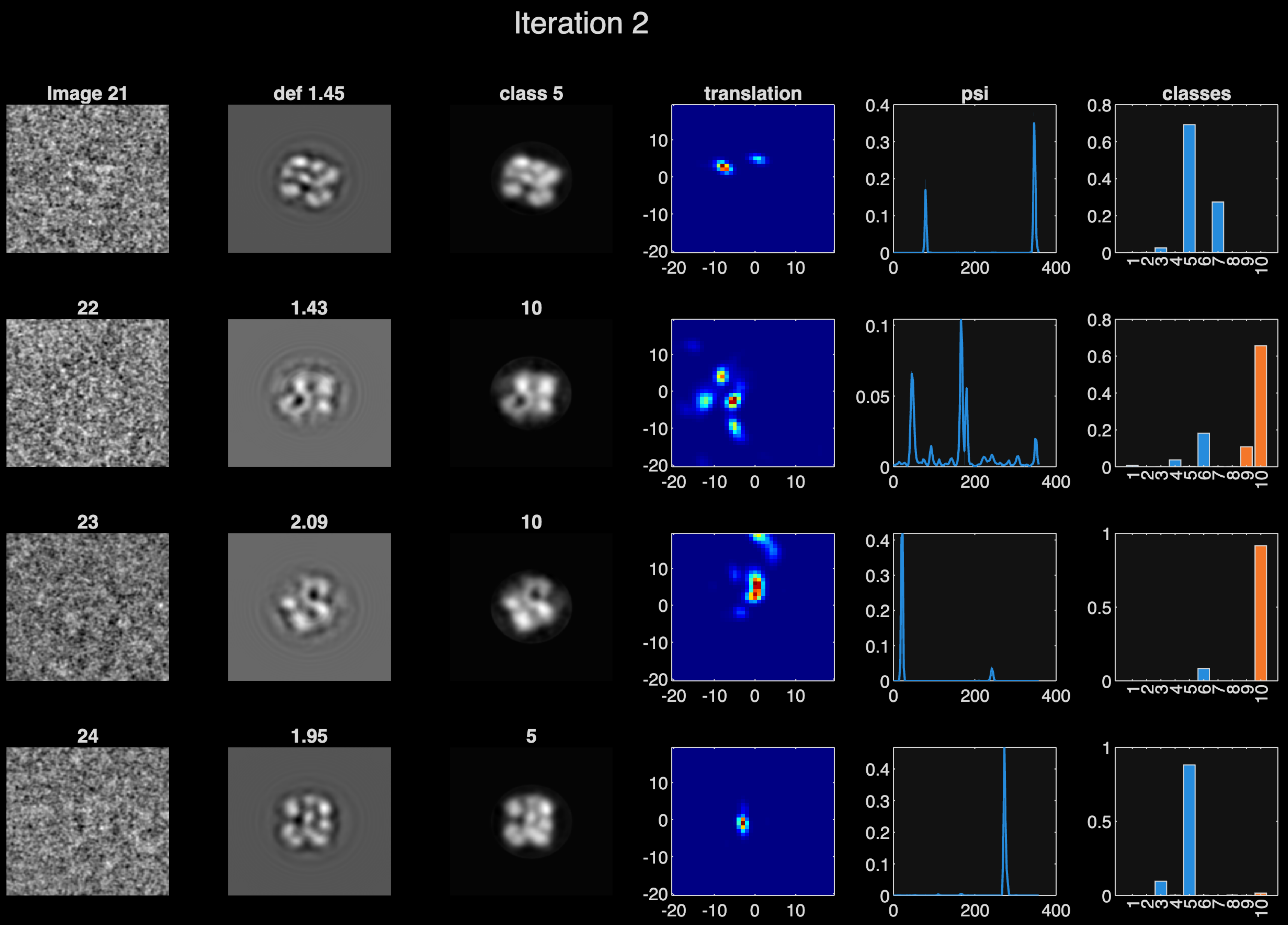
Iteration 0



2D classification by maximum likelihood

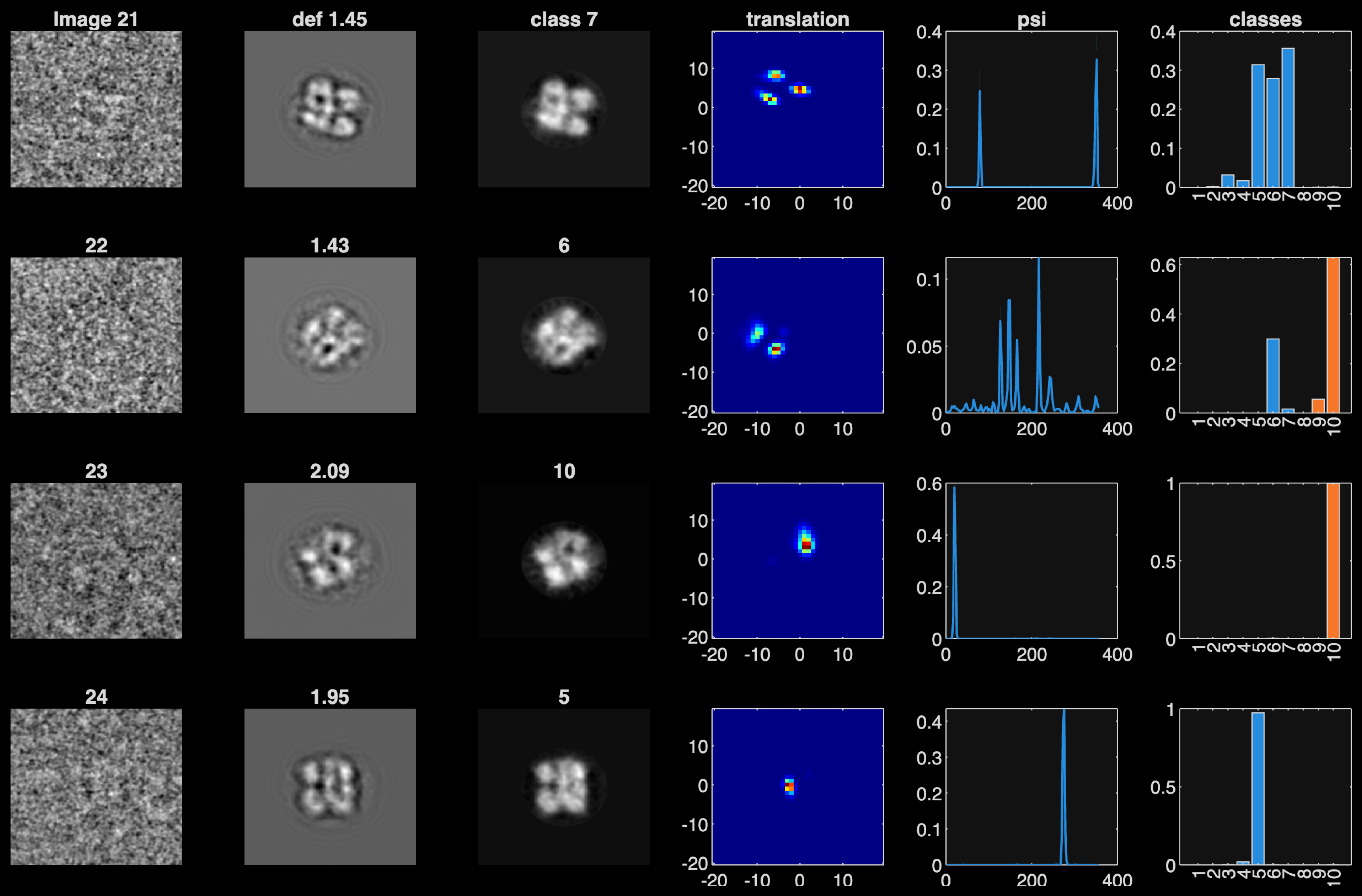


2D classification by maximum likelihood



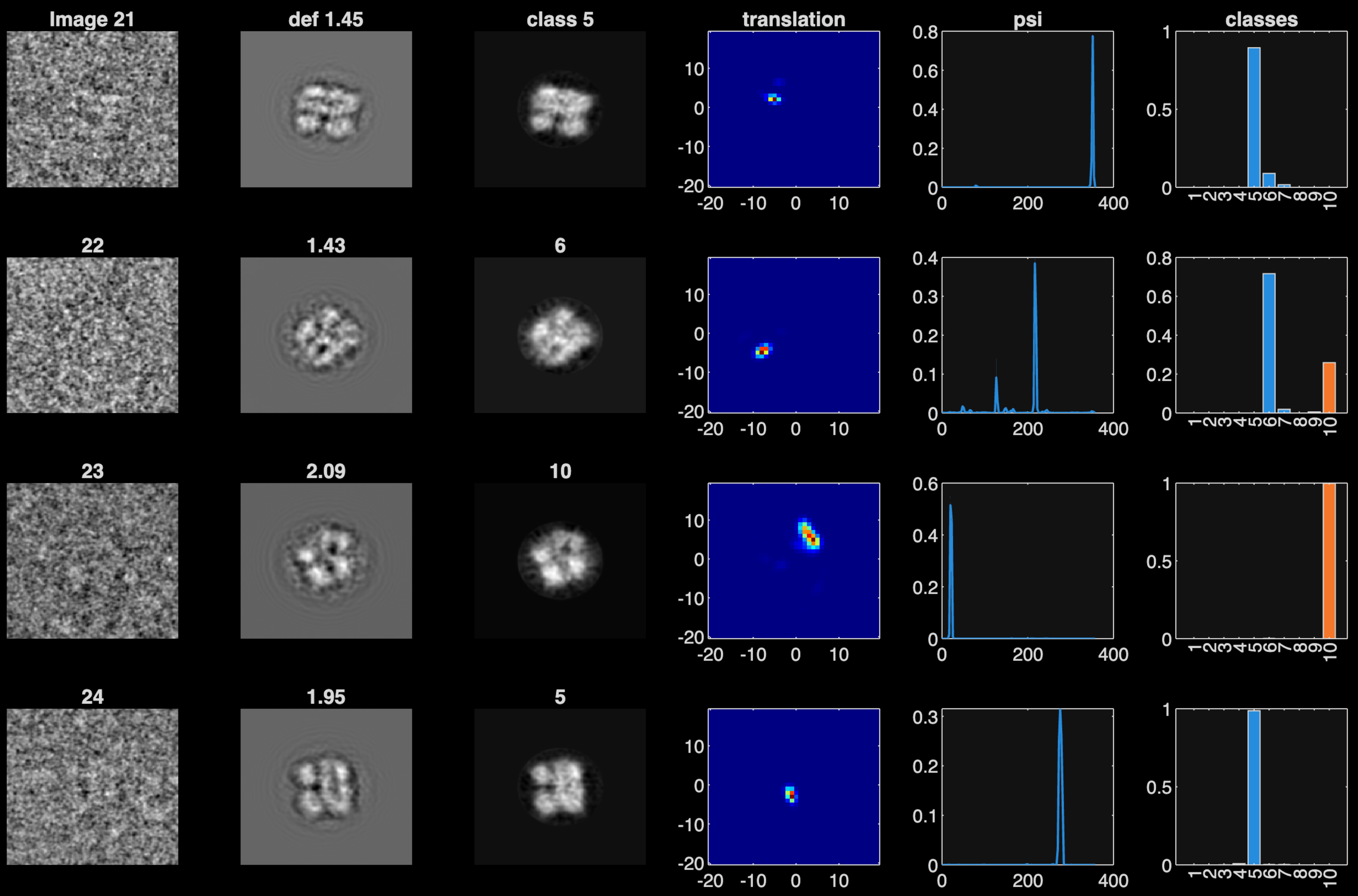
2D classification by maximum likelihood

Iteration 4



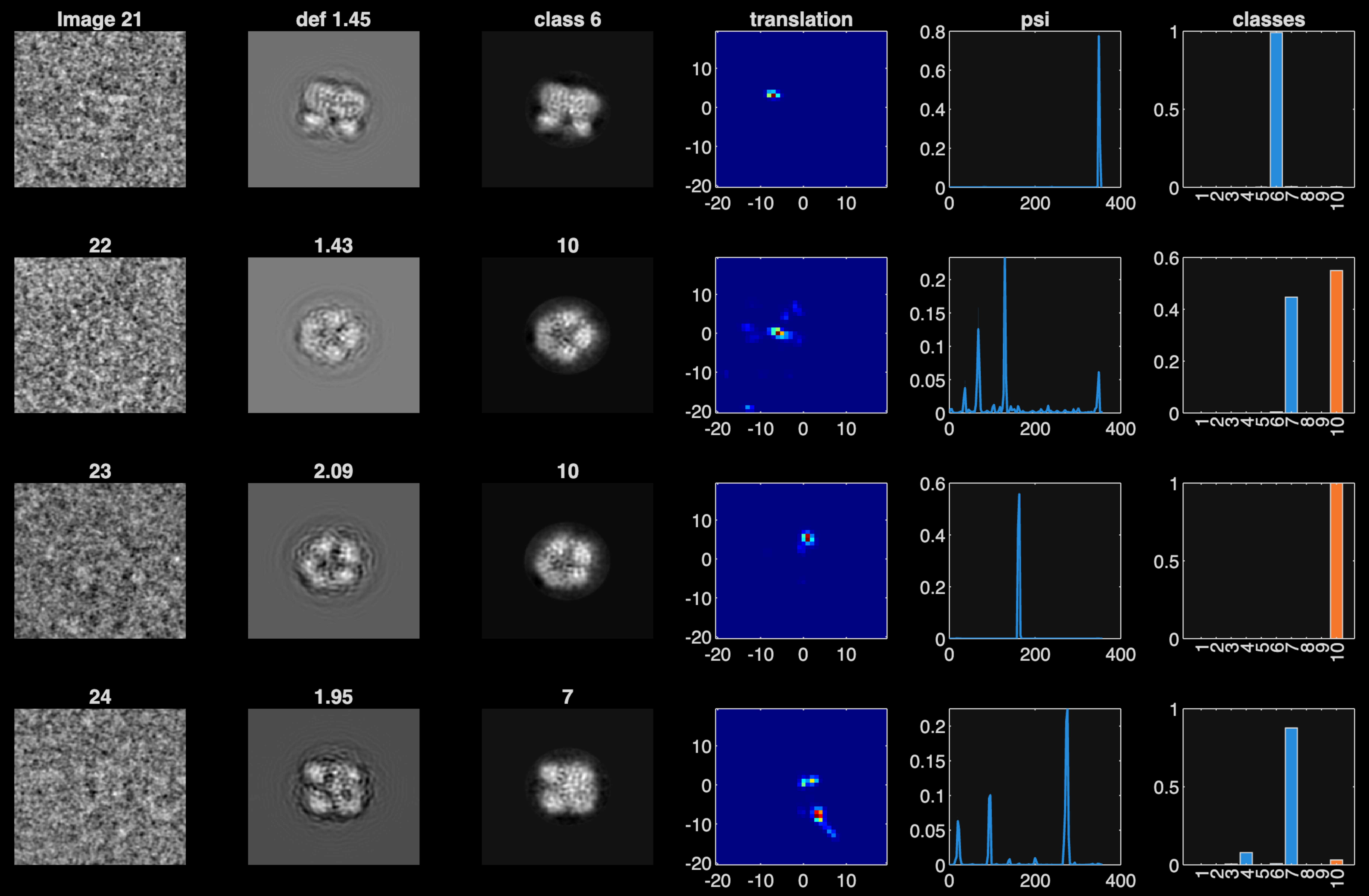
2D classification by maximum likelihood

Iteration 6

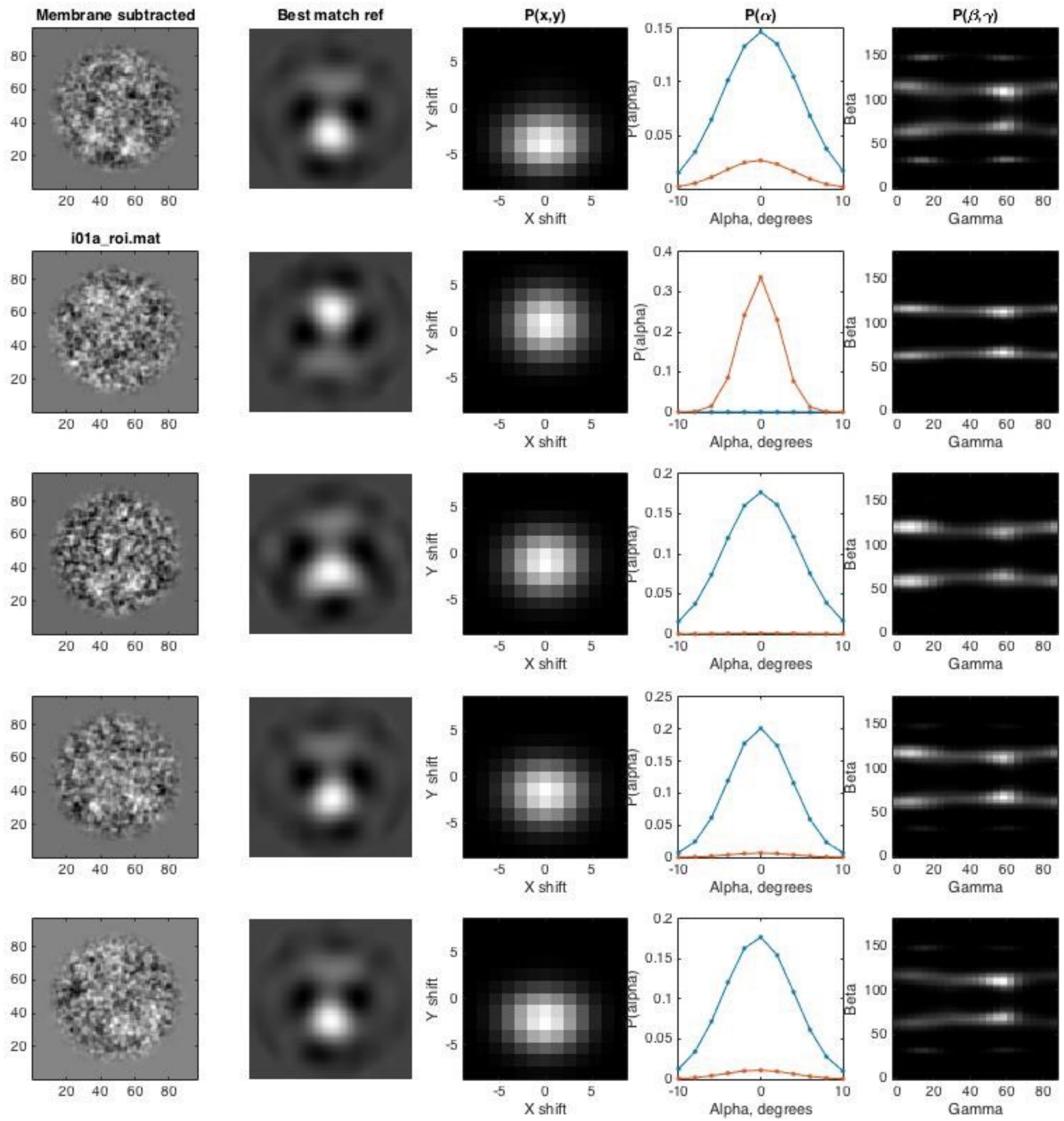


2D classification by maximum likelihood

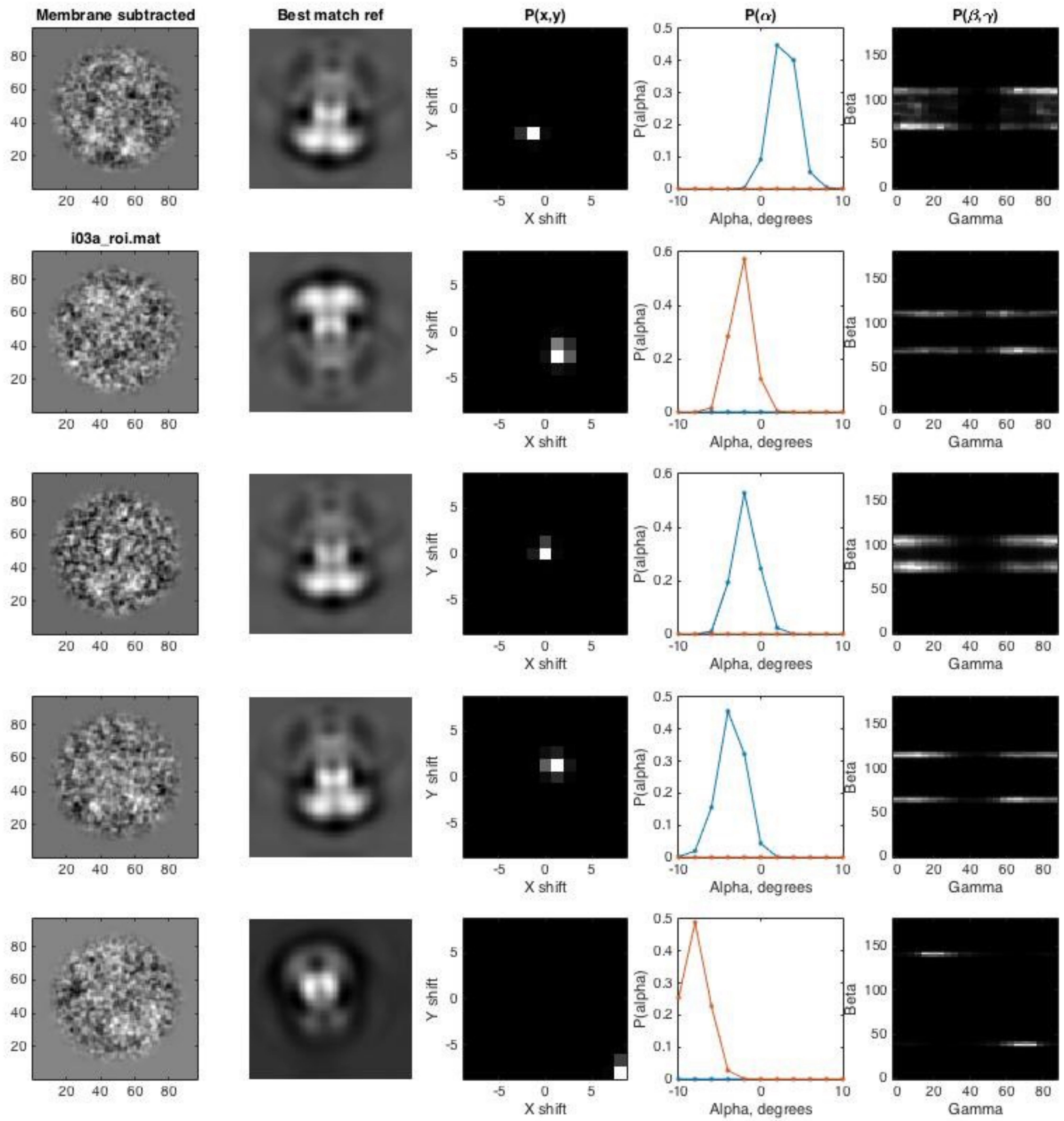
Iteration 25



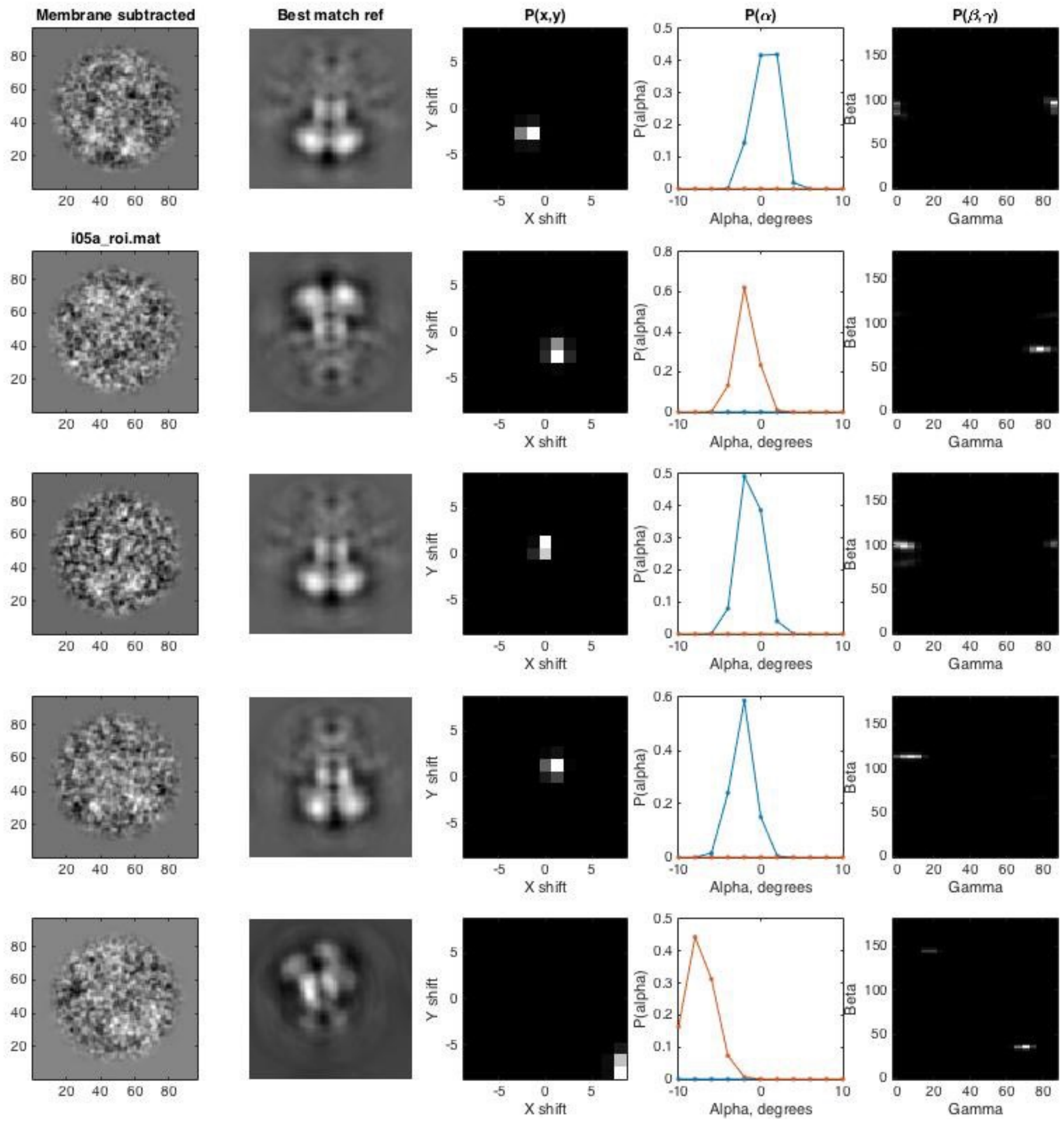
3D Reconstruction: on the first EM iteration, angle assignments are not sharp



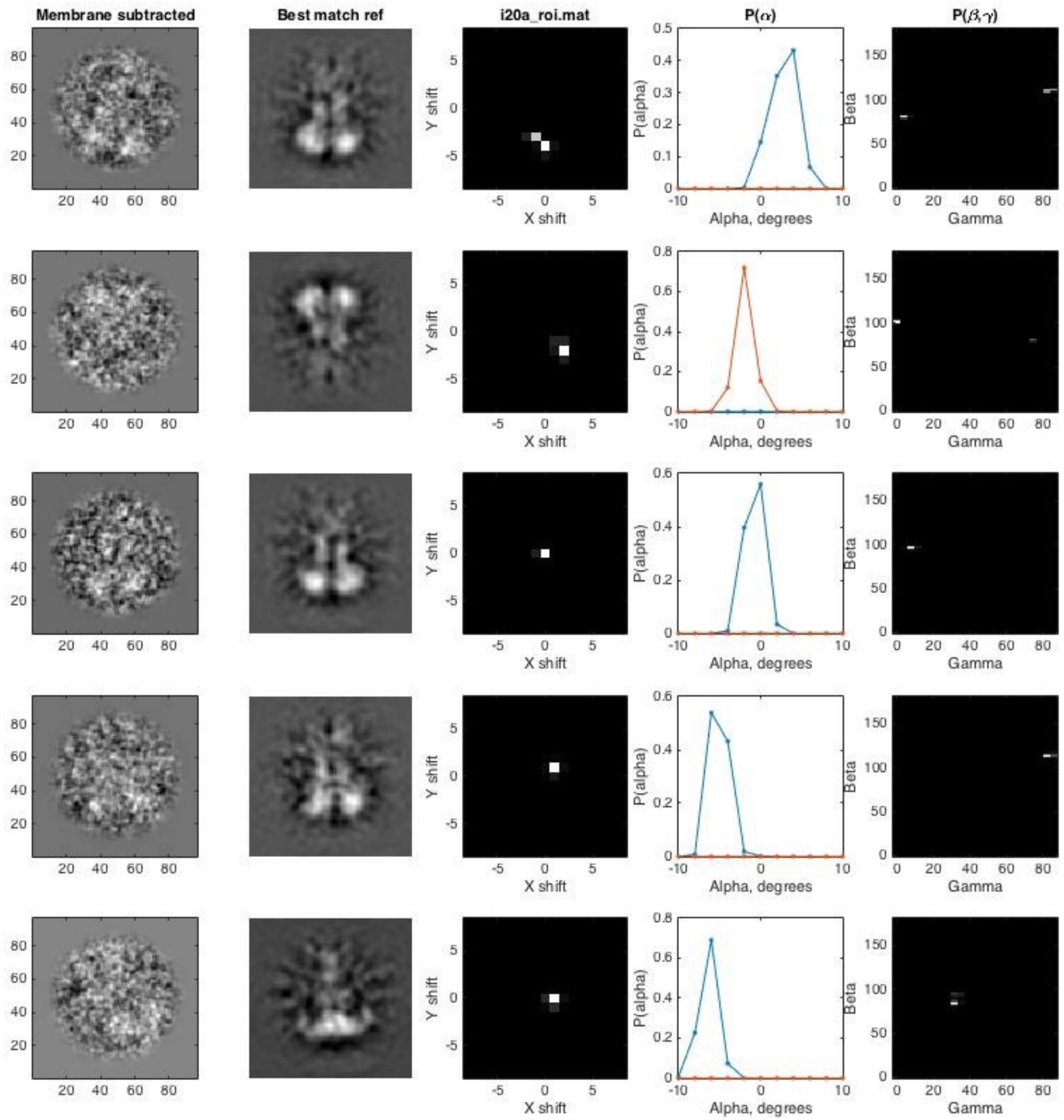
Iteration 3


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 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



The orientation determination is the most expensive step

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

The orientation determination is the most expensive step

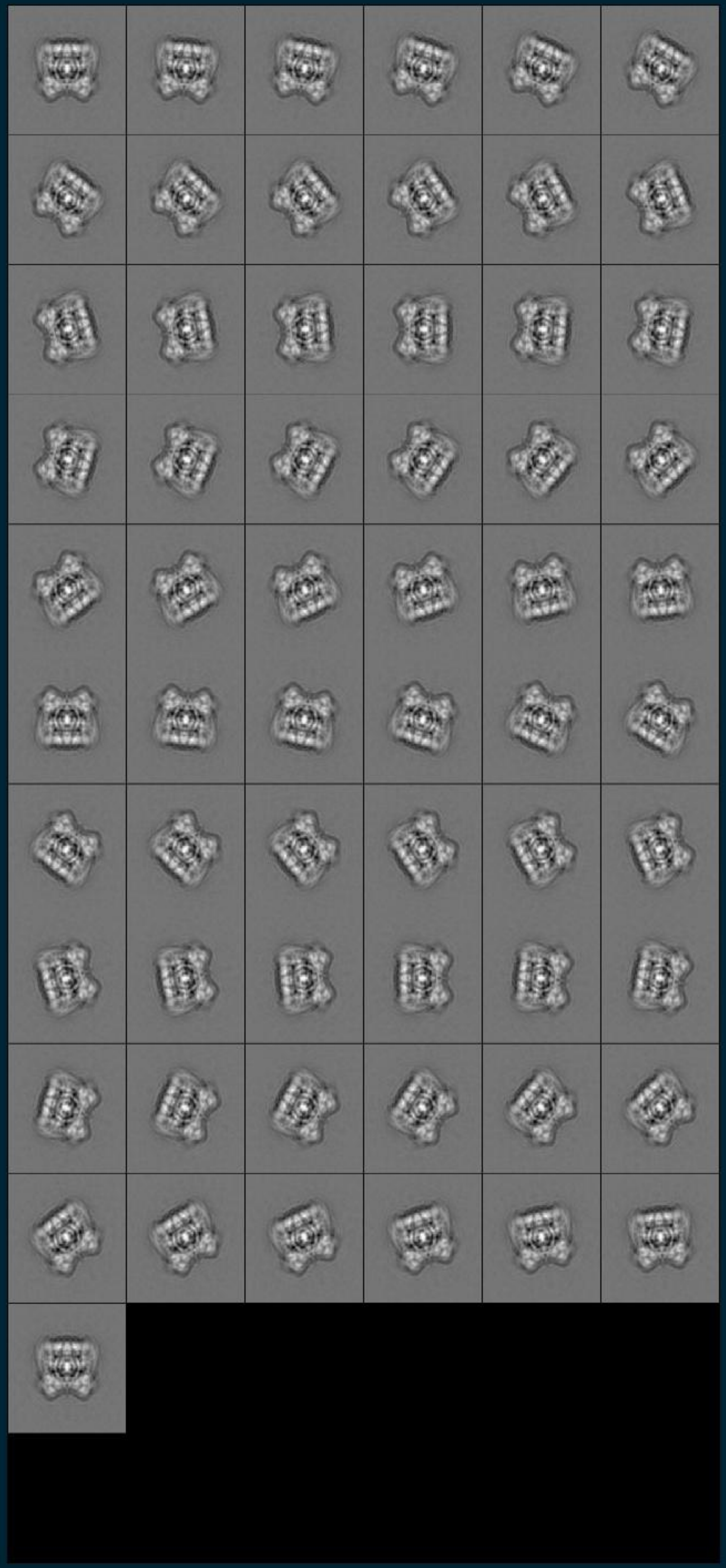
$$\text{No. operations} \approx \underbrace{\frac{\pi^3}{8} t^2 n^5 N}_{\text{finding orientations}} + \underbrace{\pi n^4 + N n^2}_{\text{3D reconstruction}}$$

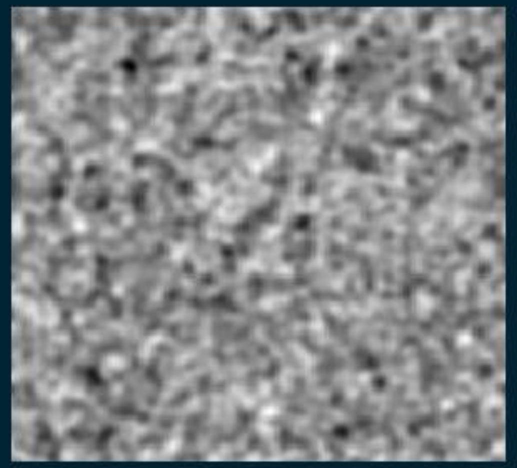
e.g. $N=10^5$, $n=128$, $t=7$

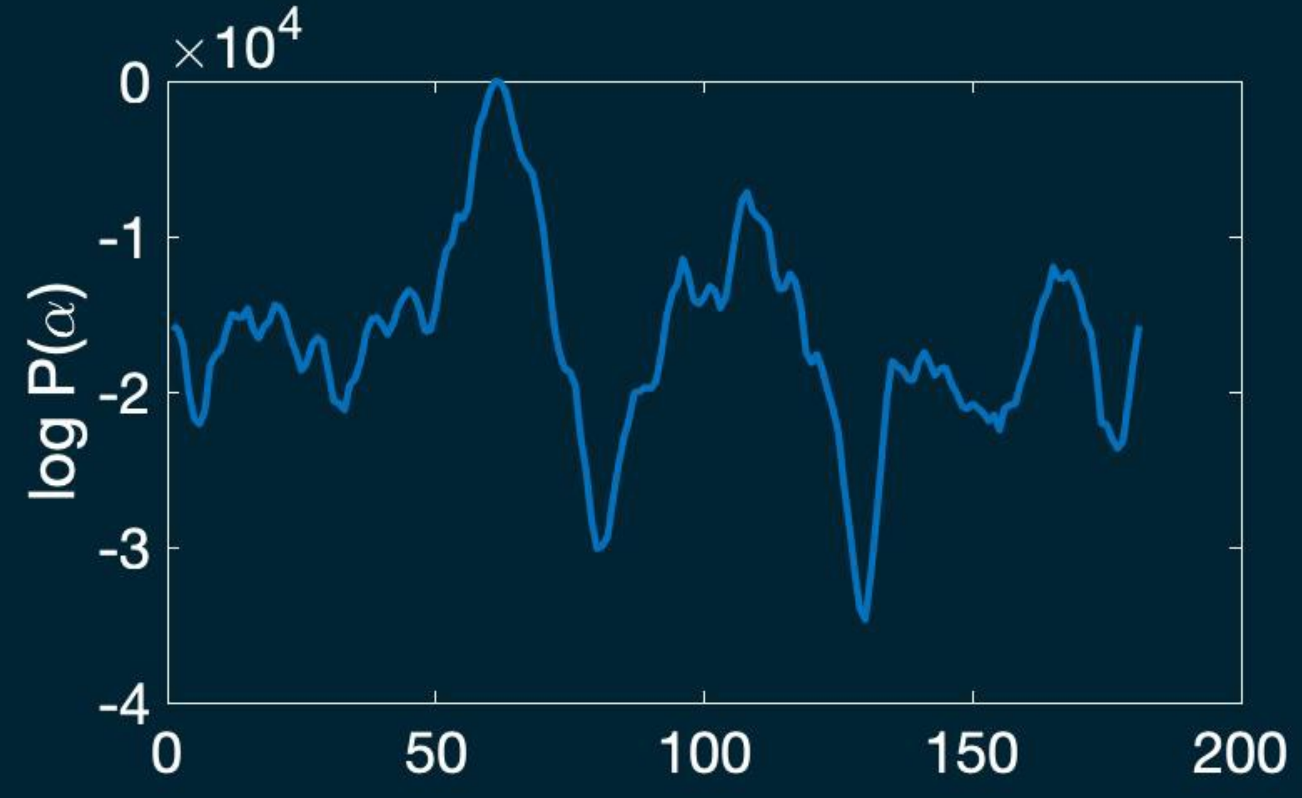
No. operations ≈ 6 x 10^{17} ≈ 19 CPU-years

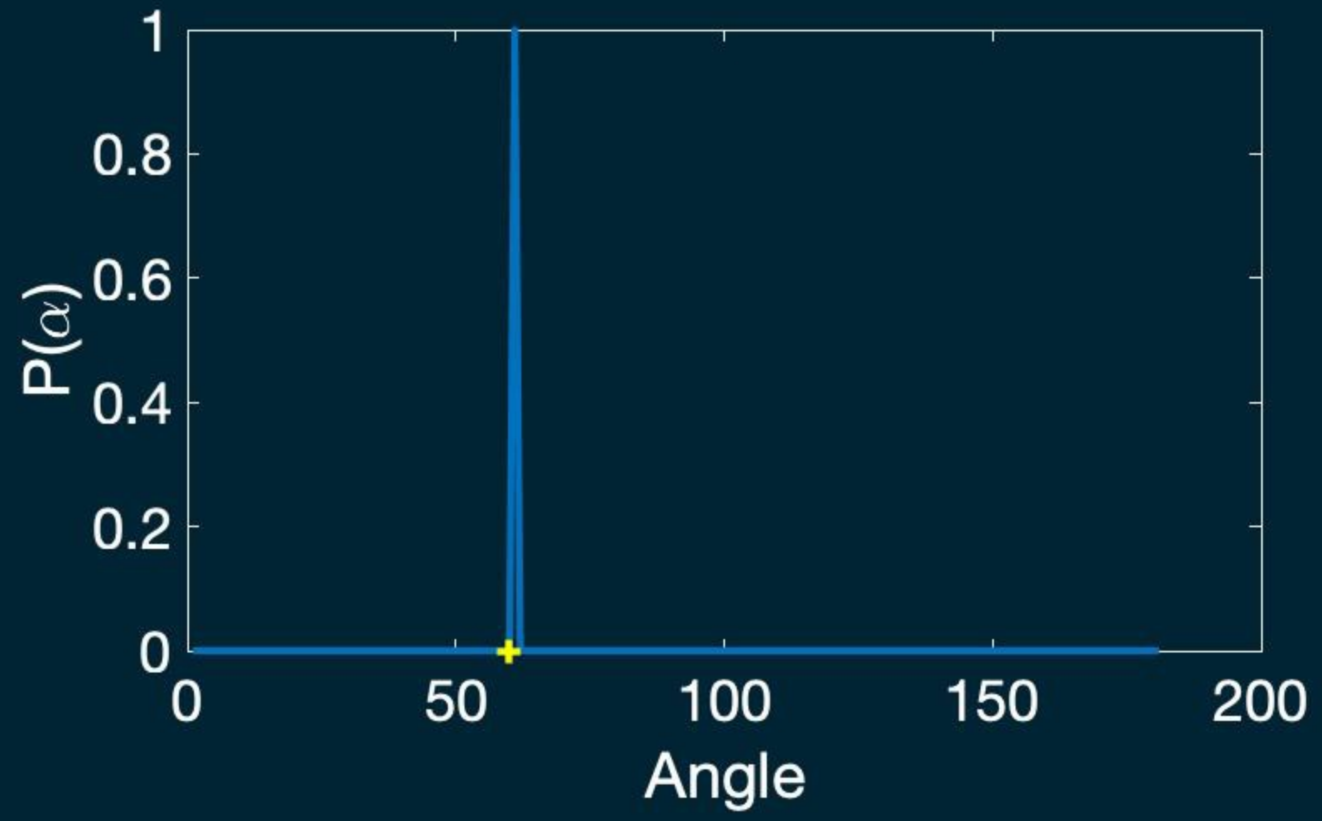
With efficient programs, ~ 1 CPU-day

Evaluating Γ_ϕ is expensive: one of 5 parameters

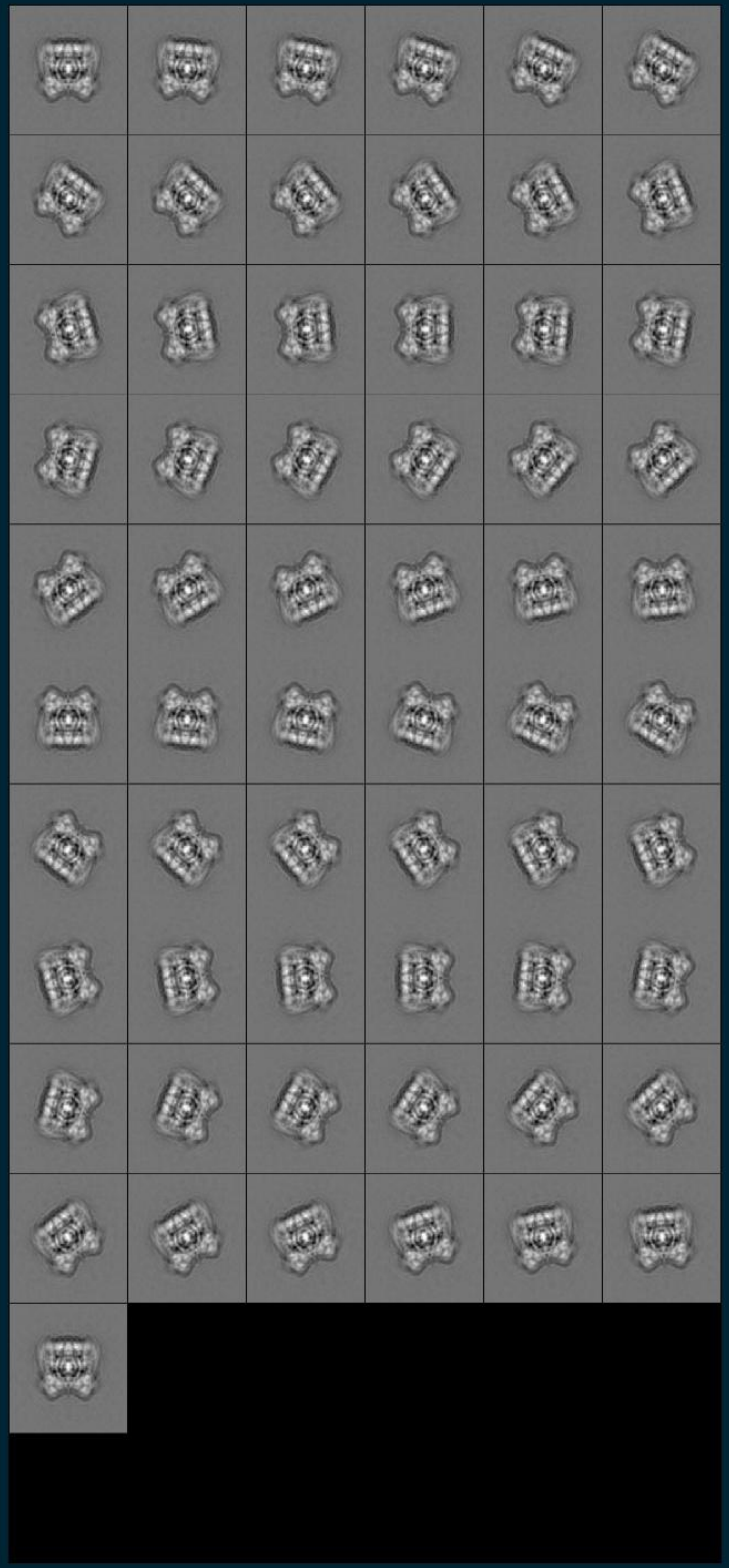


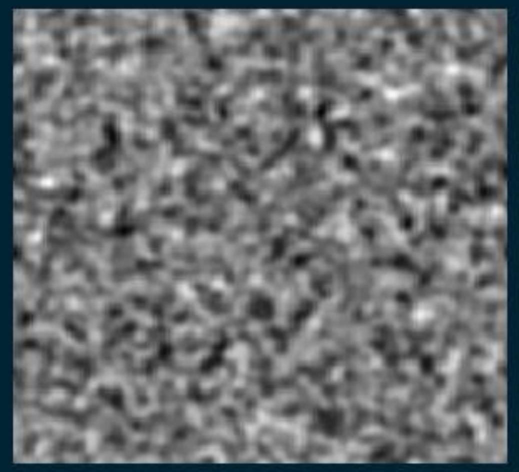


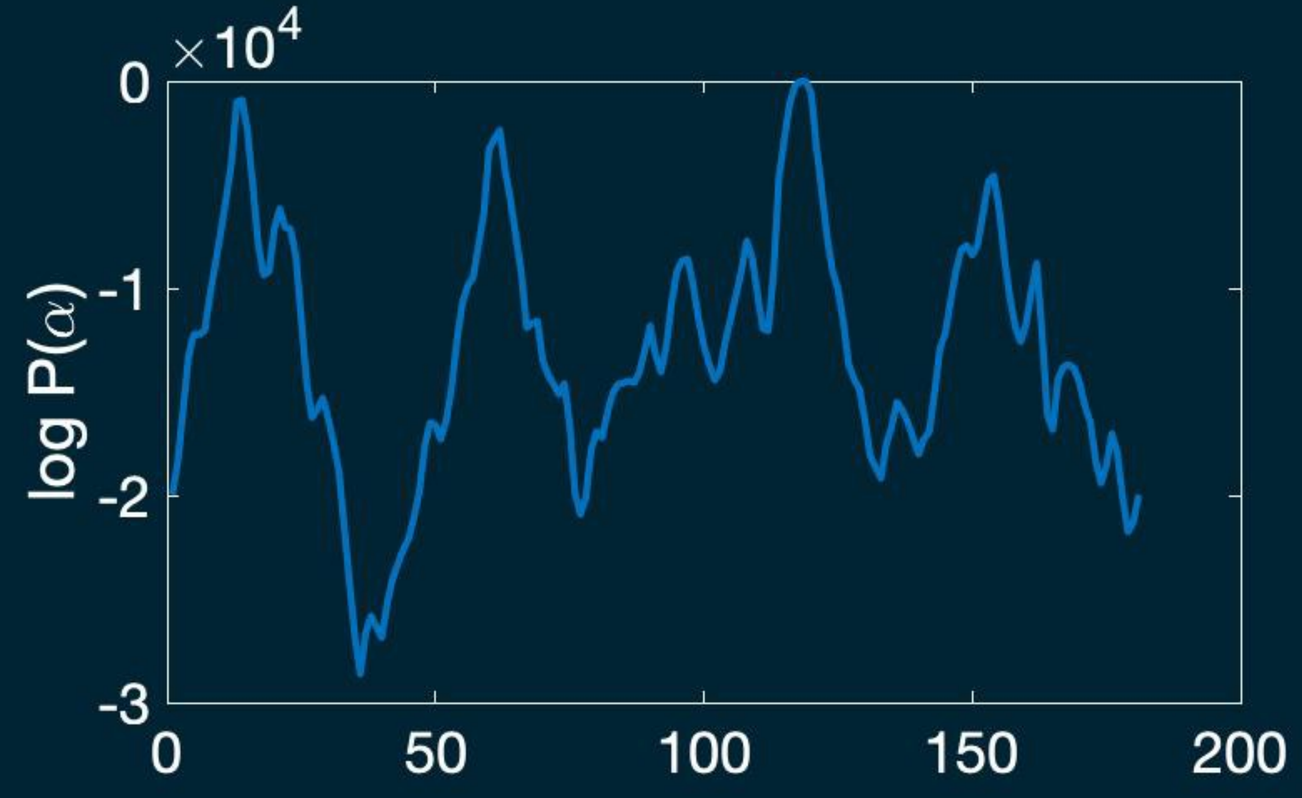


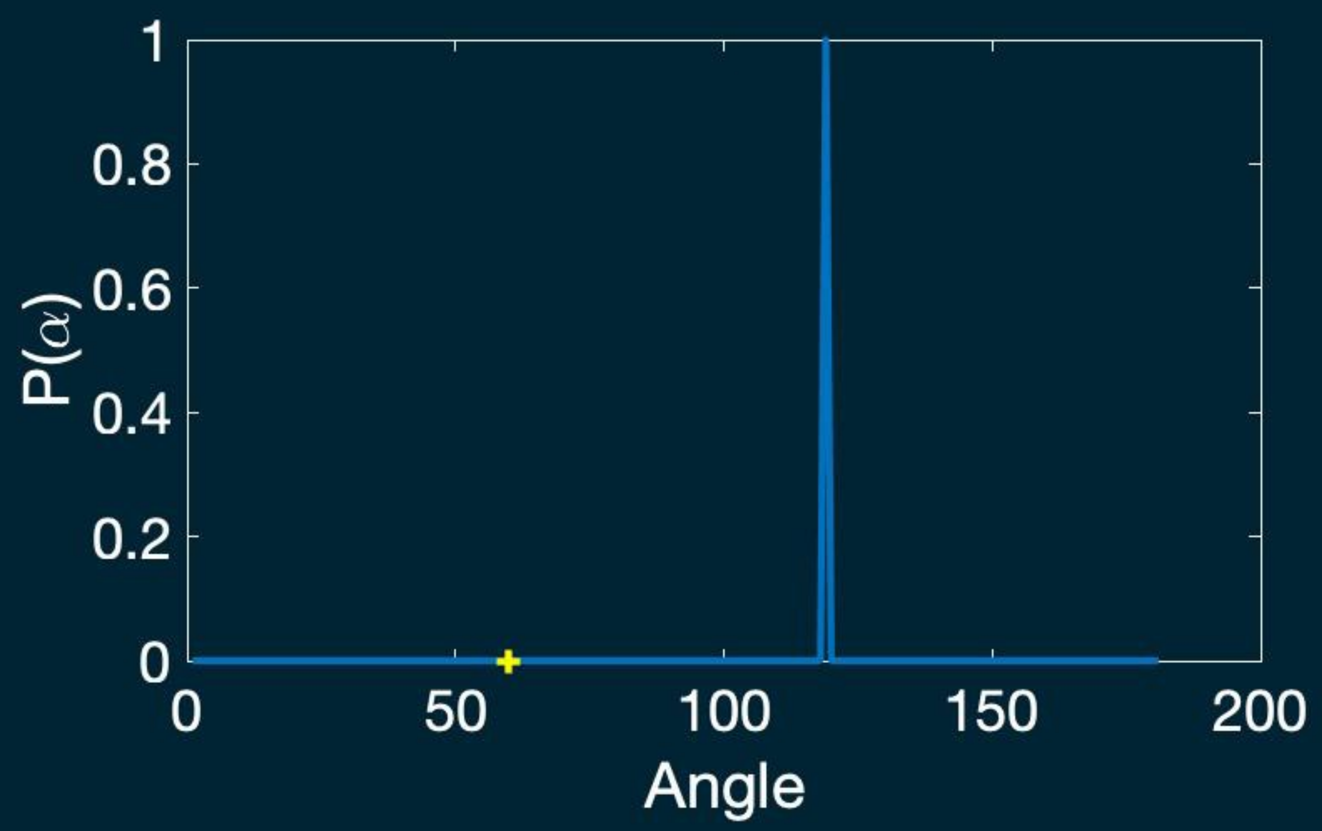


Evaluating Γ_ϕ is expensive: one of 5 parameters

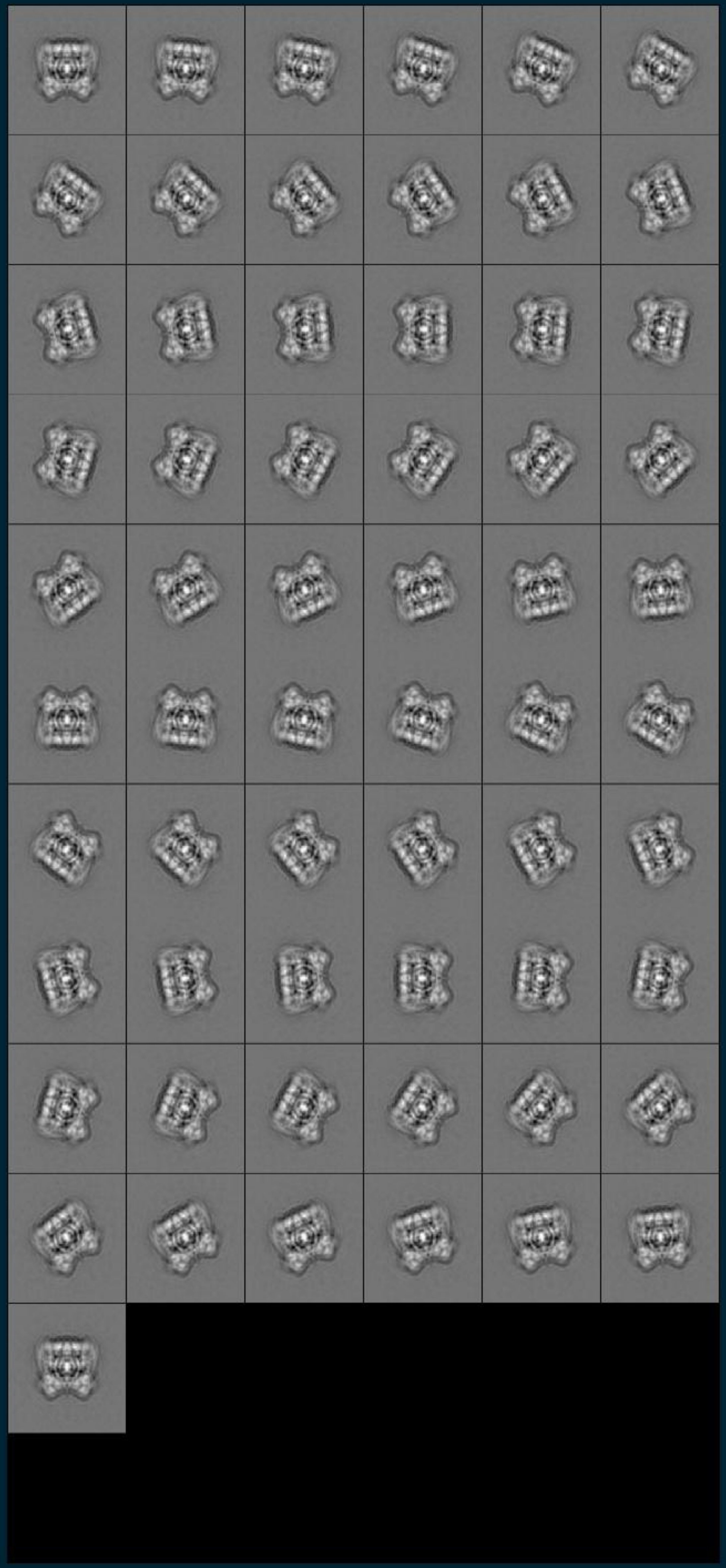


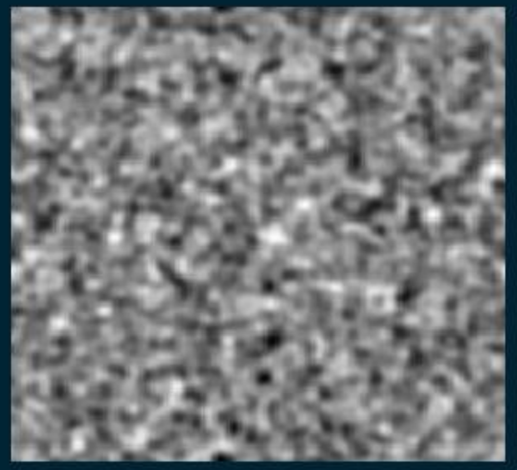


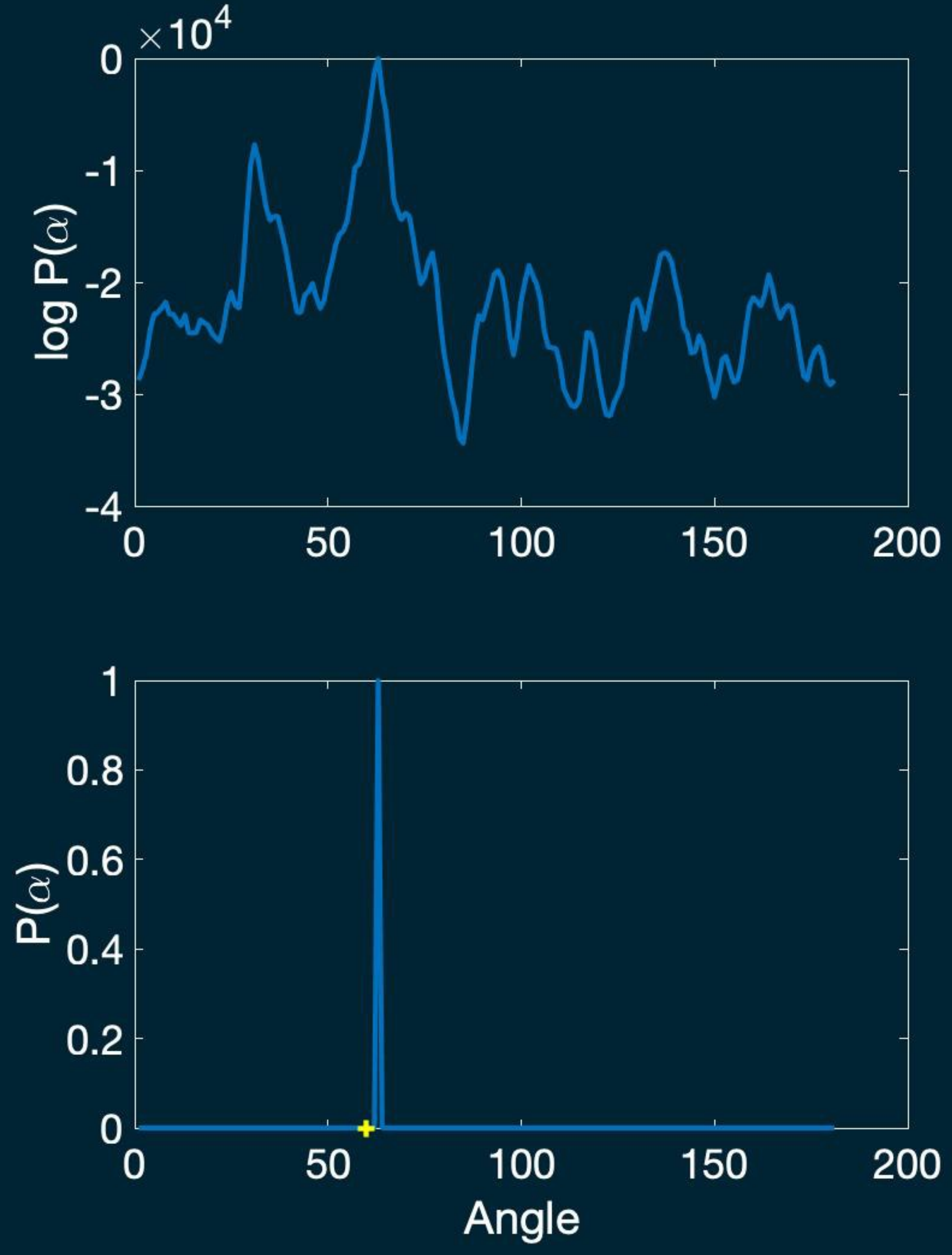




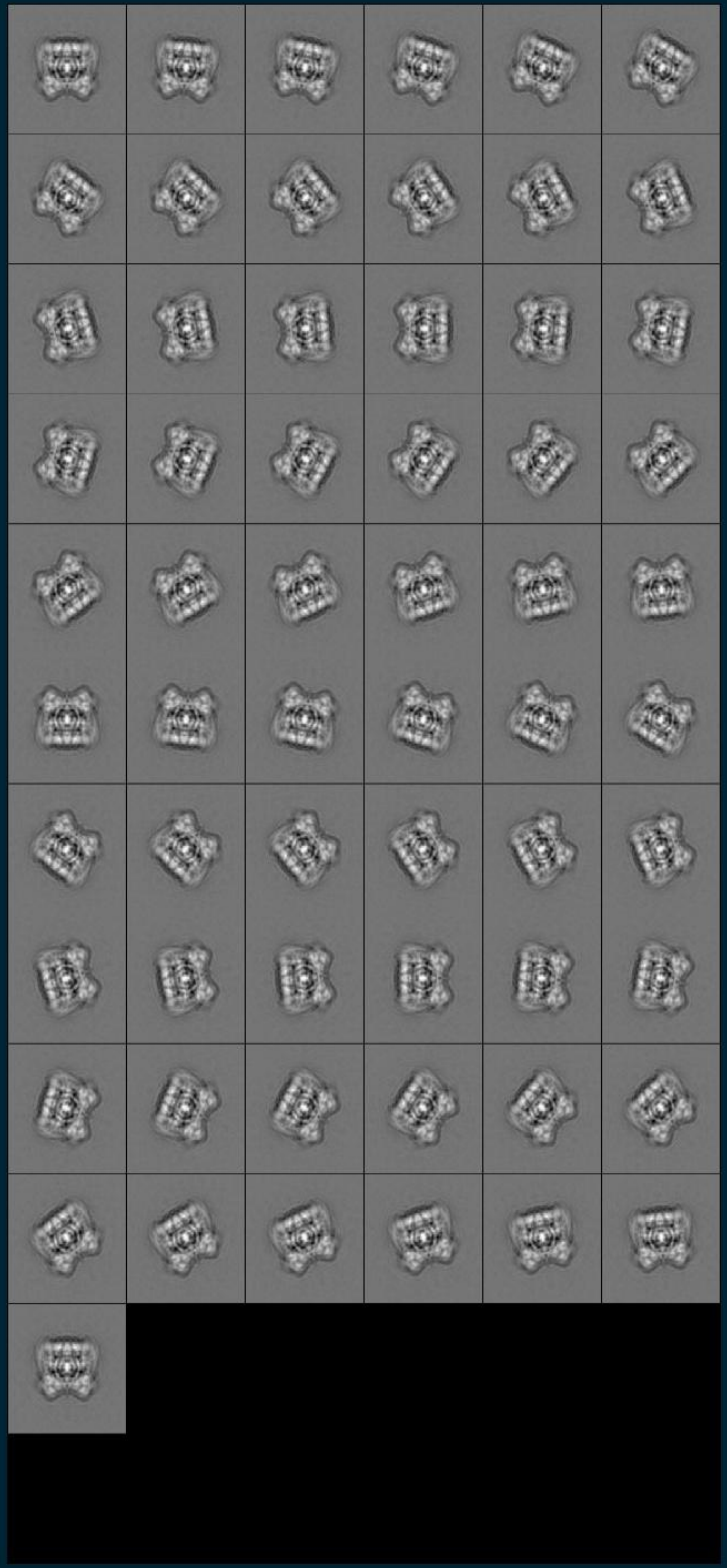
Evaluating Γ_ϕ is expensive: one of 5 parameters

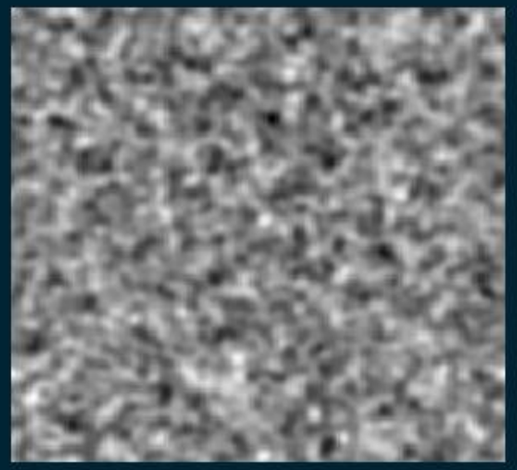


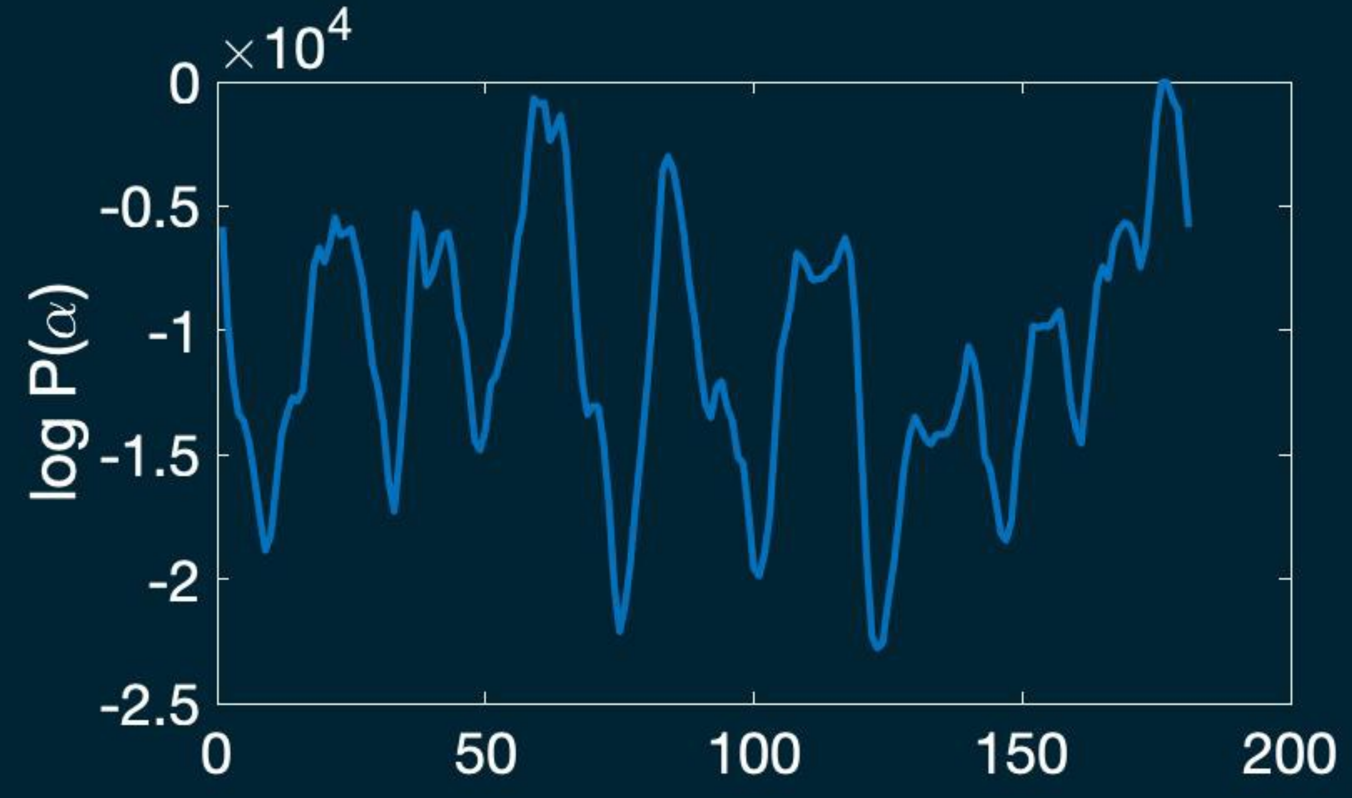


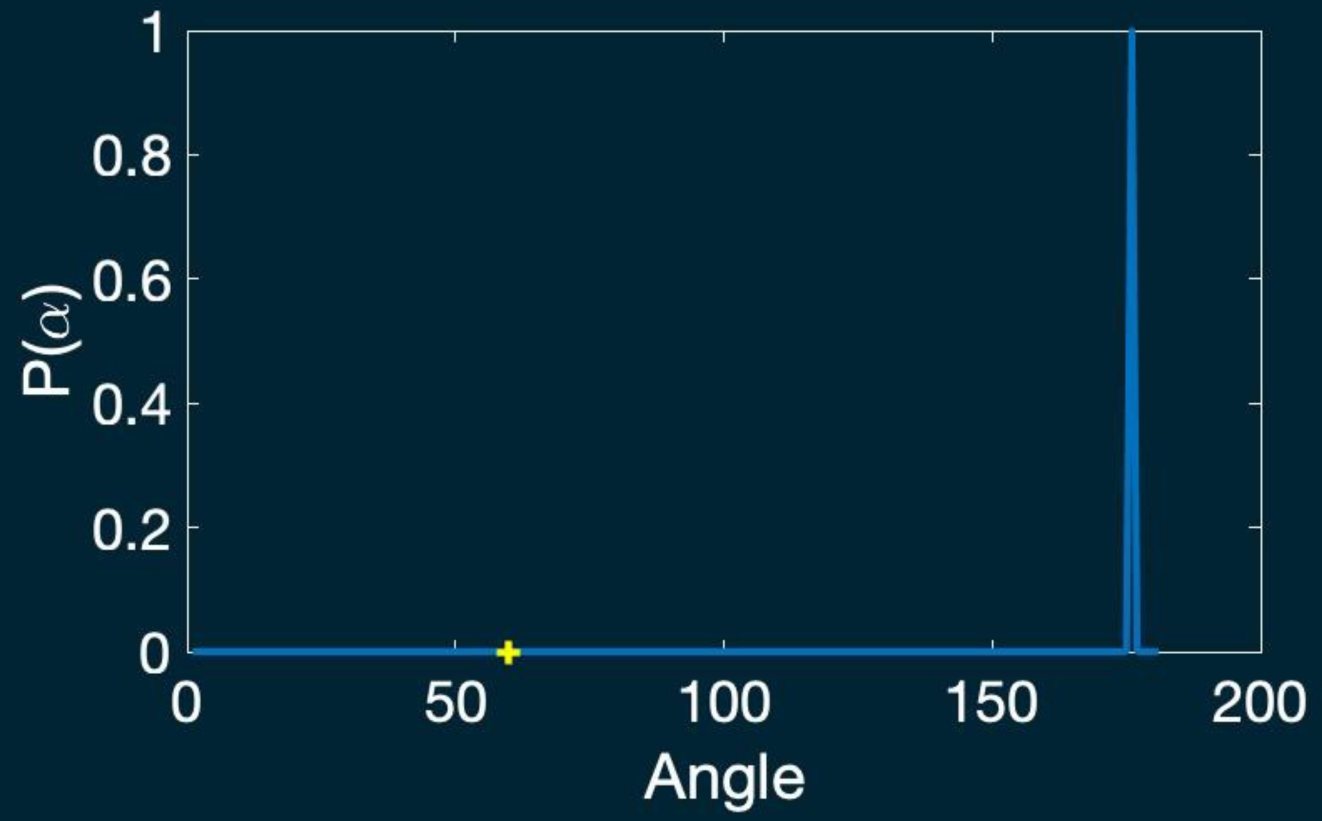


Evaluating Γ_ϕ is expensive: one of 5 parameters



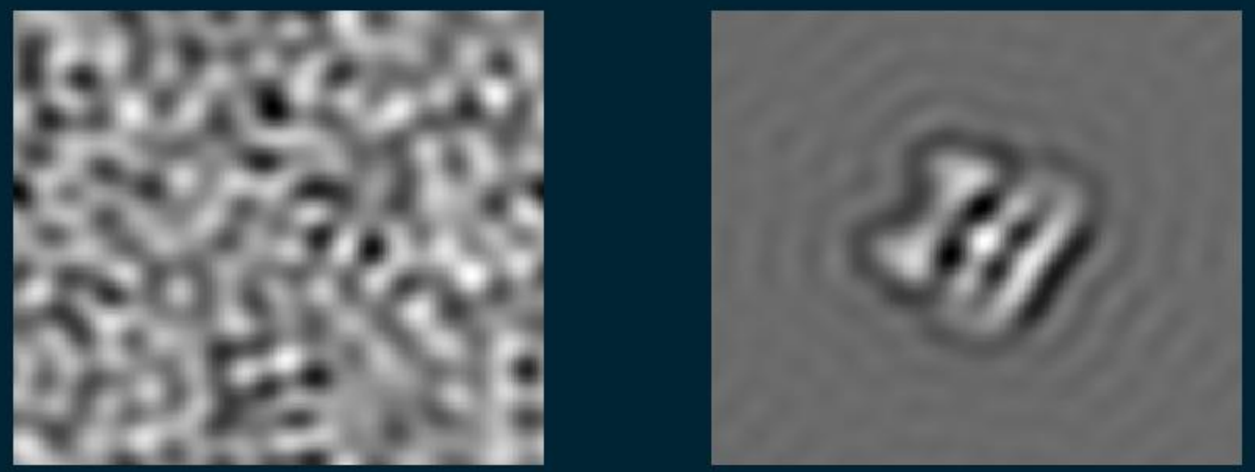




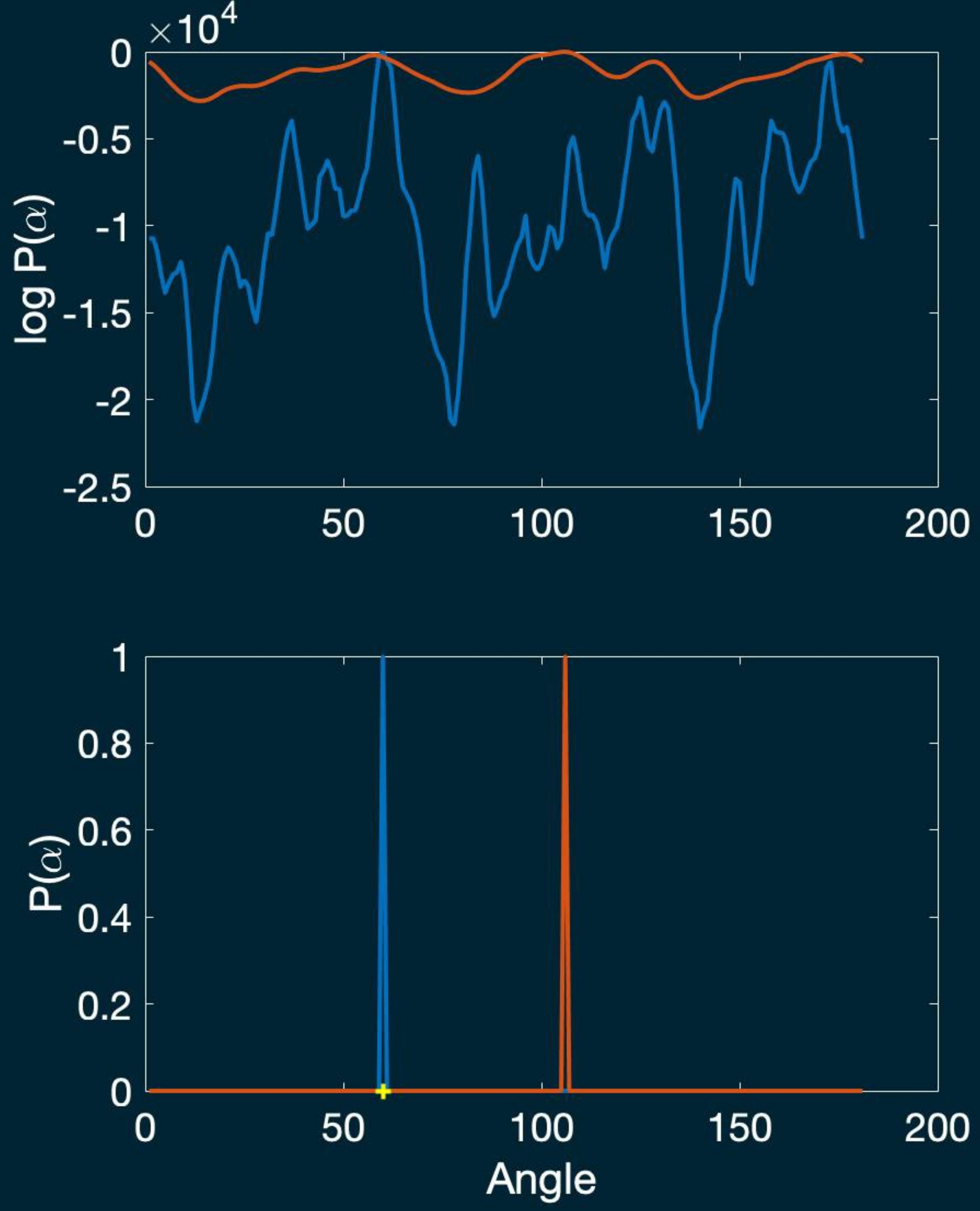


How to decrease the effort?

Domain reduction: branch and bound, illustrated for 1D

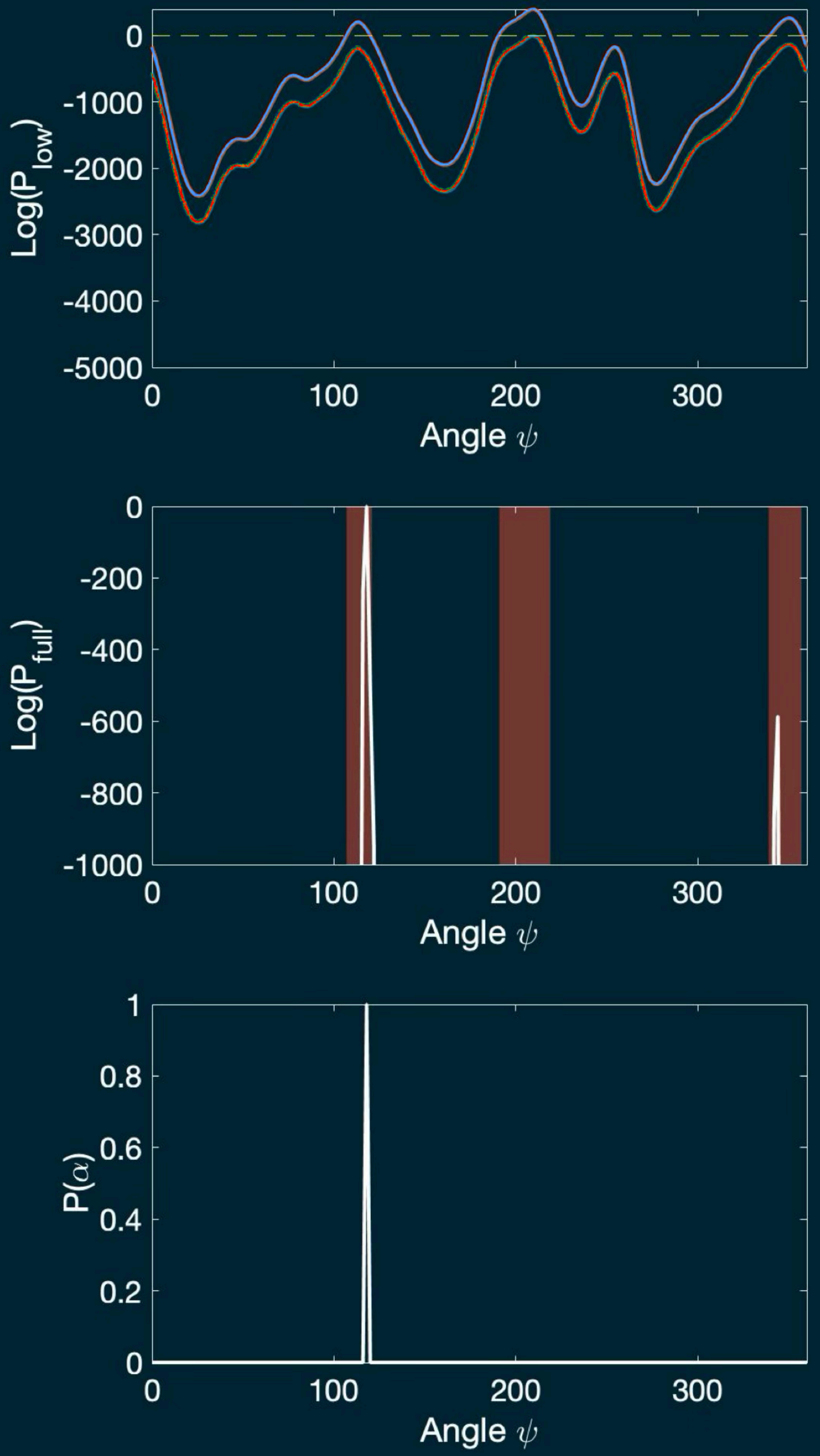


1. To save time, we compute probabilities of orientations at low resolution.

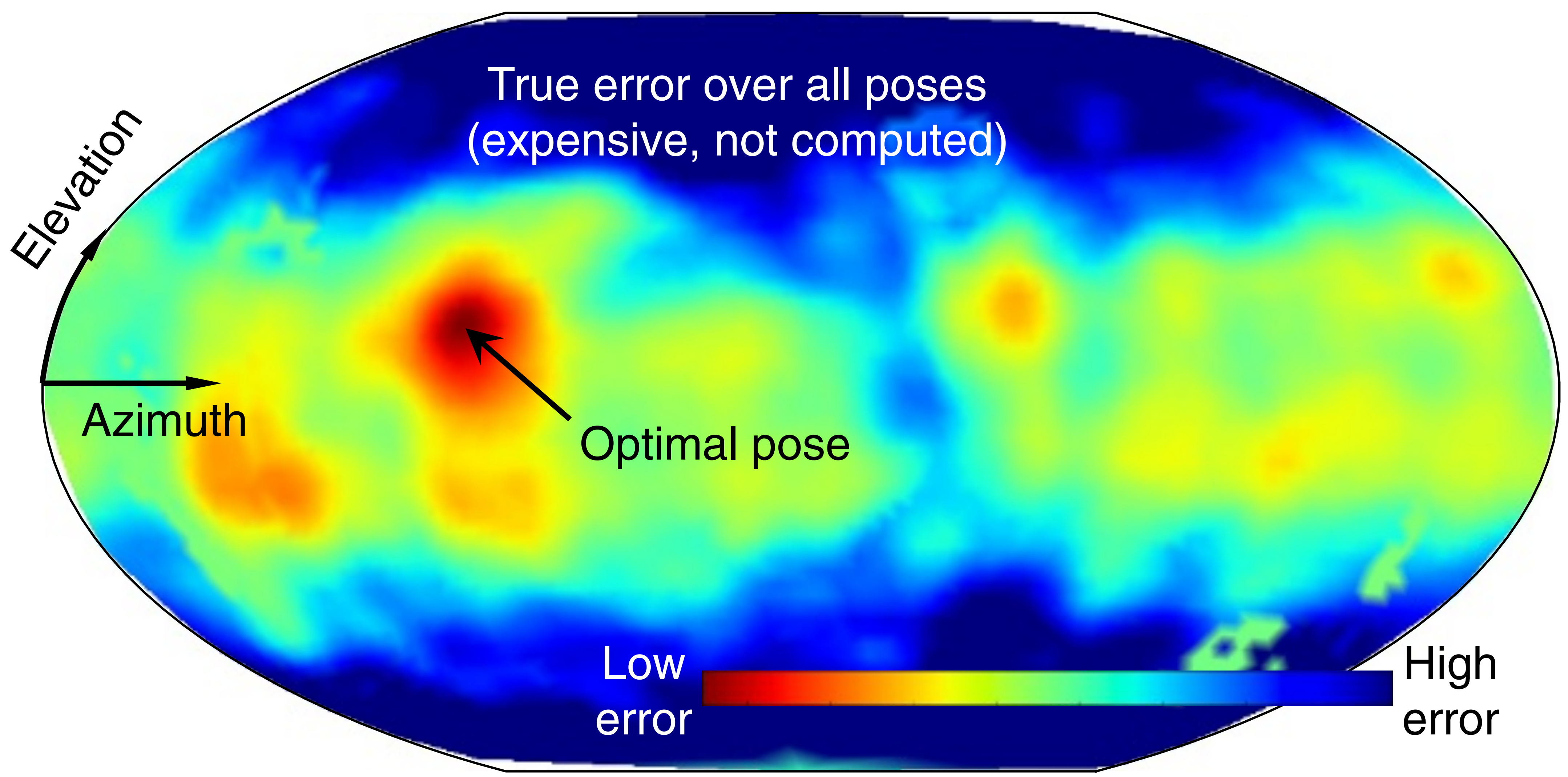


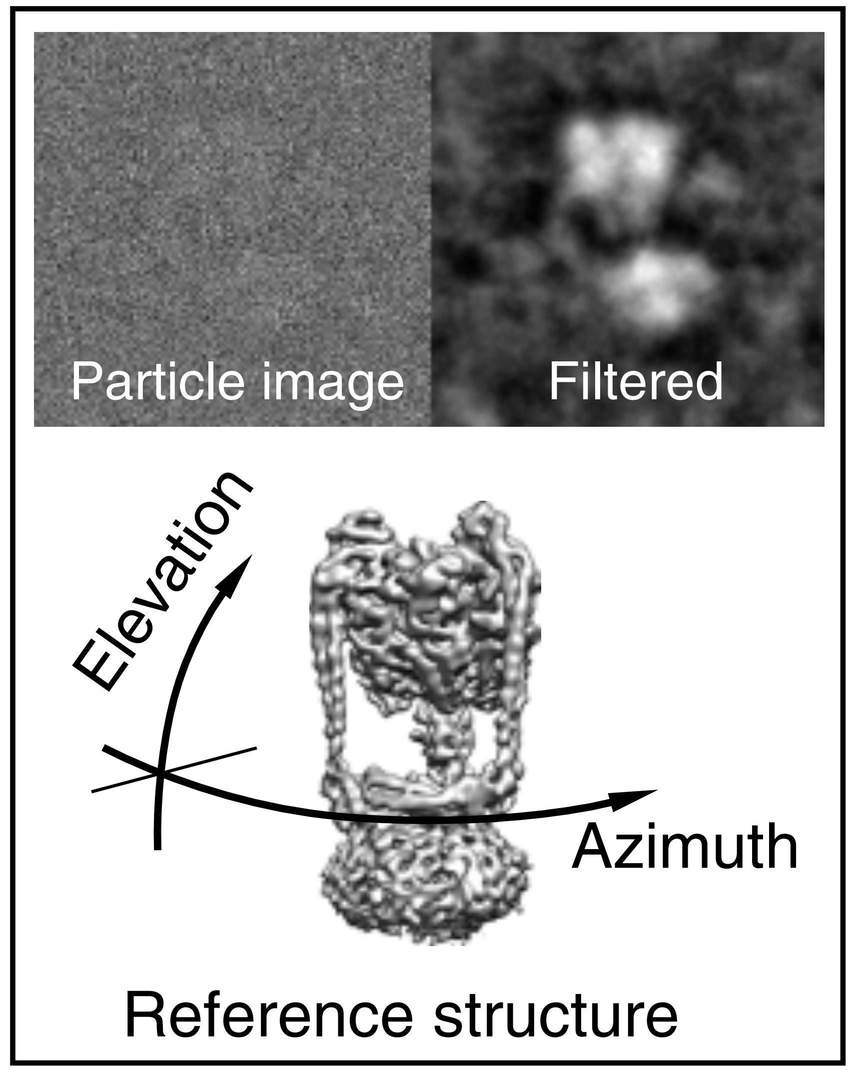
2. We place bounds on how much higher the probabilities could be at full resolution.

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

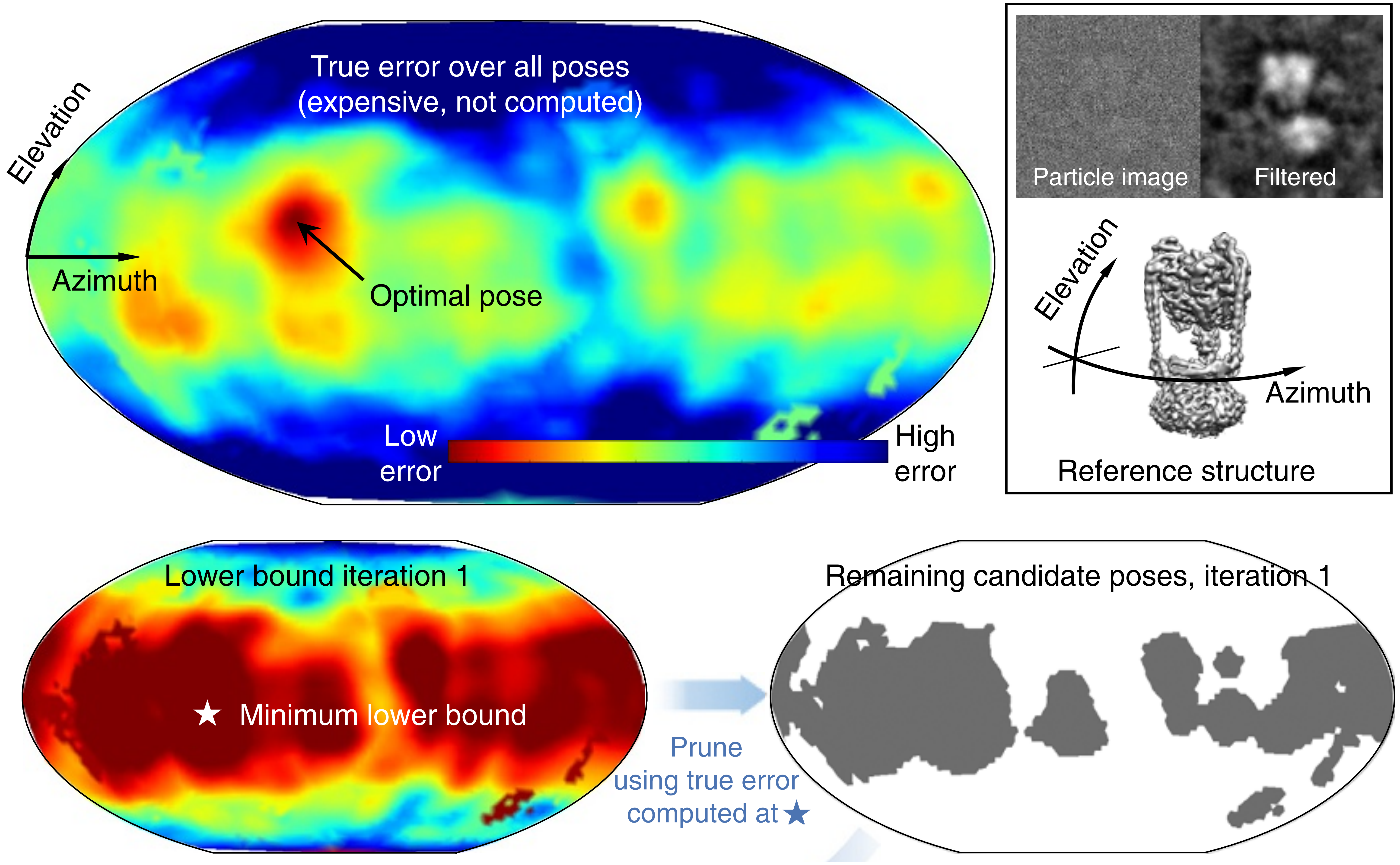


Branch-and-bound in cryoSPARC for integrating over orientations

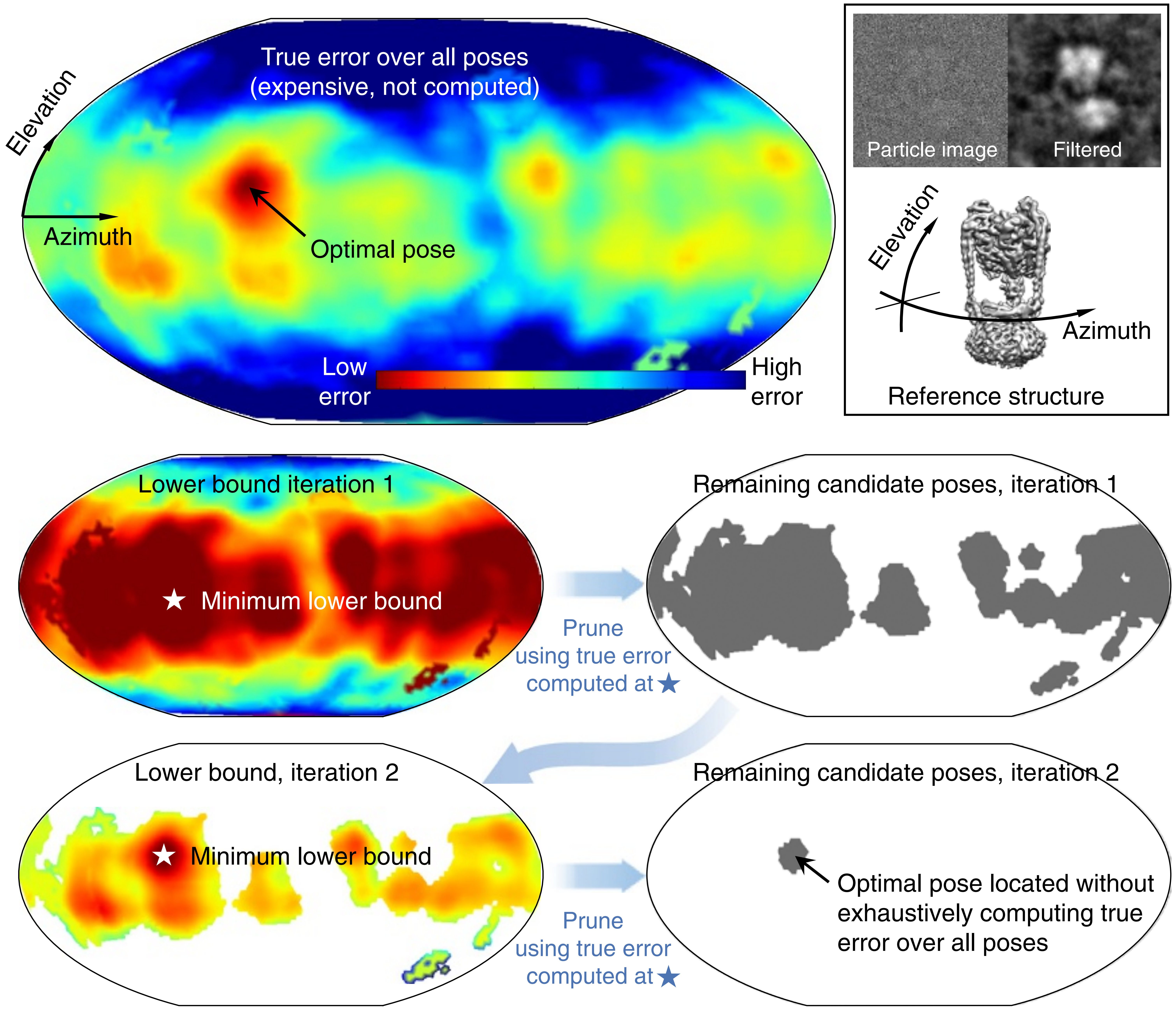




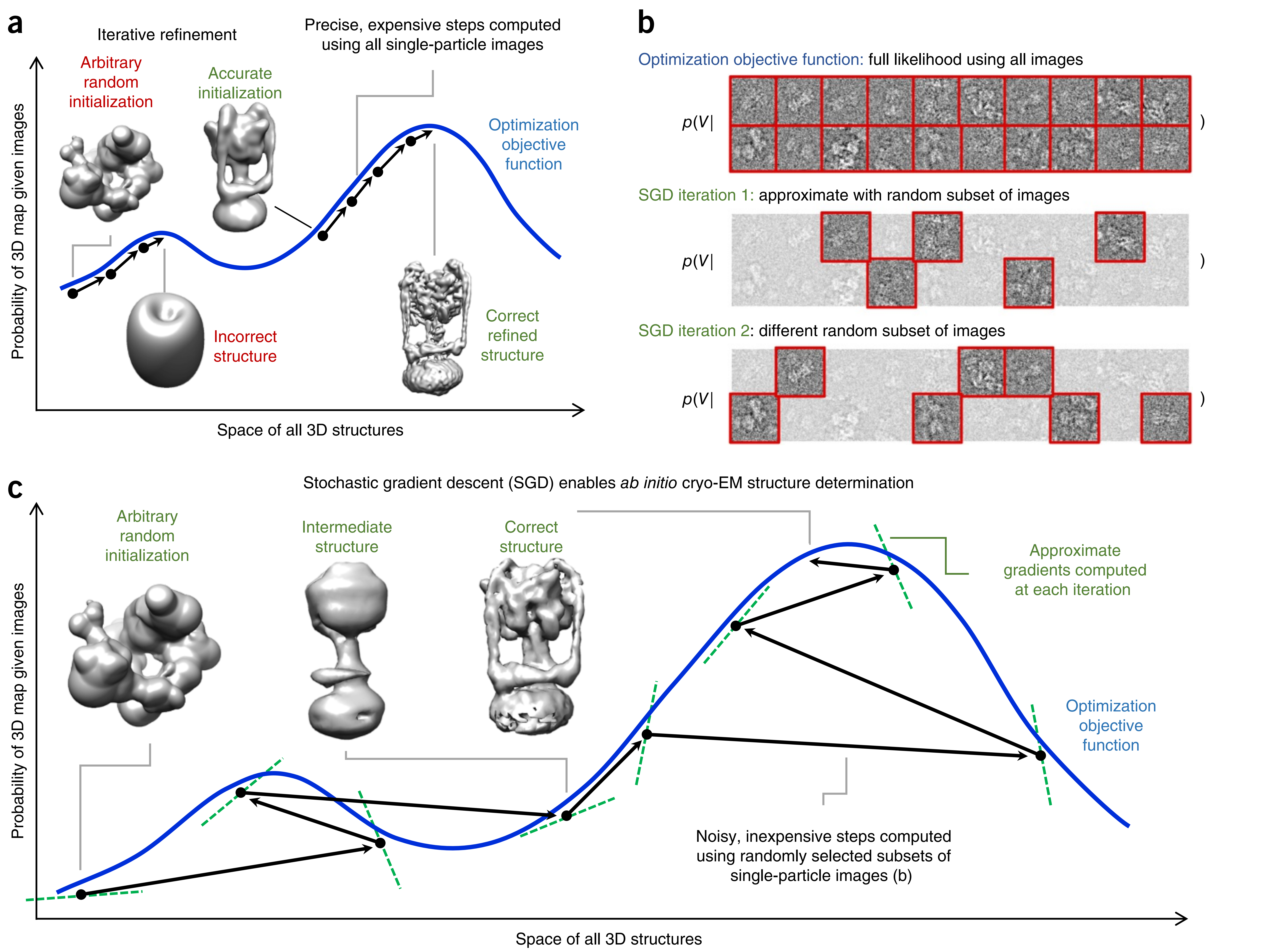
Branch-and-bound in cryoSPARC for integrating over orientations

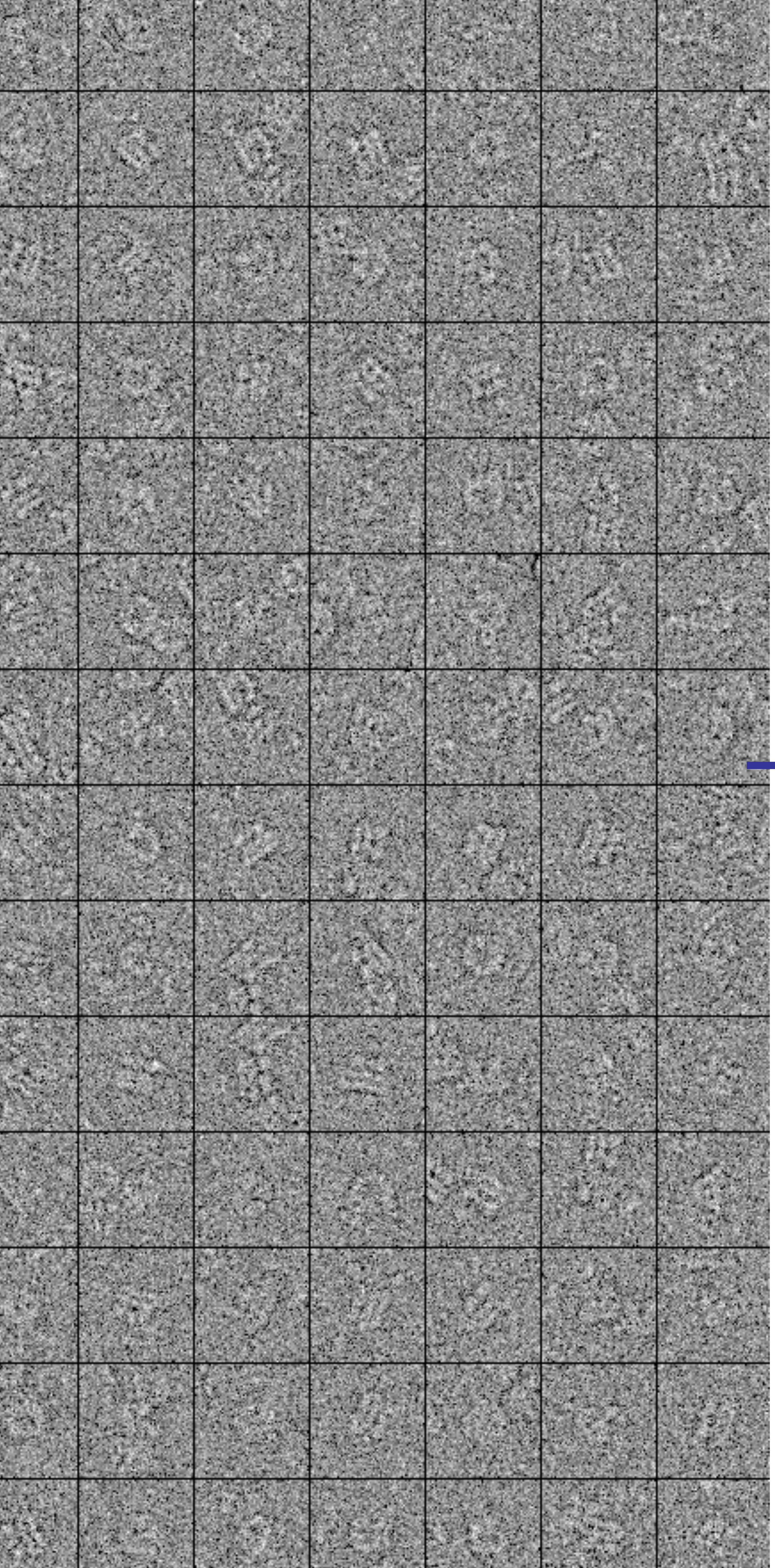


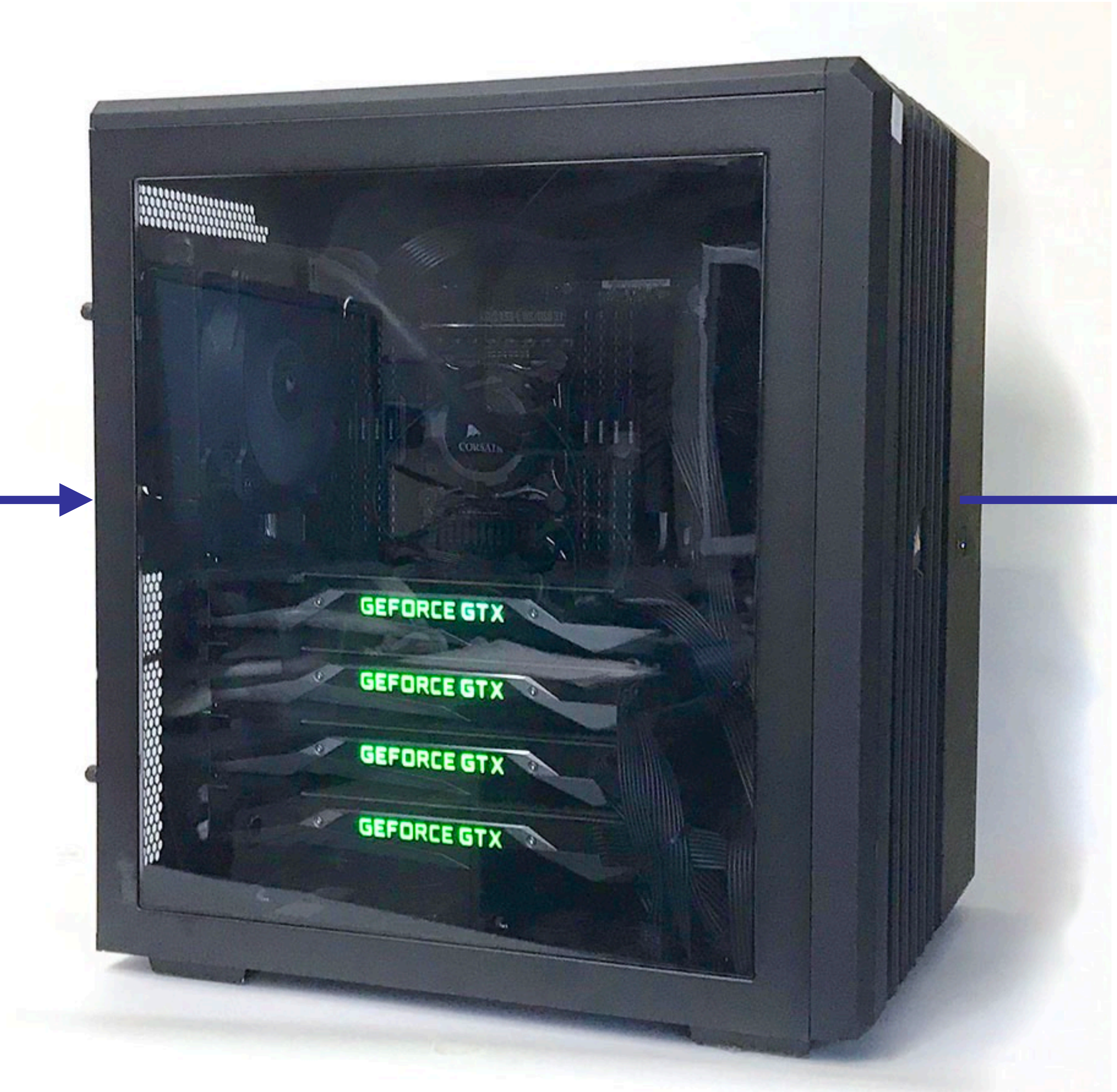
Branch-and-bound in cryoSPARC for integrating over orientations

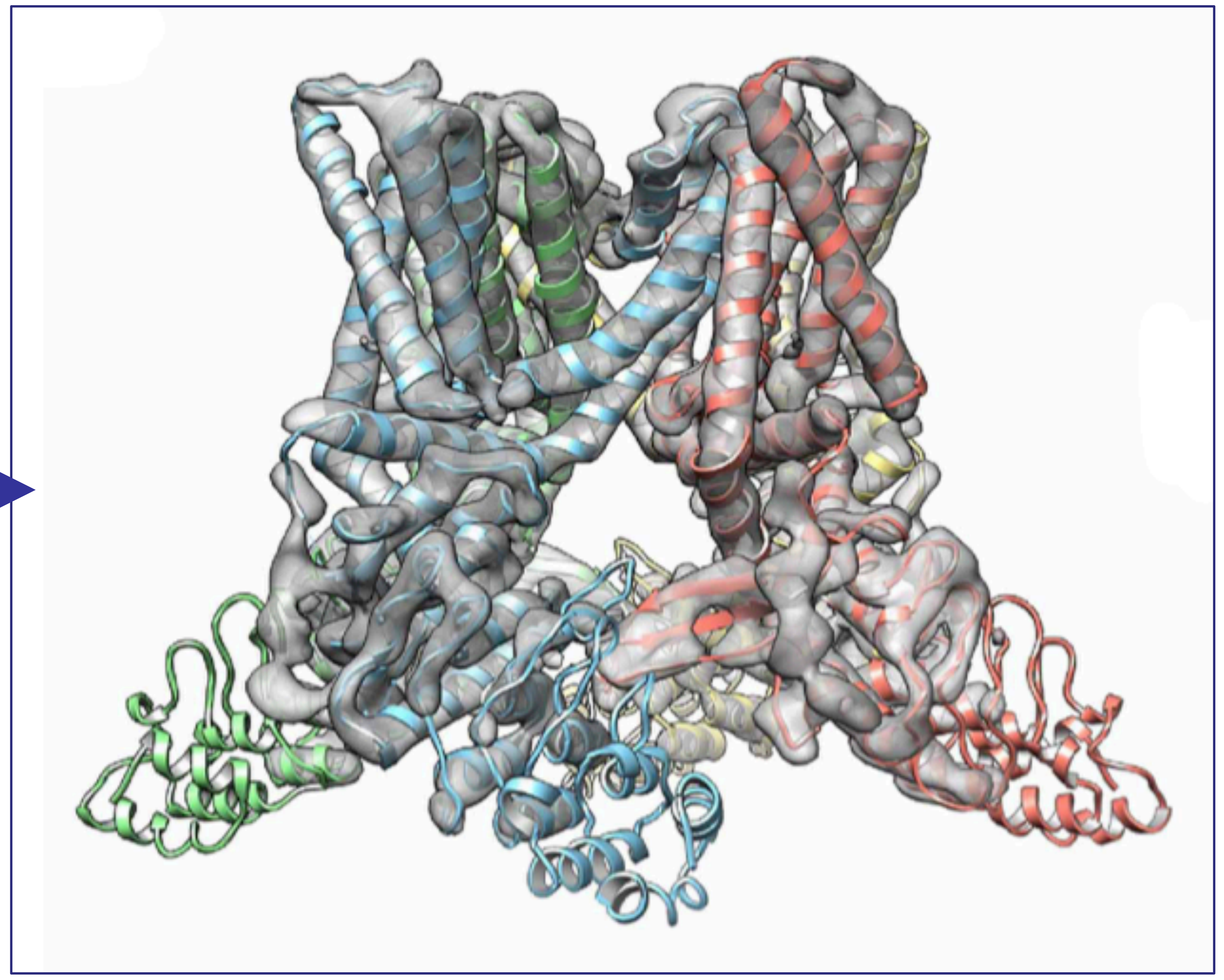


Stochastic gradient descent to avoid model bias









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