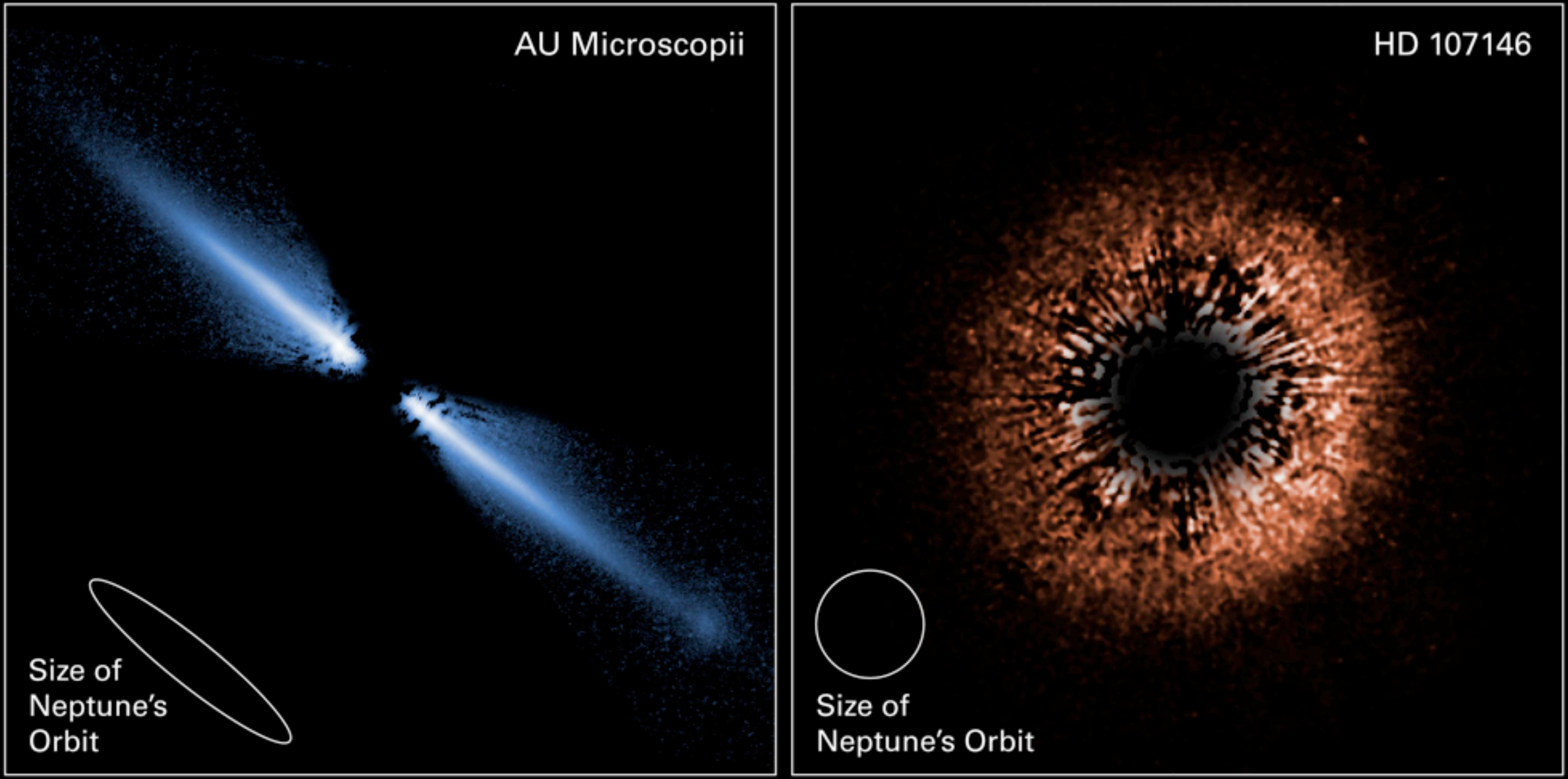
Radiative Equilibrium

Thomas Robitaille (Max Planck Institute for Astronomy)

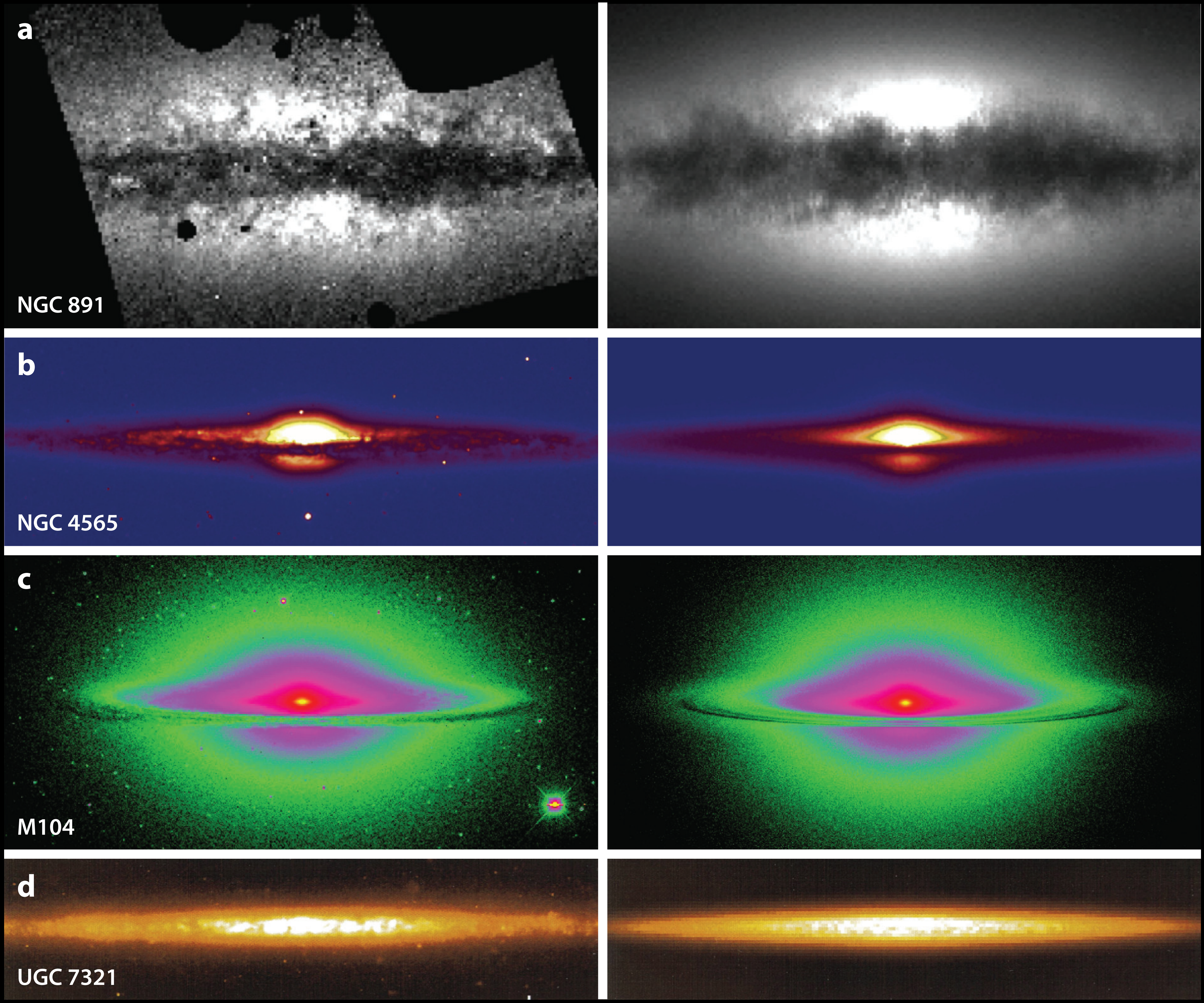


Circumstellar Debris Disks
Hubble Space Telescope • ACS HRC

NASA, ESA, J. Krist (STScI/JPL), D.R. Ardila (JHU), D.A. Golimowski (JHU), M. Clampin (NASA/Goddard), H. Ford (JHU), G. Hartig (STScI), G. Illingworth (UCO-Lick) and the ACS Science Team

STScI-PRC04-33a

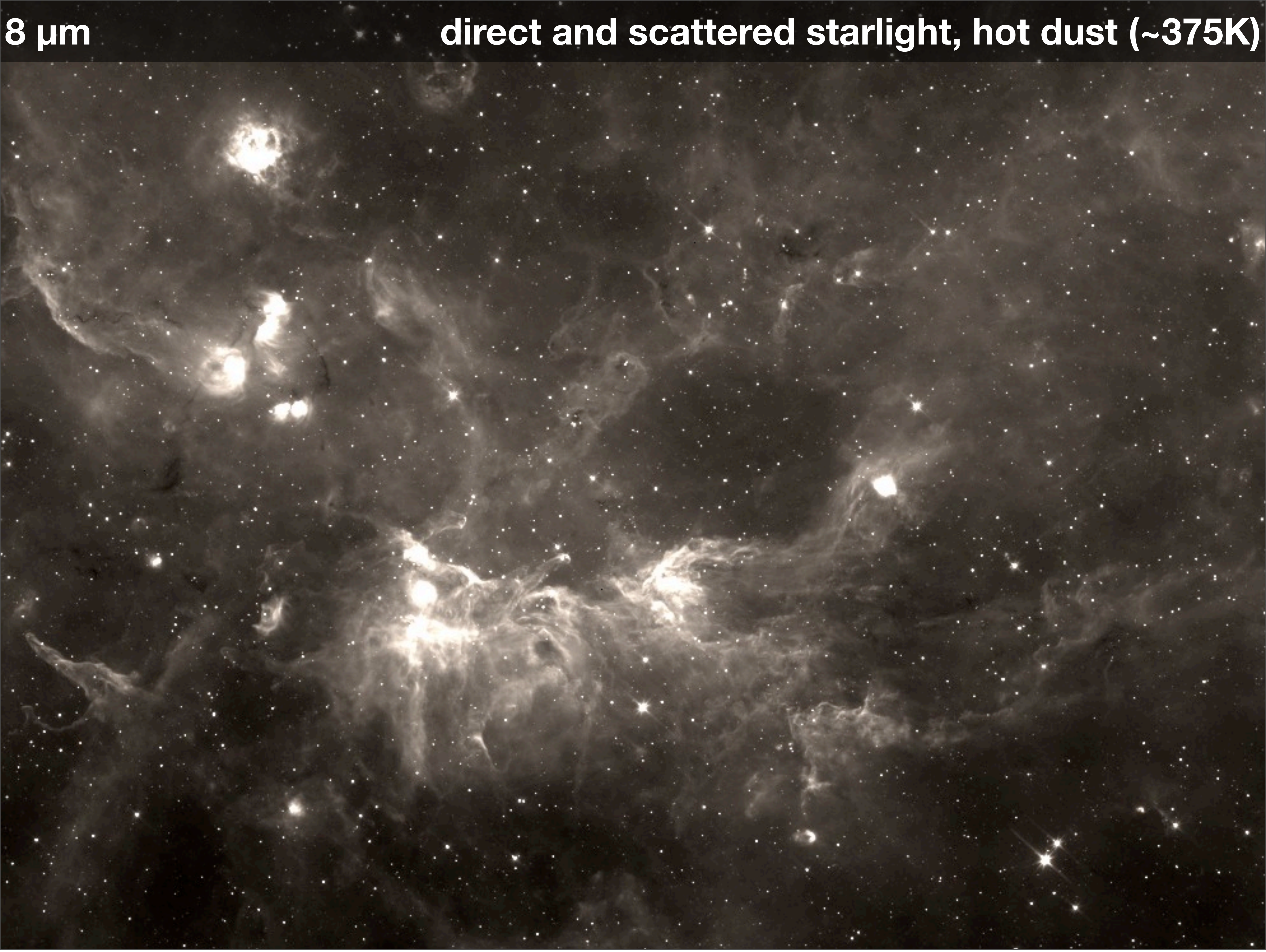
Steinacker et al (2013)



1 µm

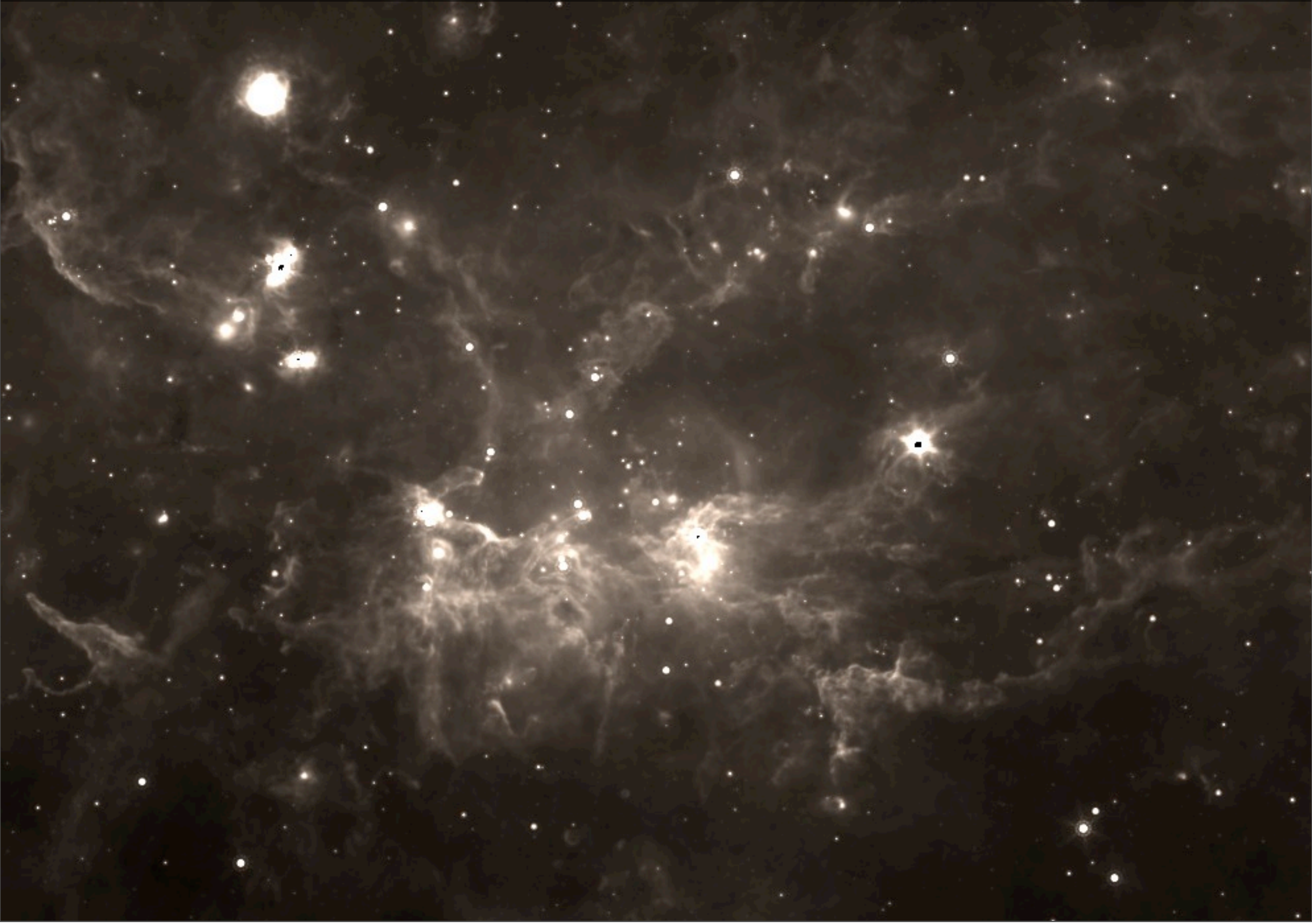
starlight, dust extinction

8 µm direct and scattered starlight, hot dust (~375K)



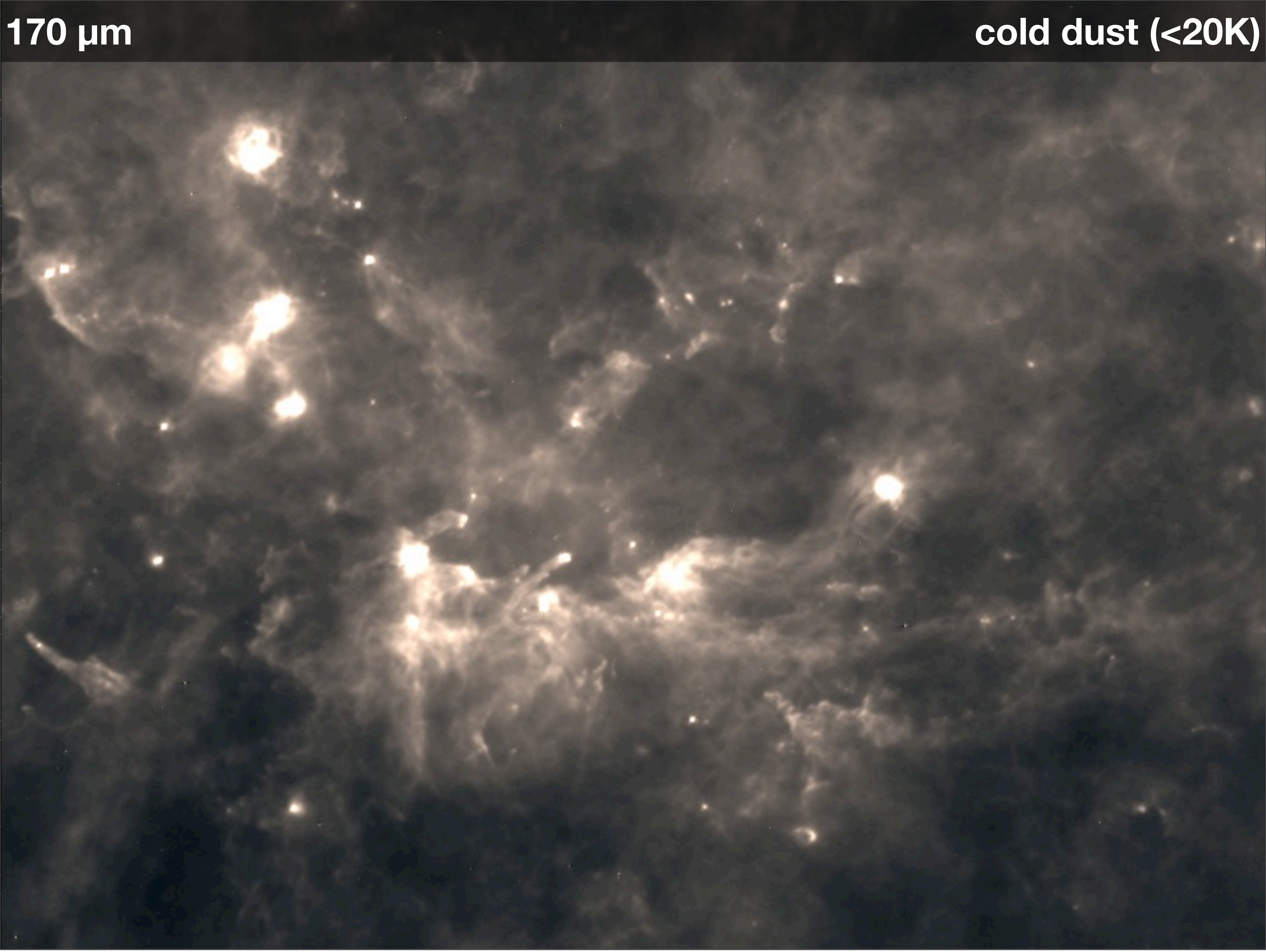
24 µm

warm dust (~125K)

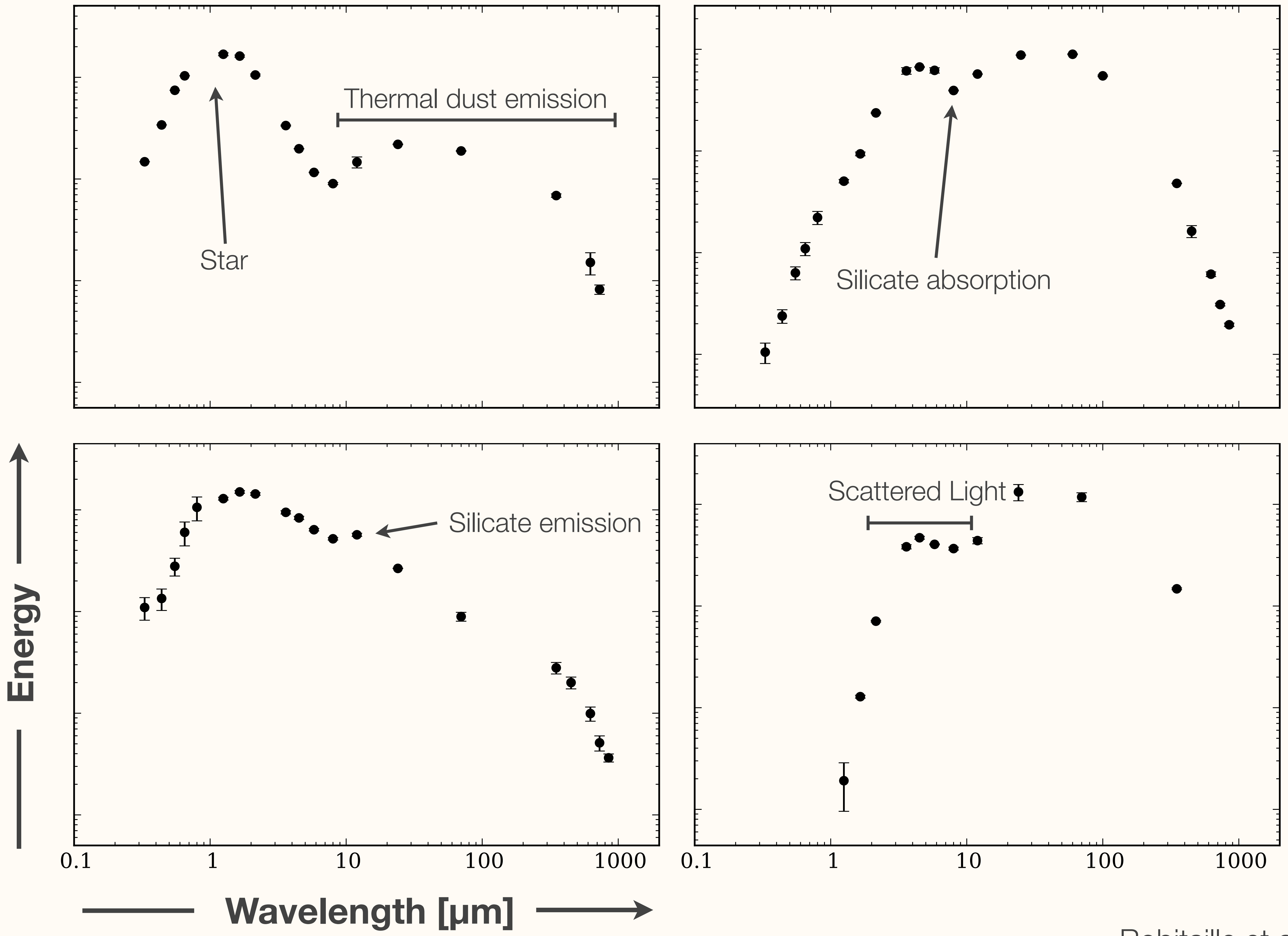


170 μm

cold dust (<20K)



Spectral energy distributions of forming stars



Robitaille et al, (2007)

$$\frac{dI_\nu}{ds} = -\,\alpha_\nu I_\nu + j_\nu$$

$$\frac{dI_{x,y,z,\theta,\phi,\nu}}{ds} = -\,\alpha_{x,y,z,\nu} I_{x,y,z,\theta,\phi,\nu} + j_{x,y,z,\theta,\phi,\nu}(T)$$

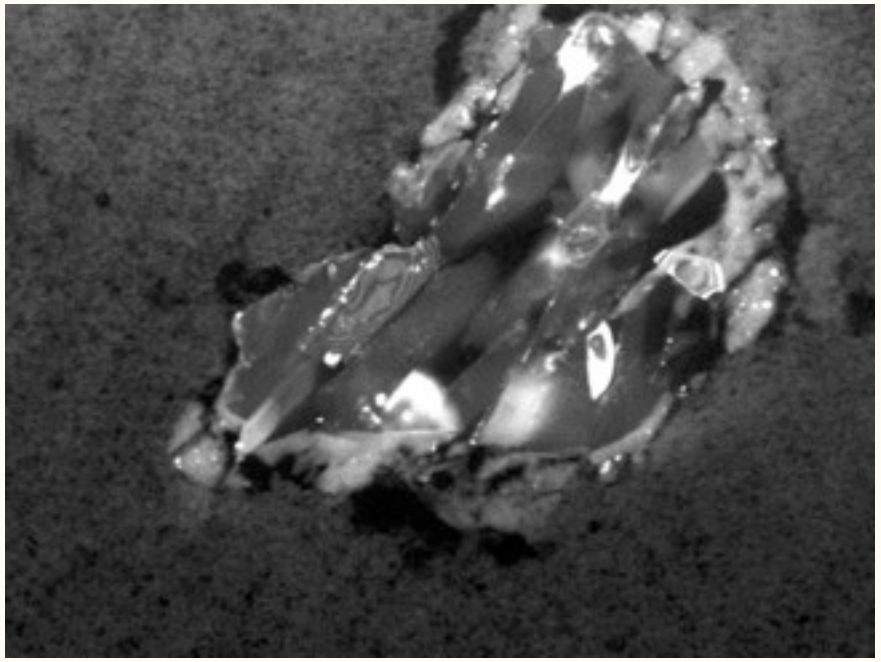
Dust versus Gas

For dust in LTE, emissivities are given by $\kappa_\nu B_\nu(T)$

For lines, need to take into account gas velocities, line profiles, level populations, etc.

In this lecture we focus on **dust** radiative transfer

Astrophoto.com

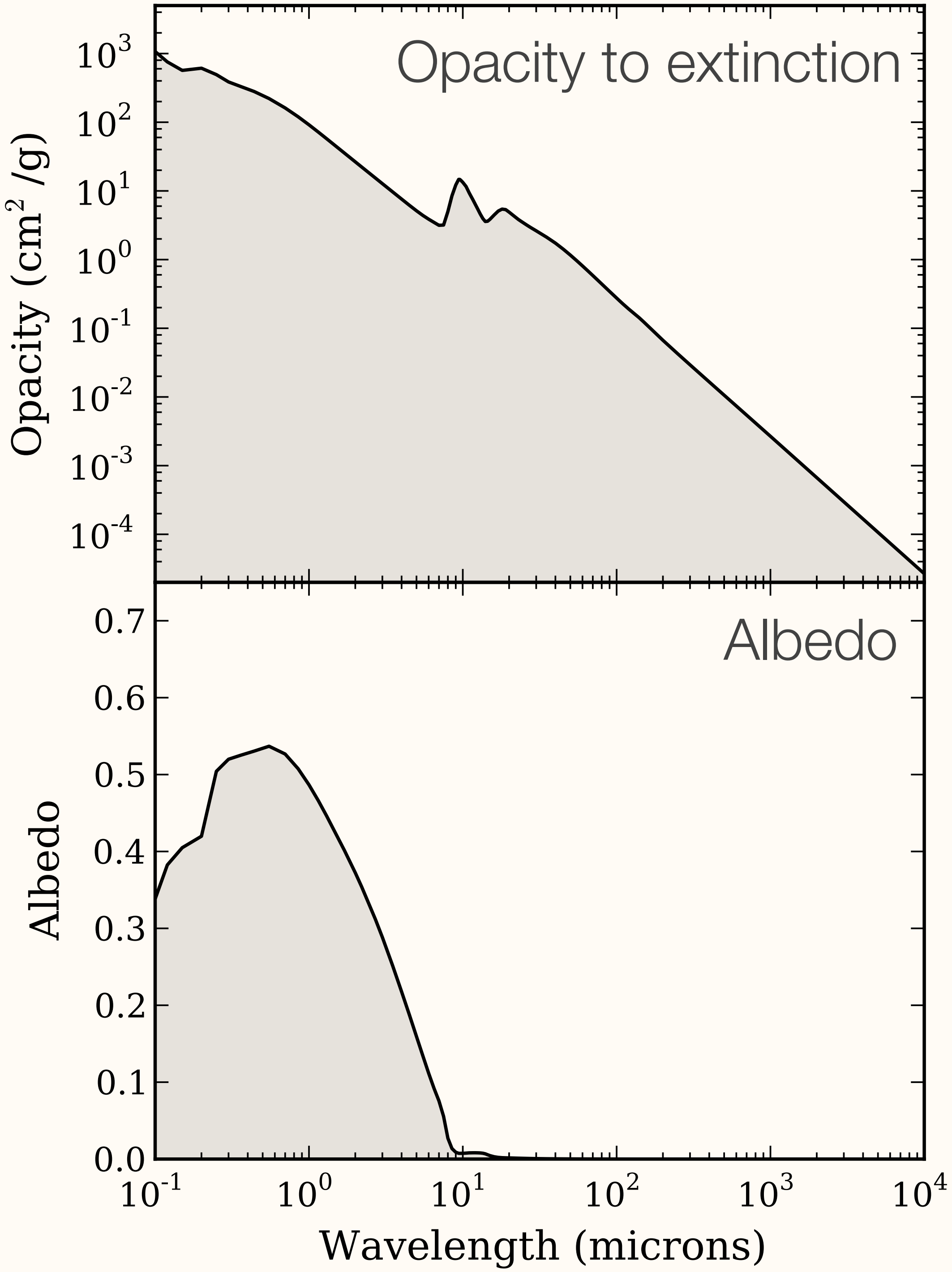


Forsterite crystal from Comet Wild 2 (NASA)

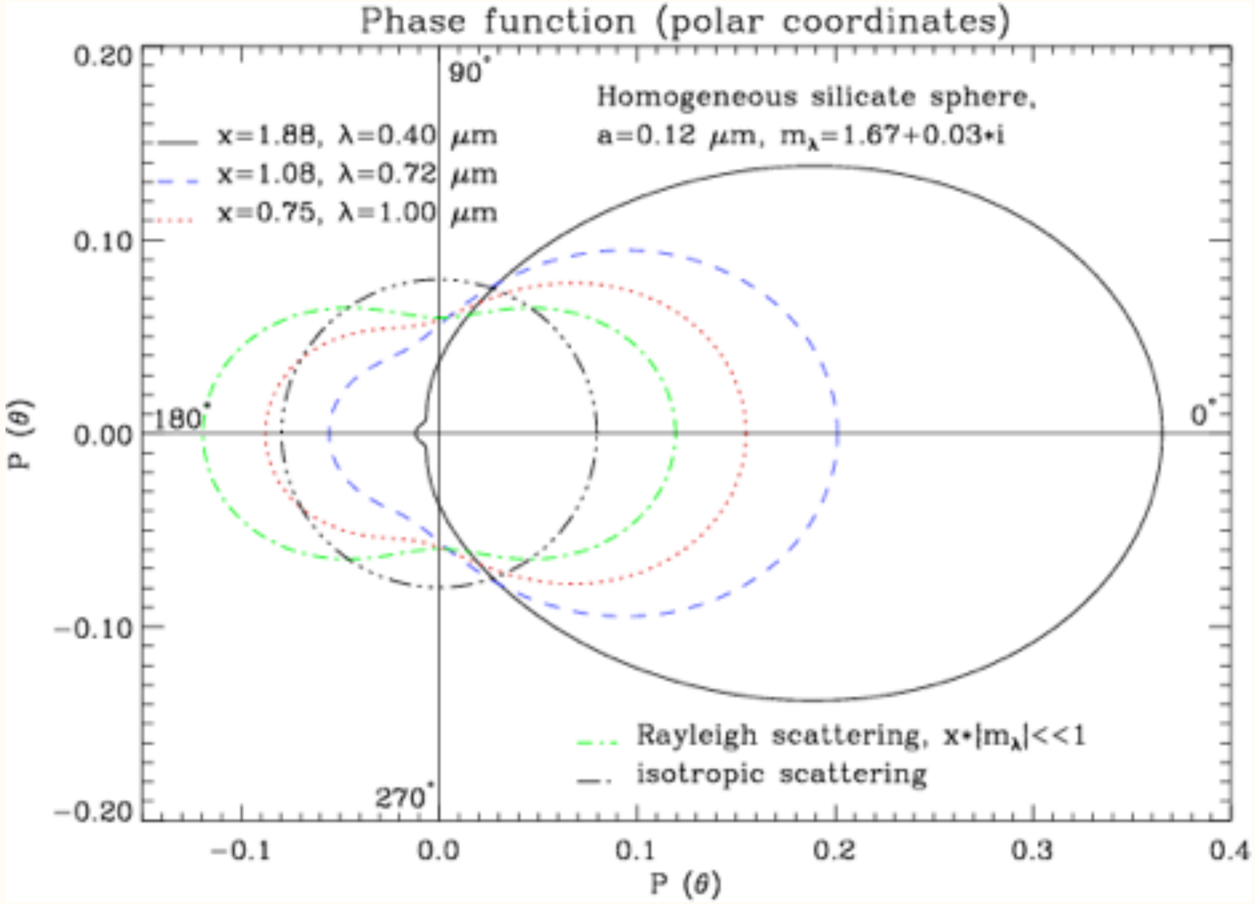
Composition:

- Silicates
- Carbon
- Water
- Organic molecules

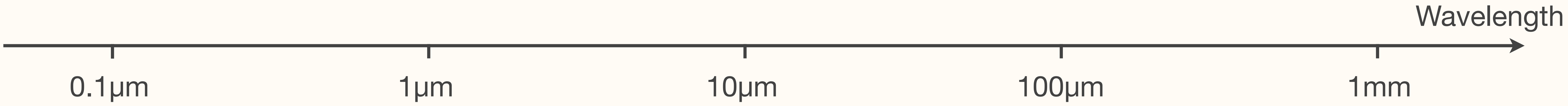
Sizes from nm to mm



Scattering Phase Function



Steinacker et al. (2003)



Wavelength

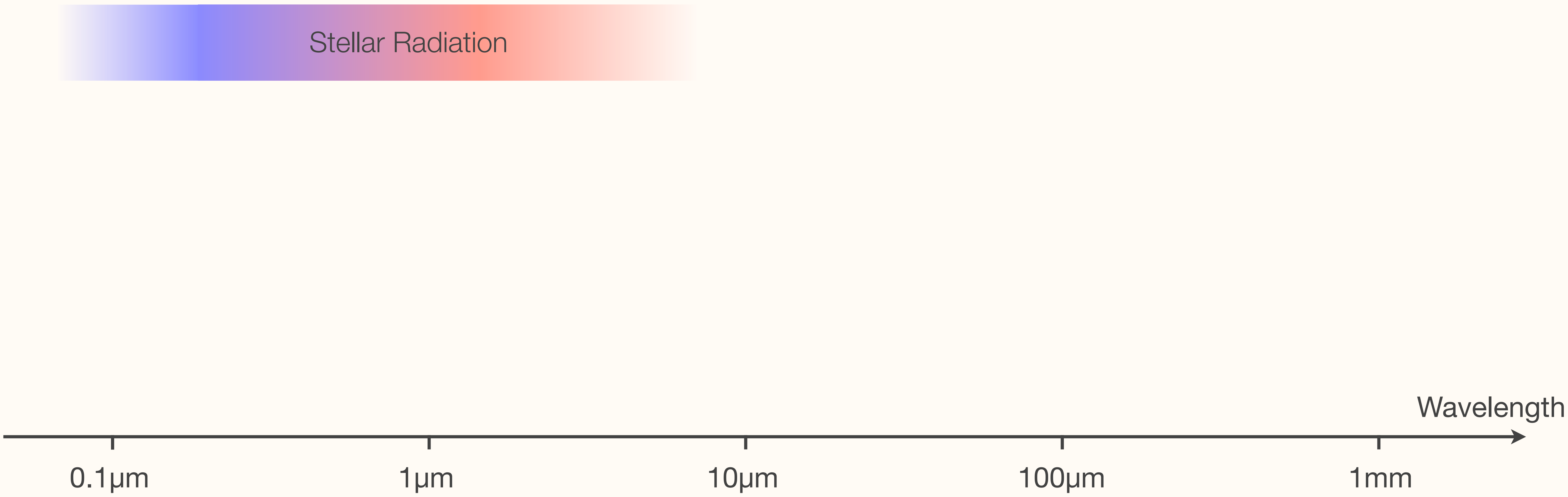
0.1μm

1μm

10μm

100μm

1mm



Stellar Radiation

Wavelength

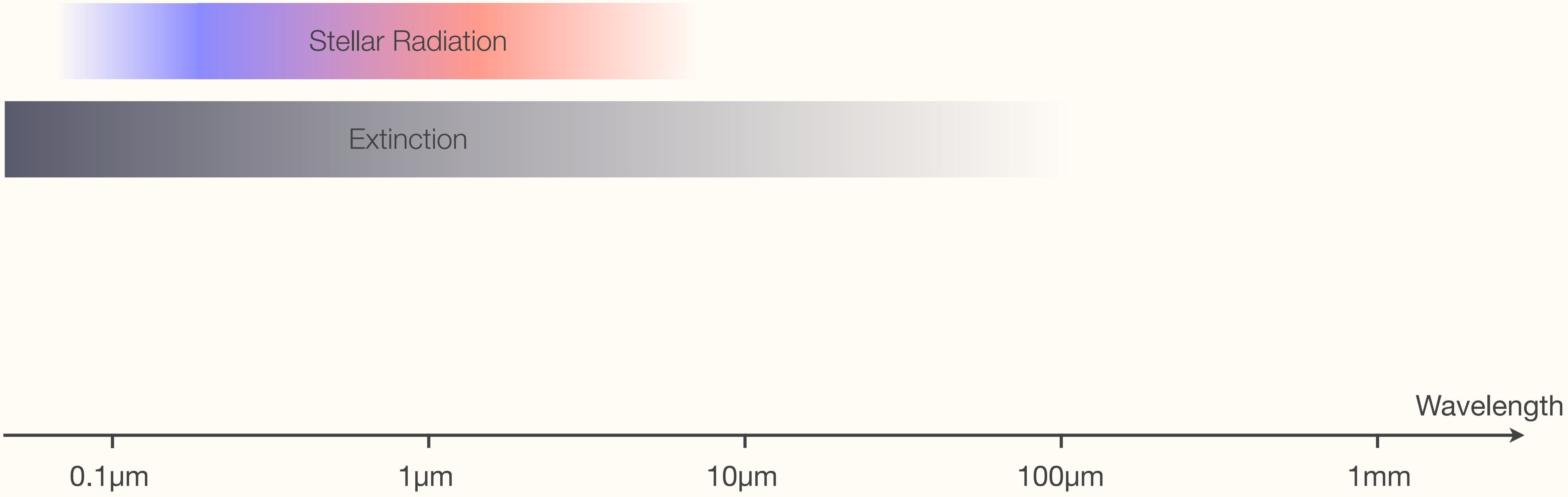
0.1µm

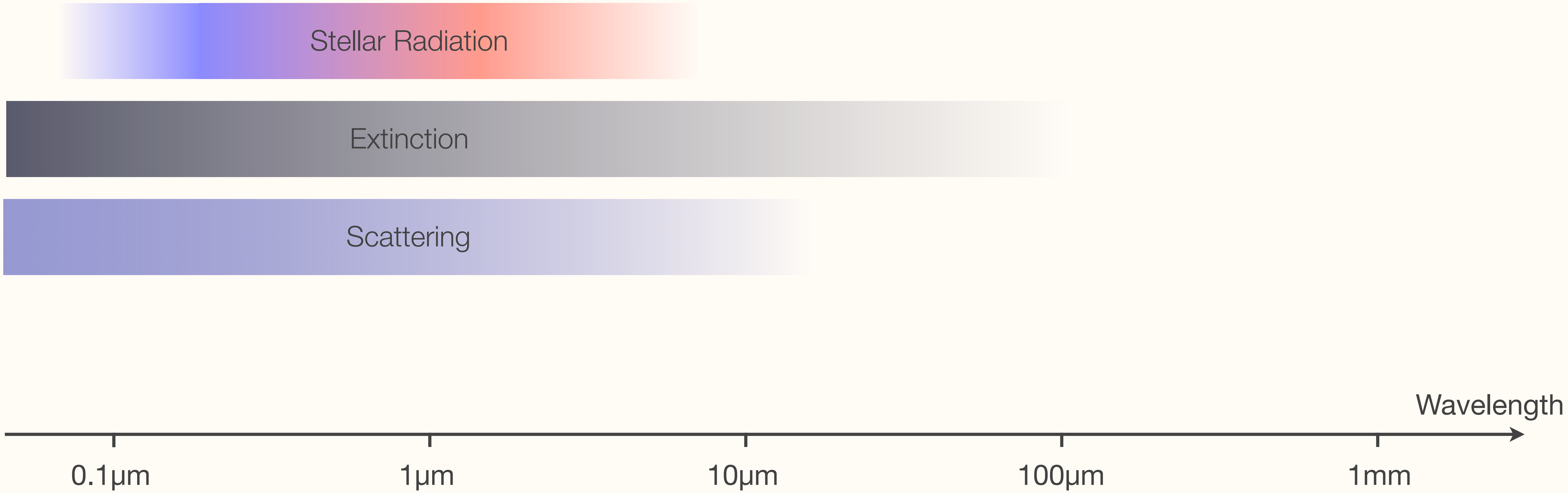
1µm

10µm

100µm

1mm


Stellar Radiation
Extinction
Wavelength
0.1μm
1μm
10μm
100μm
1mm



Stellar Radiation

Extinction

Scattering

Wavelength

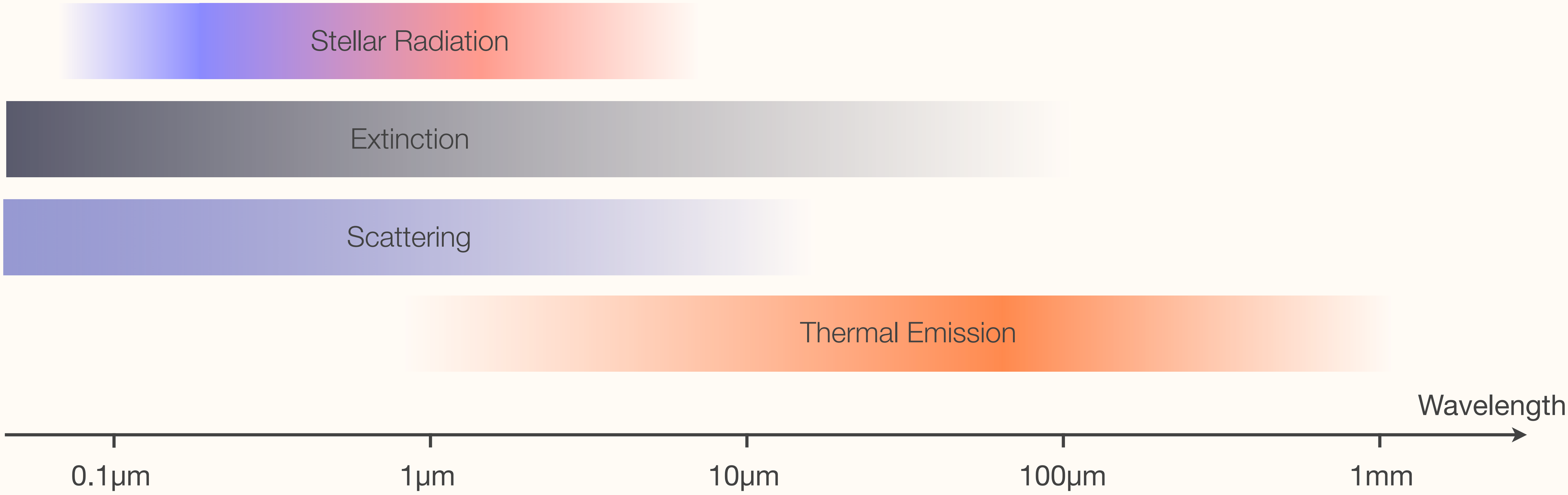
0.1μm

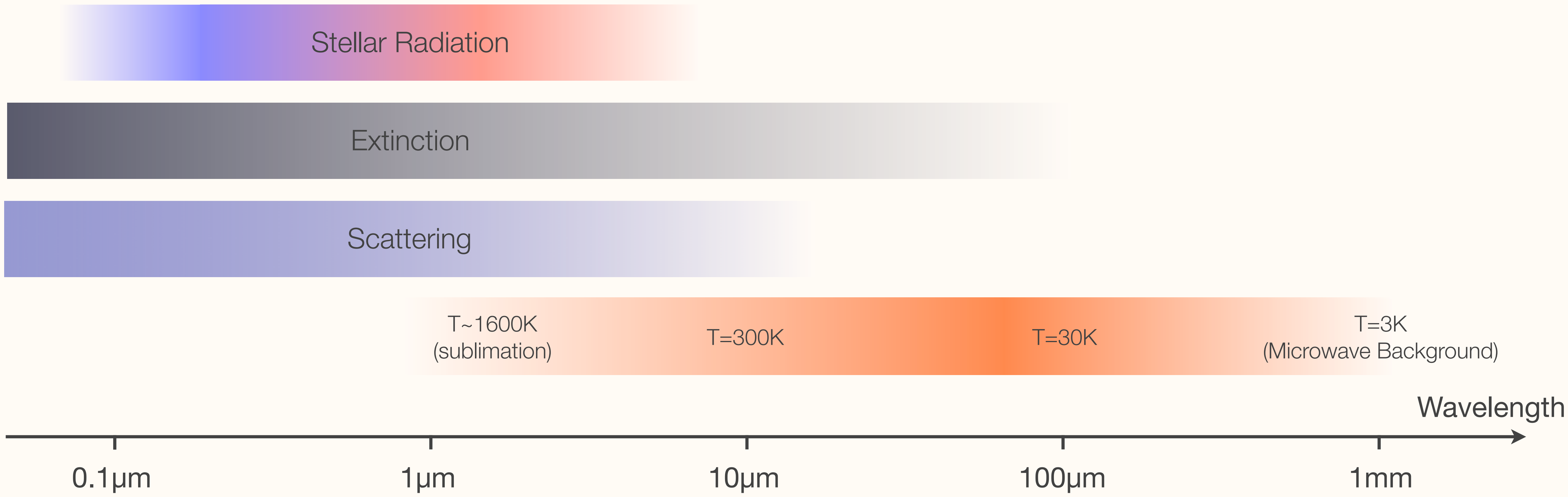
1μm

10μm

100μm

1mm


Stellar Radiation
Extinction
Scattering
Thermal Emission
Wavelength
0.1µm
1µm
10µm
100µm
1mm



Stellar Radiation

Extinction

Scattering

T~1600K
(sublimation)

T=300K

T=30K

T=3K
(Microwave Background)

Wavelength

0.1µm

1µm

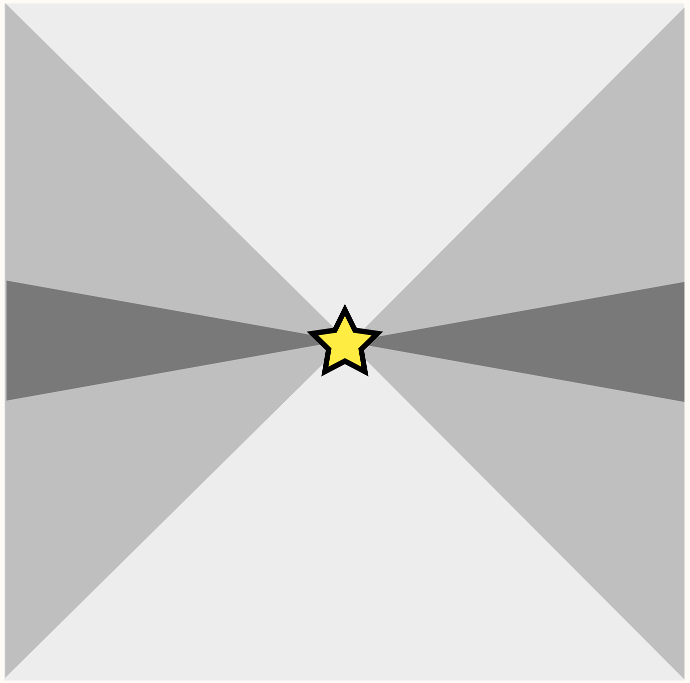
10µm

100µm

1mm

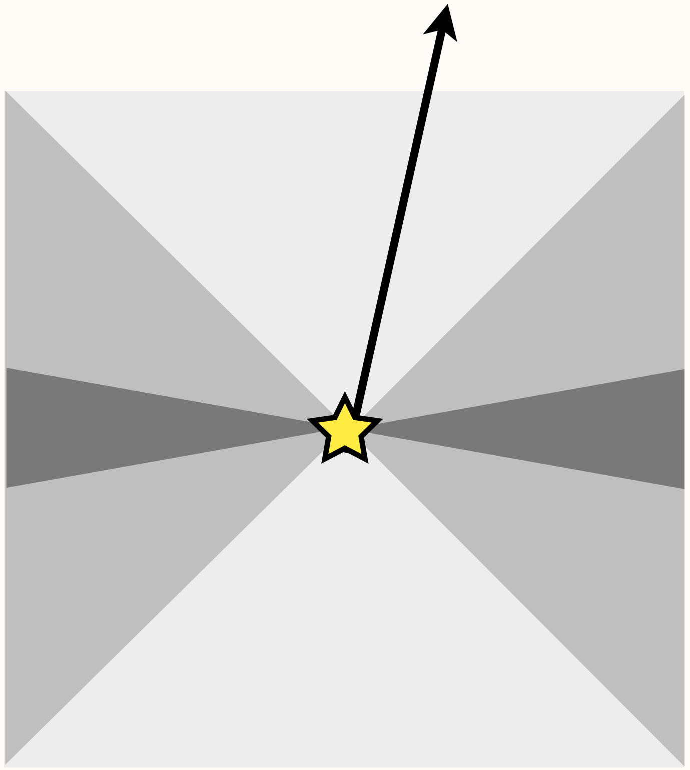
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



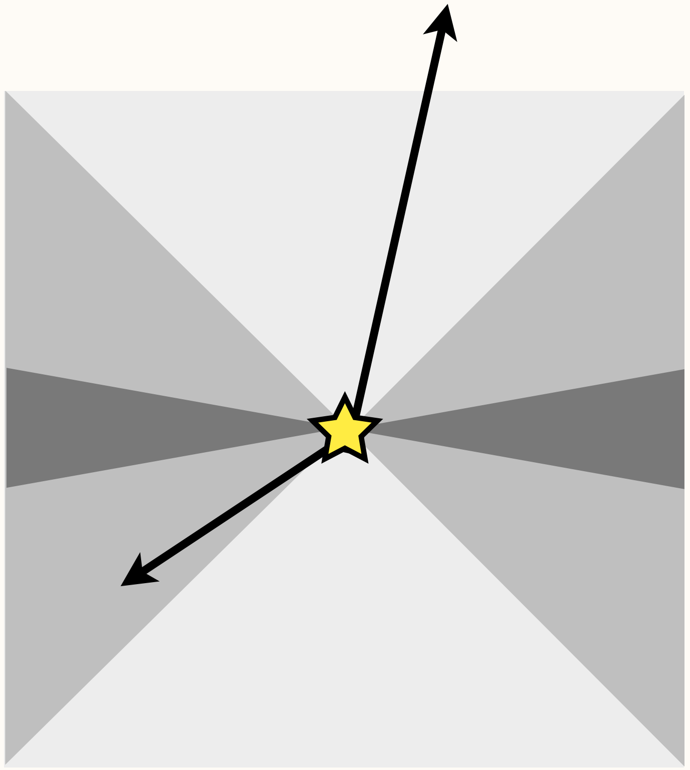
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



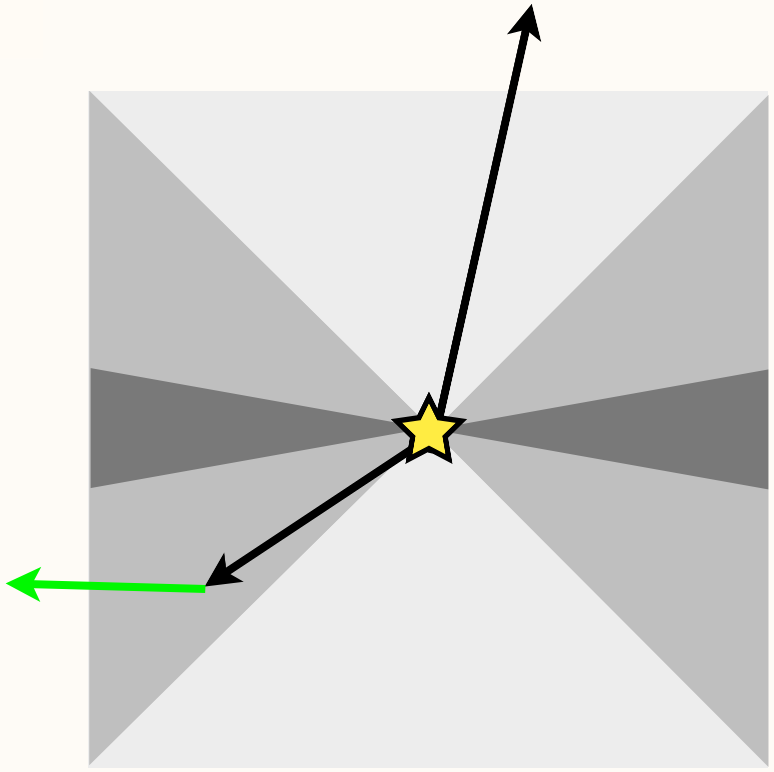
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



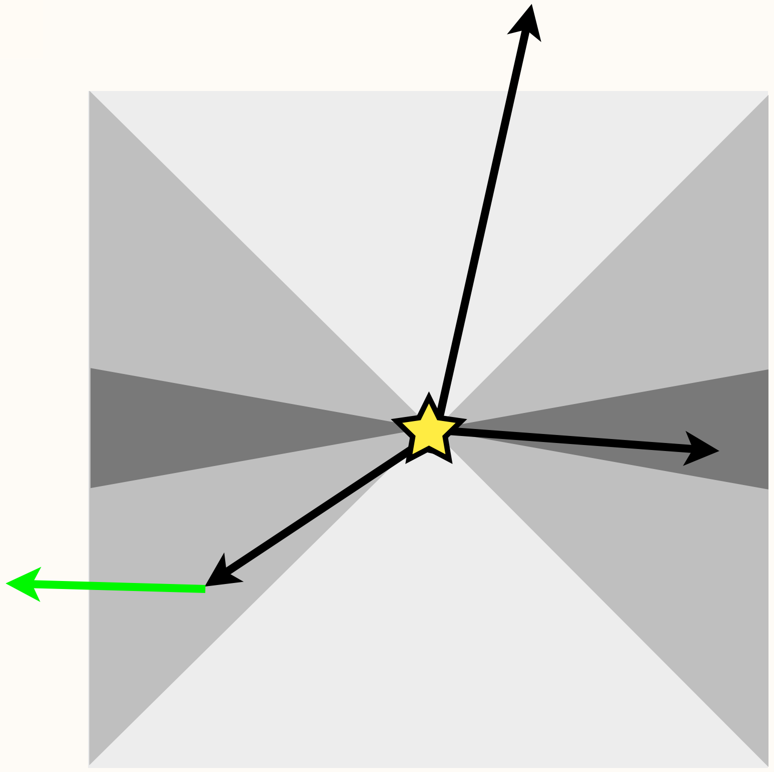
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



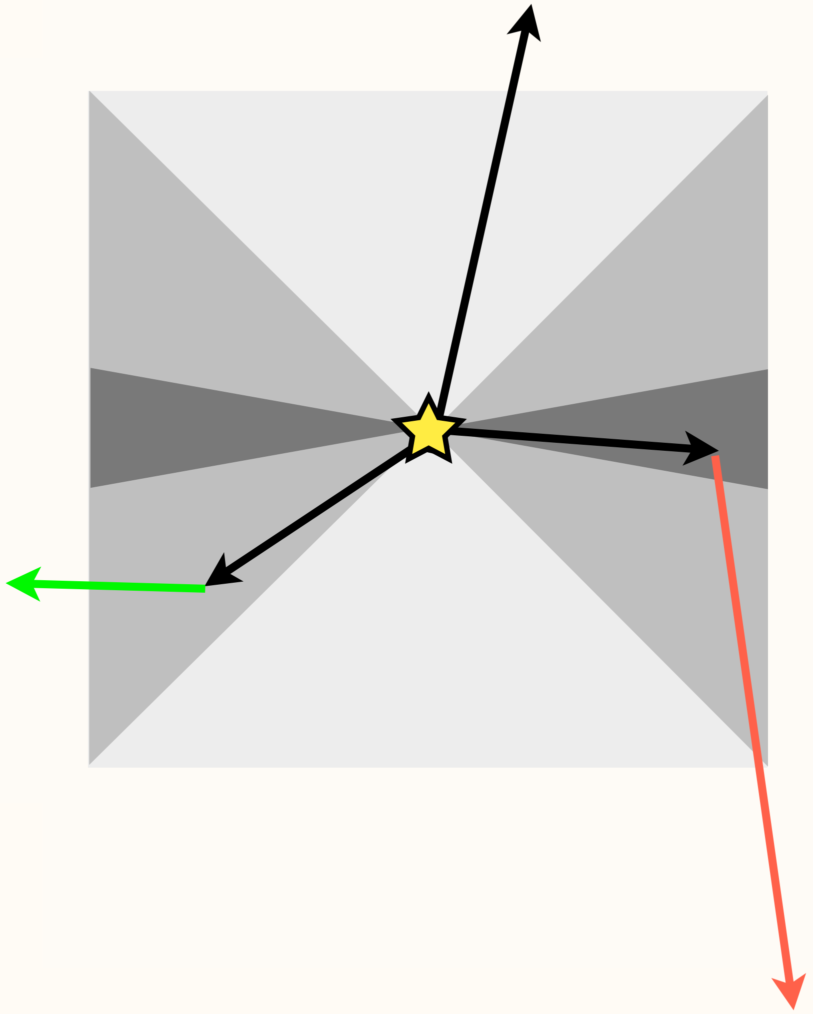
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



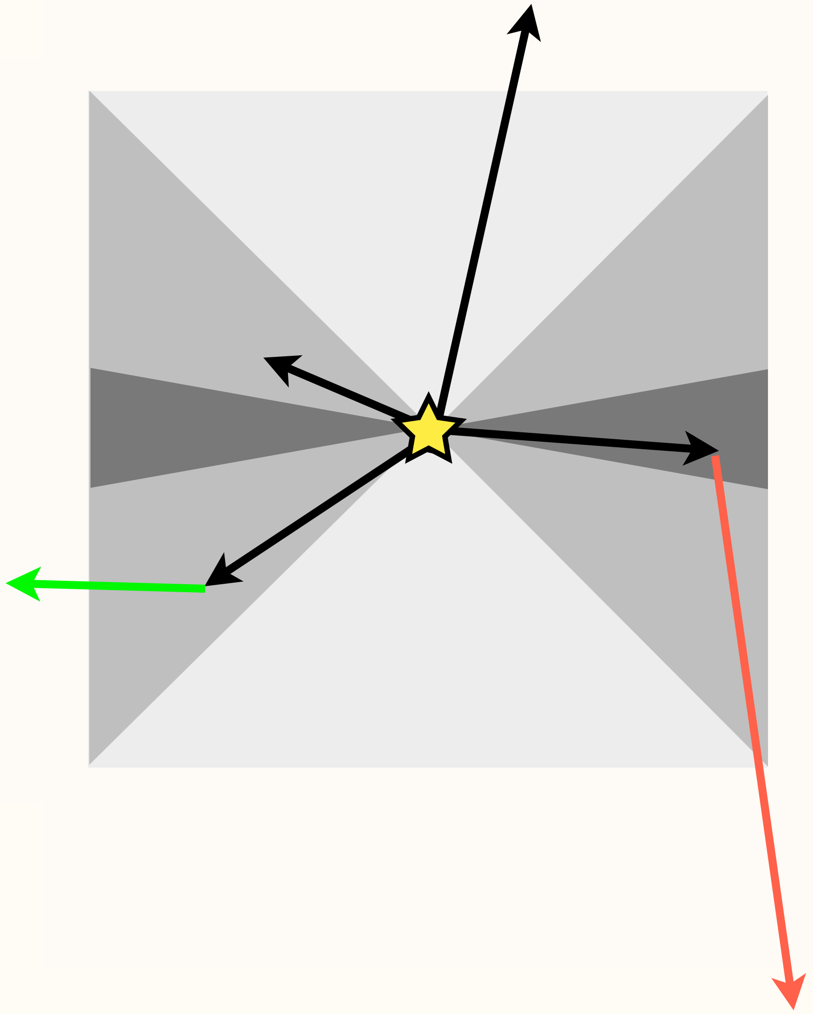
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



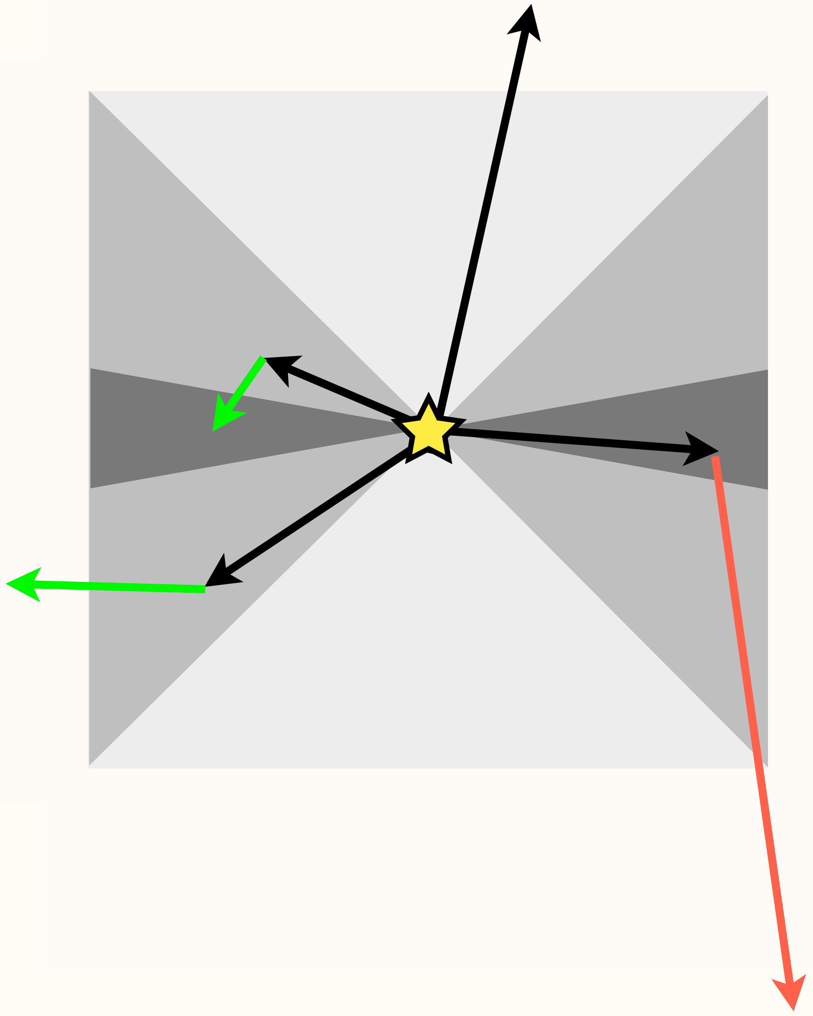
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



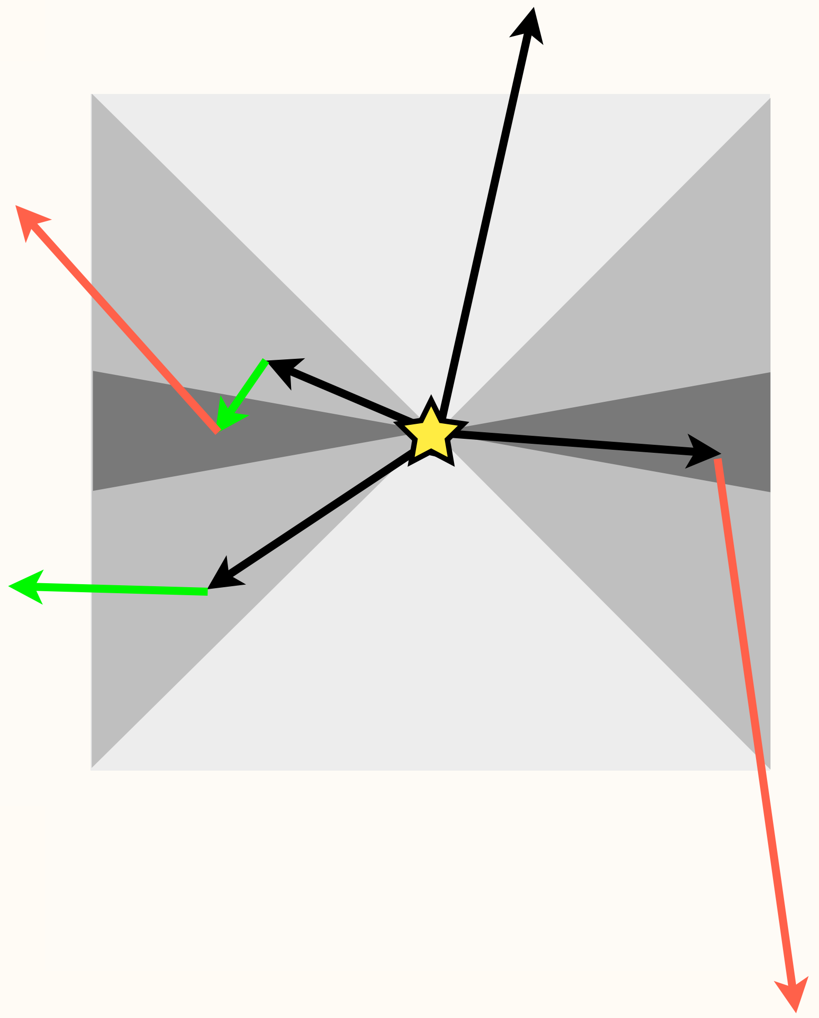
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



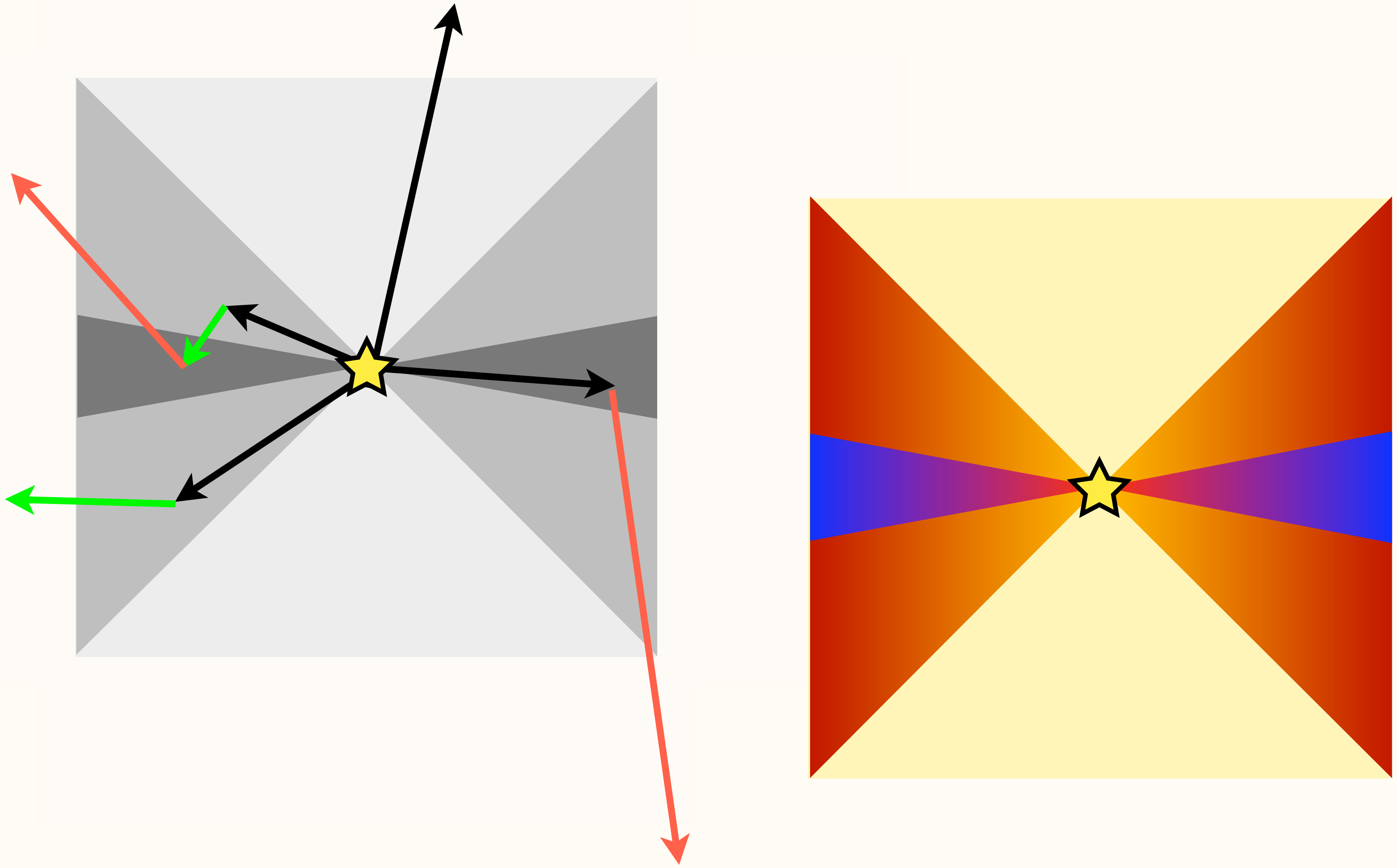
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



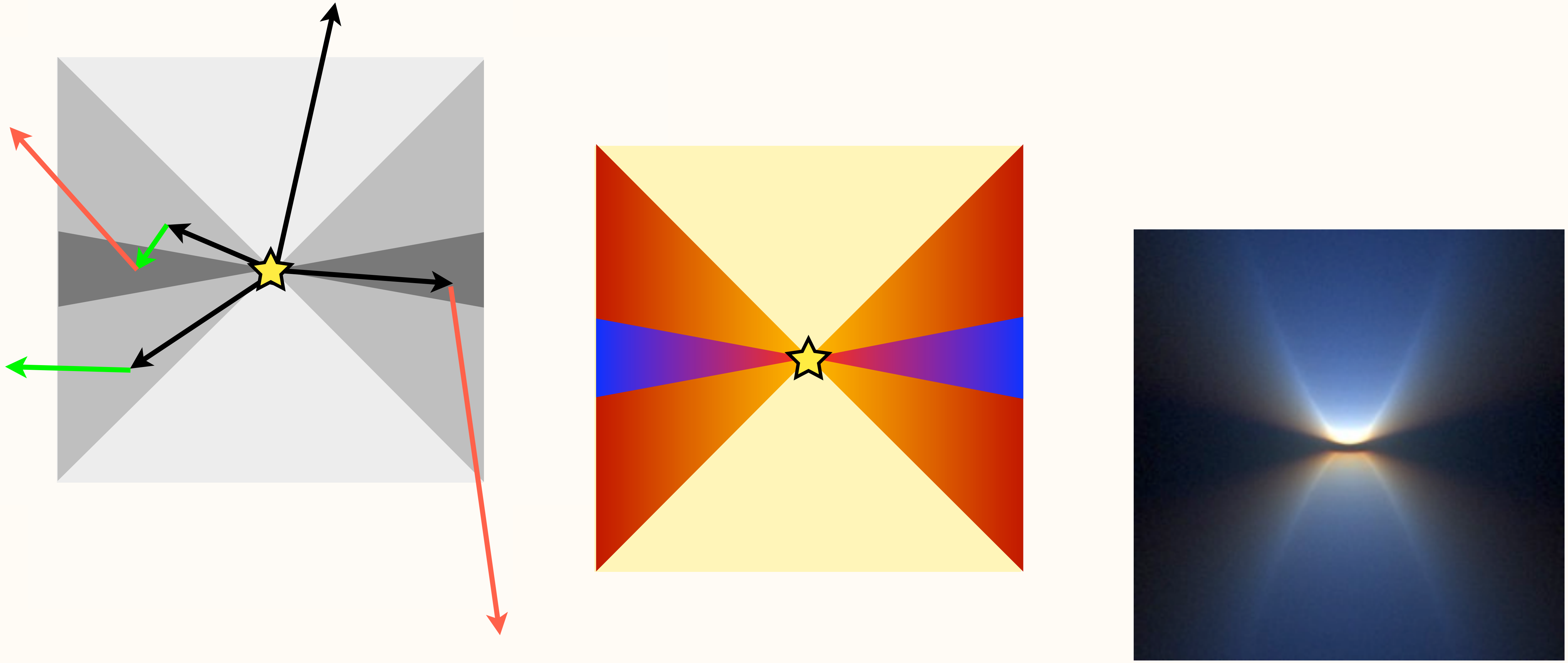
Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.

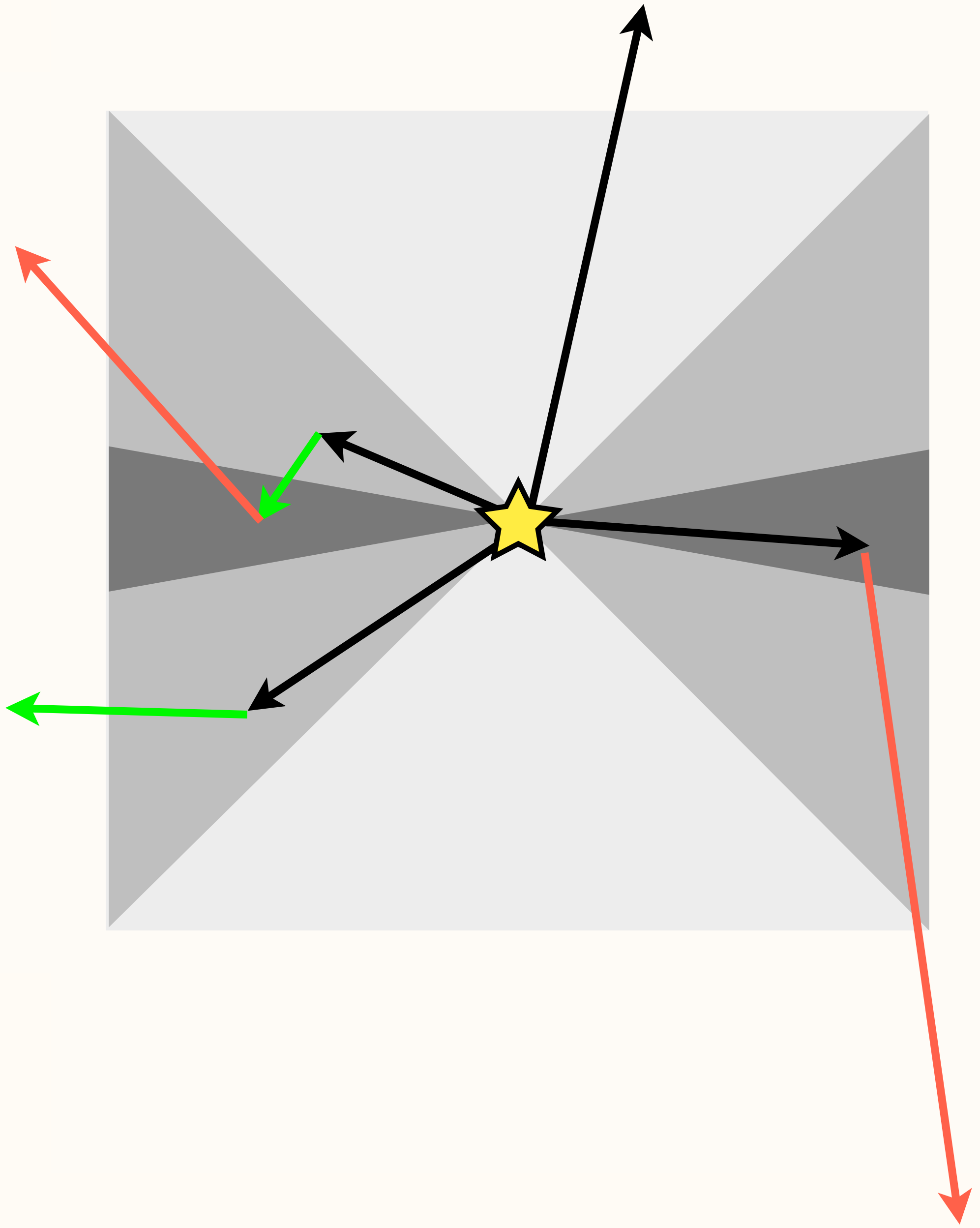


Monte-Carlo Radiative Transfer:

Propagate many photon packets by randomly sampling from probability density functions for directions, frequencies, path lengths, interactions.



Propagating photon packets on a grid

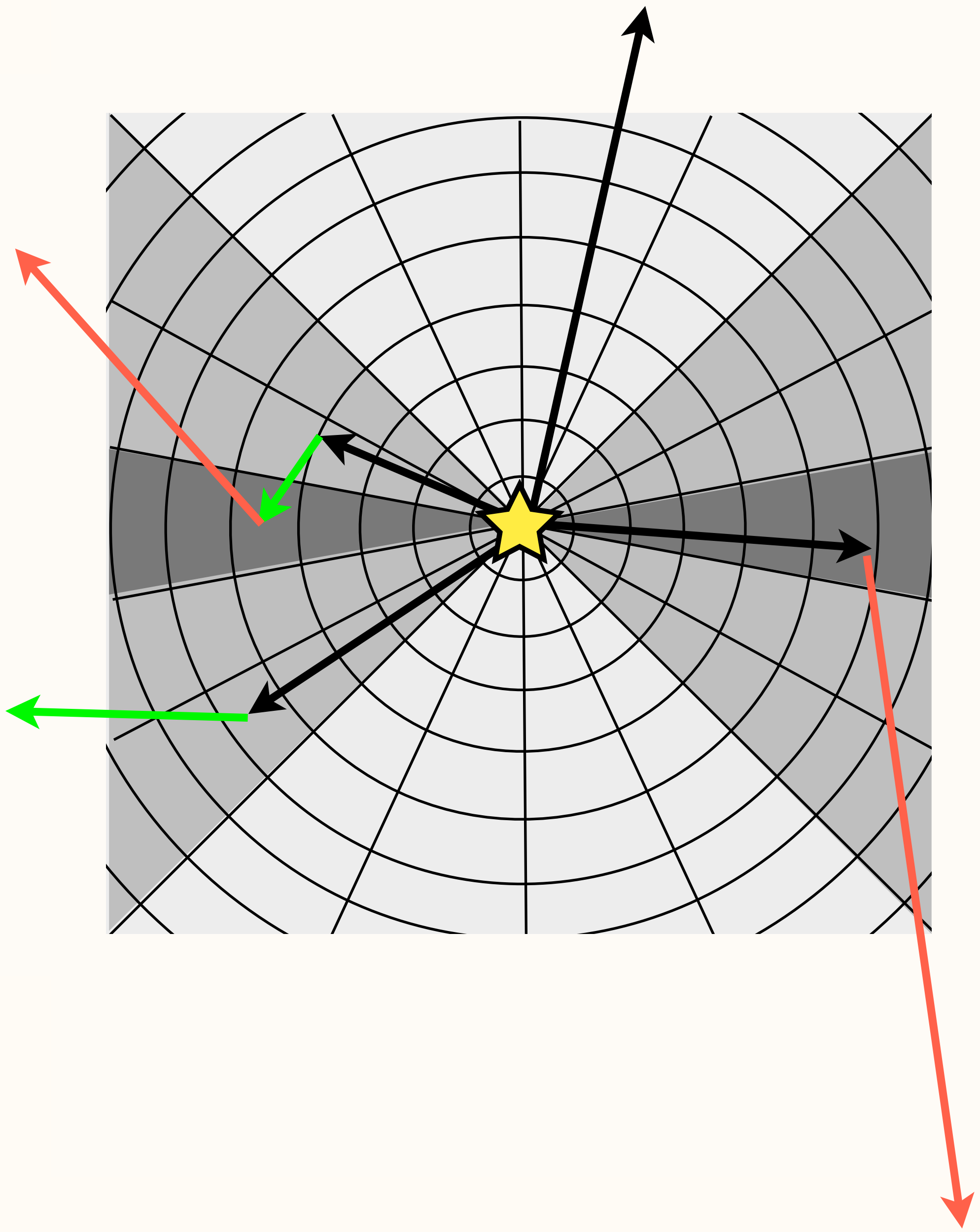


If density is constant, optical depth is proportional to distance.

For arbitrary 3-d model, not trivial!

We therefore split density structure into cells of **constant density**

Propagating photon packets on a grid

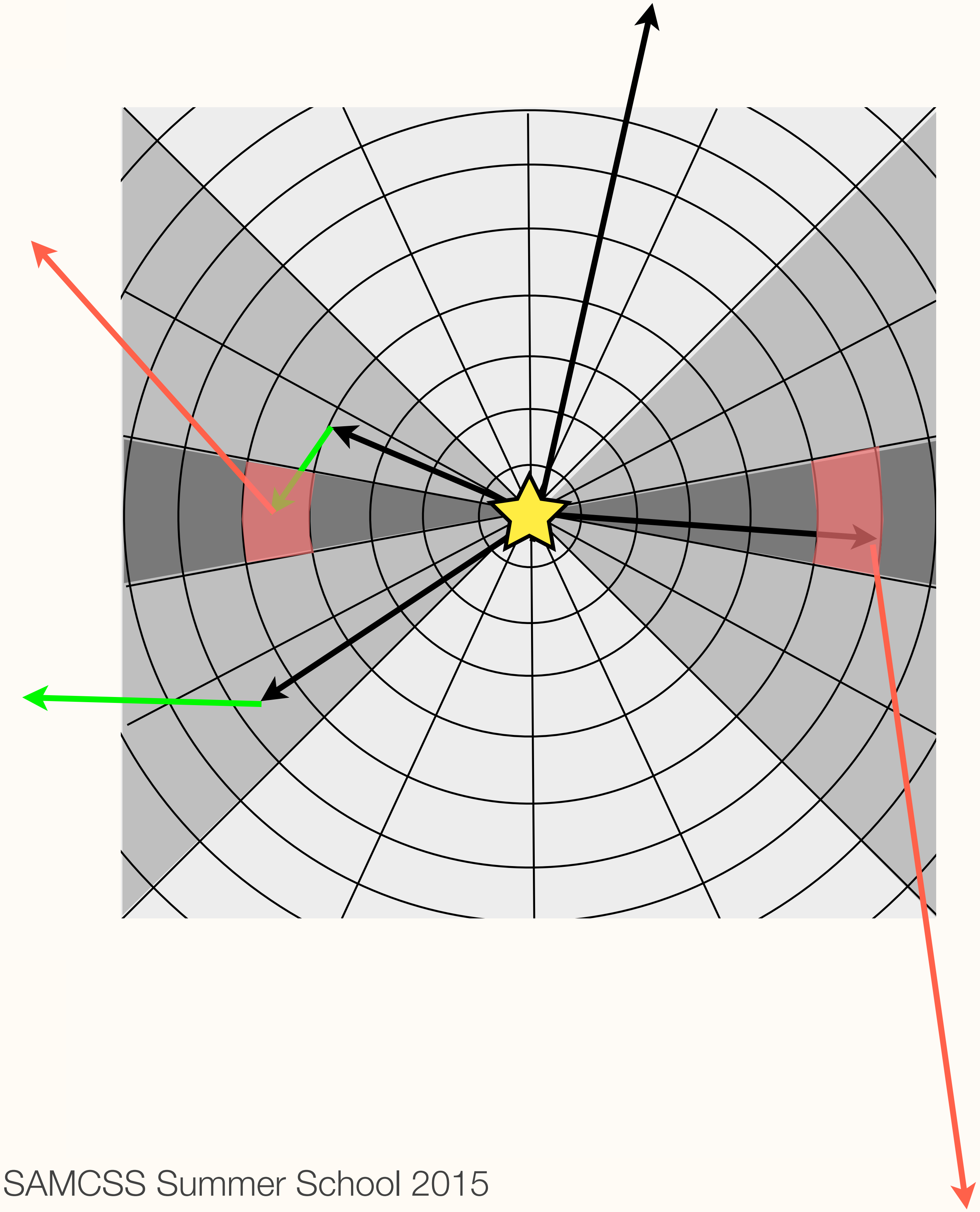


If density is constant, optical depth is proportional to distance.

For arbitrary 3-d model, not trivial!

We therefore split density structure into cells of **constant density**

Radiative Equilibrium: Computing temperatures



Naive algorithm: count energy absorbed in each cell

In equilibrium, energy emitted should be equal to energy absorbed!

Radiative Equilibrium: Computing temperatures

We can then convert specific absorbed energy rate to temperature:

$$4\,\sigma\,\kappa_{\rm P}(T)\,T^4 = \dot{A}$$

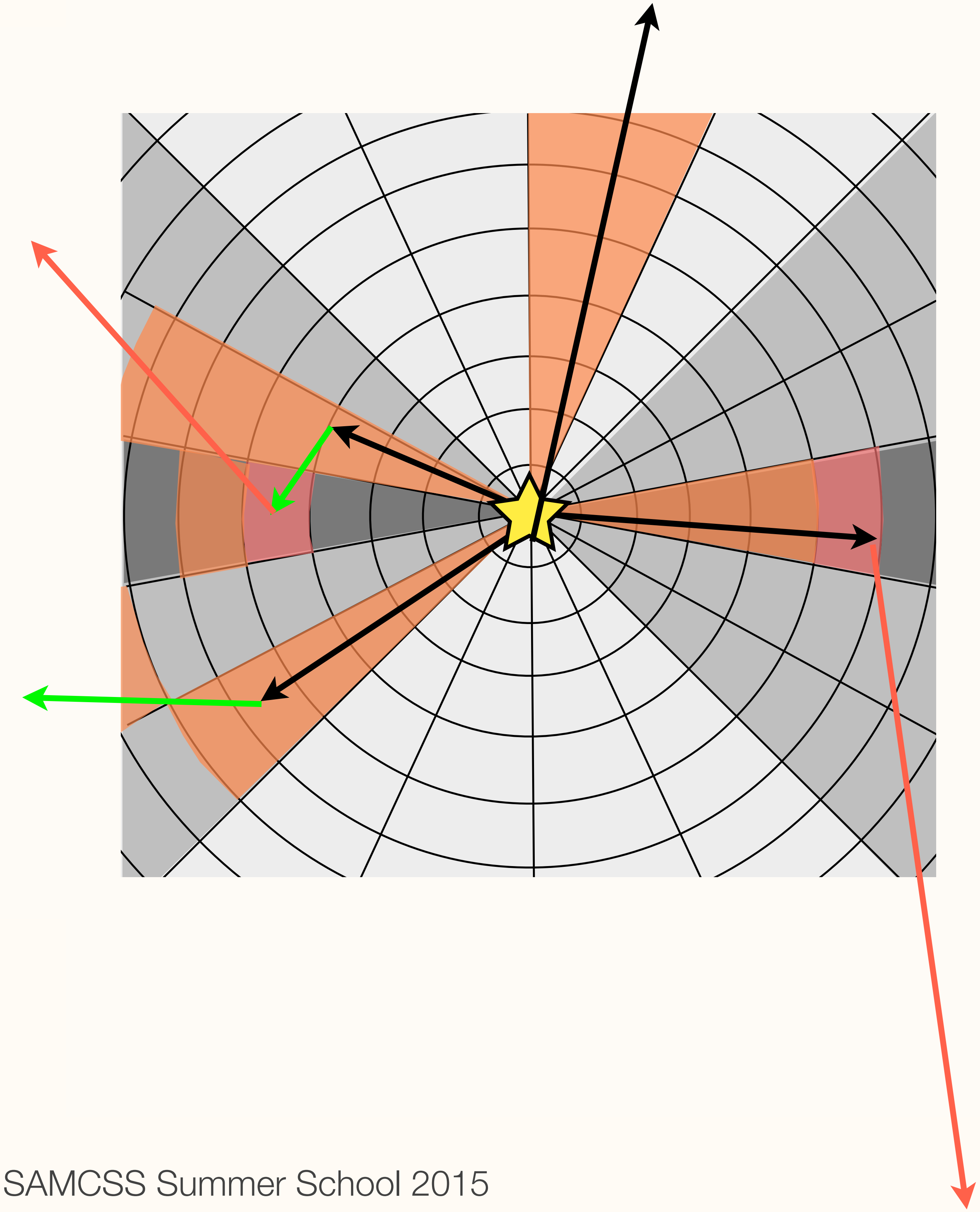
... but emissivity depends on temperature:

$$\frac{dI_{x,y,z,\theta,\phi,\nu}}{ds} = -\,\alpha_{x,y,z,\nu} I_{x,y,z,\theta,\phi,\nu} + j_{x,y,z,\theta,\phi,\nu}(T)$$

so once we have computed temperature, need to start over until temperature converges!

Alternatively, adjust emissivity on-the-fly (Bjorkman & Wood, 2001)

Radiative Equilibrium: Computing temperatures

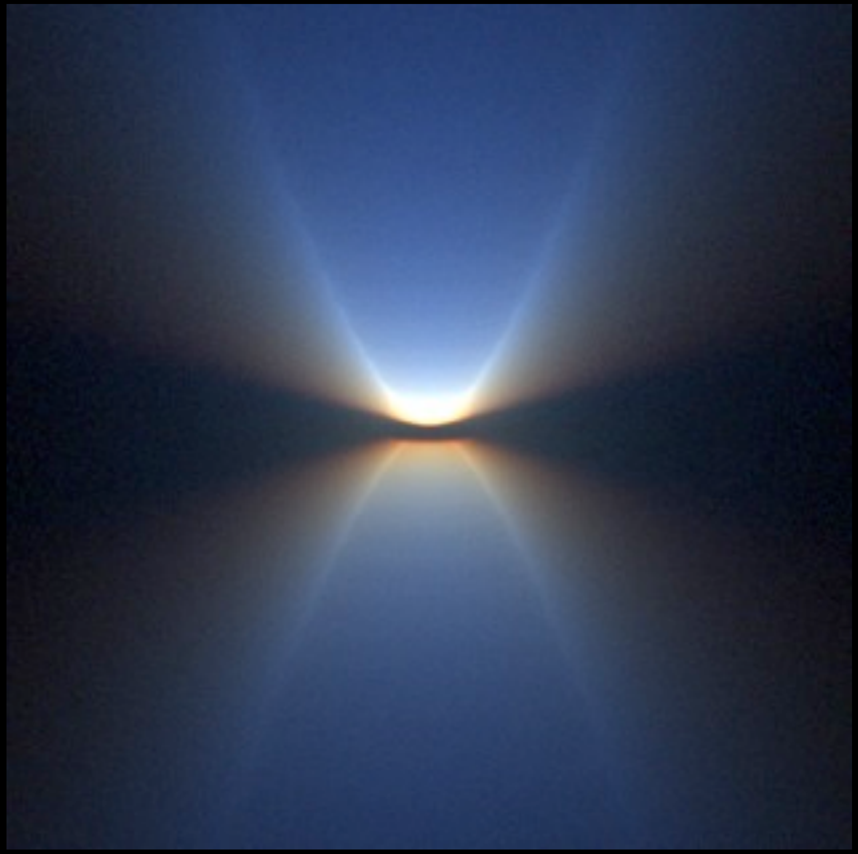


Lucy (1999) algorithm: count **possible** energy absorbed in each cell

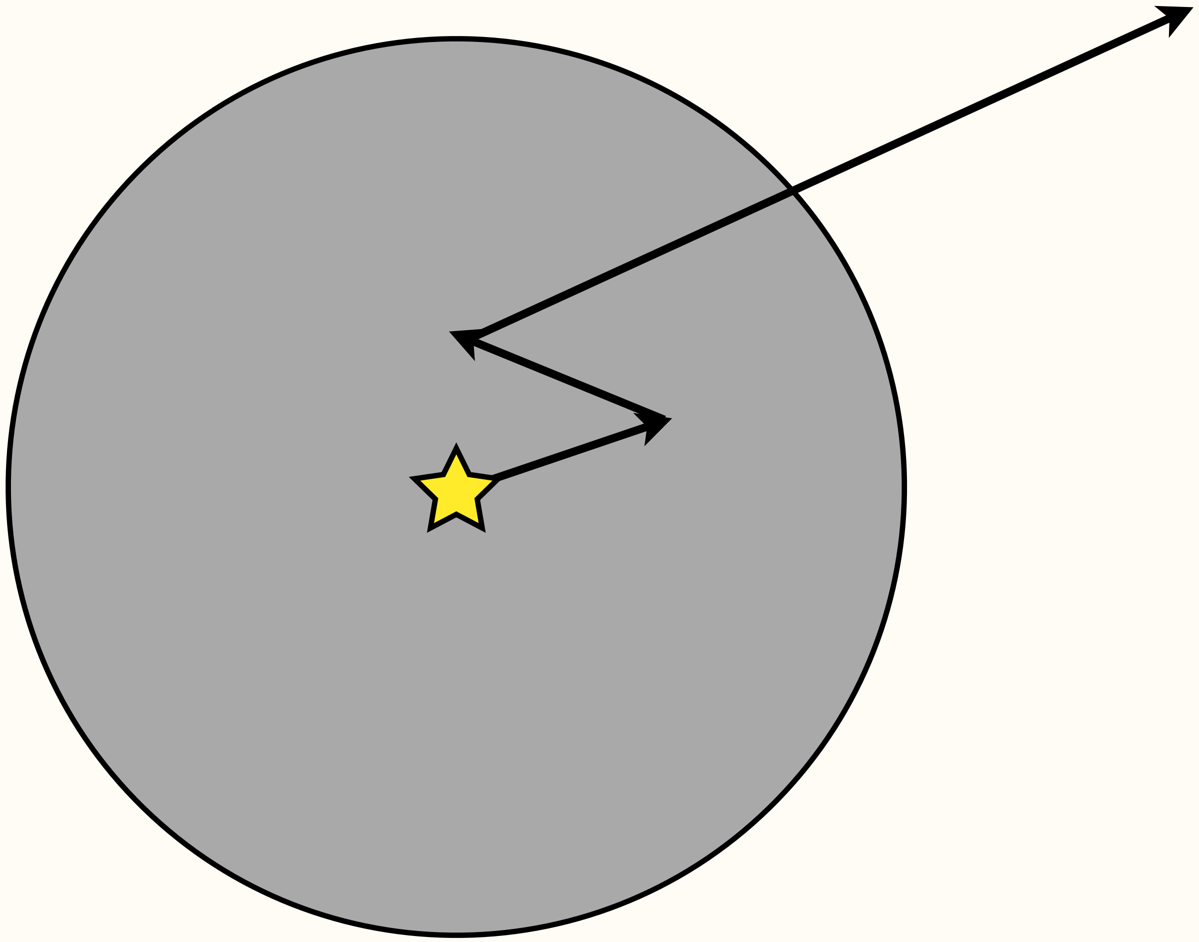
= P(absorption) x E_photon

Every cell crossing counts

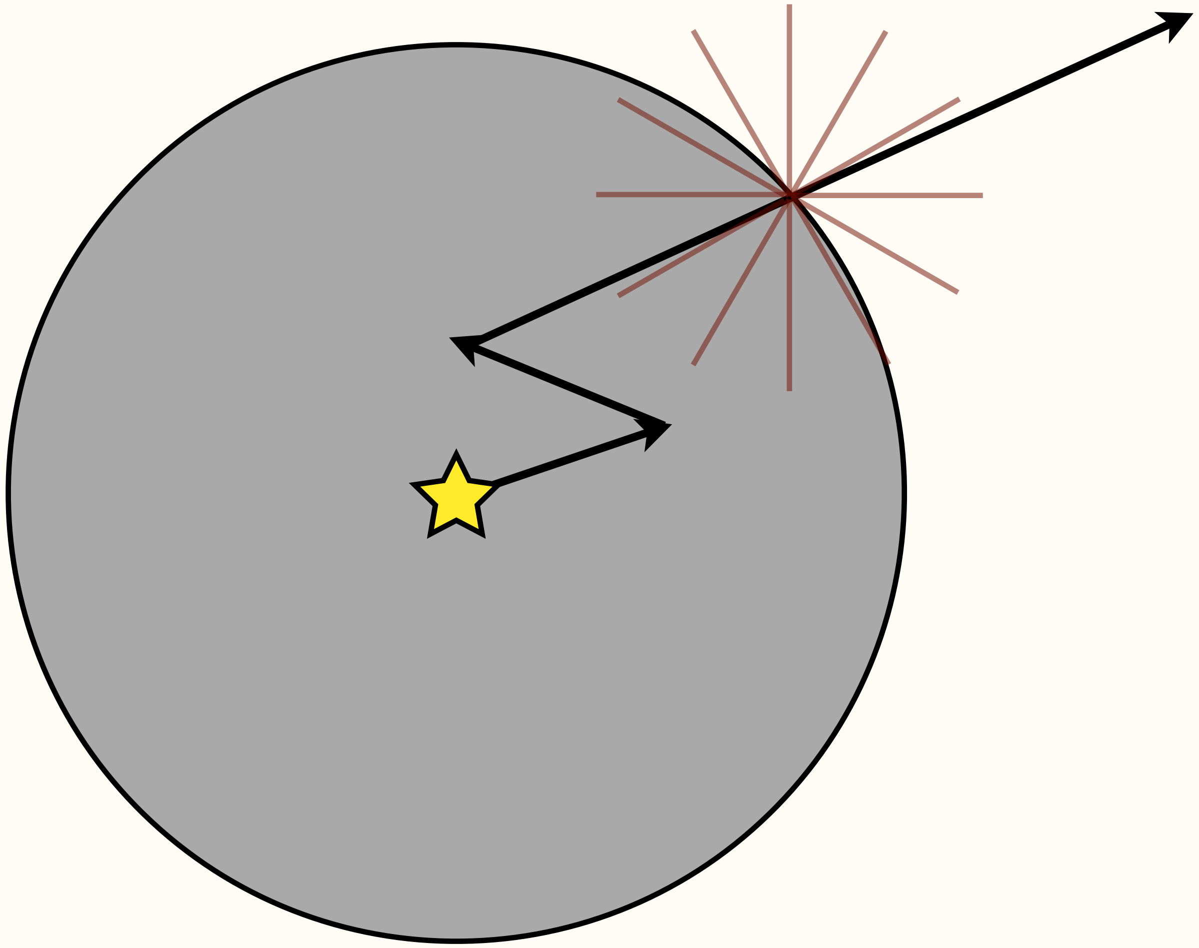
Computing SEDs/Images



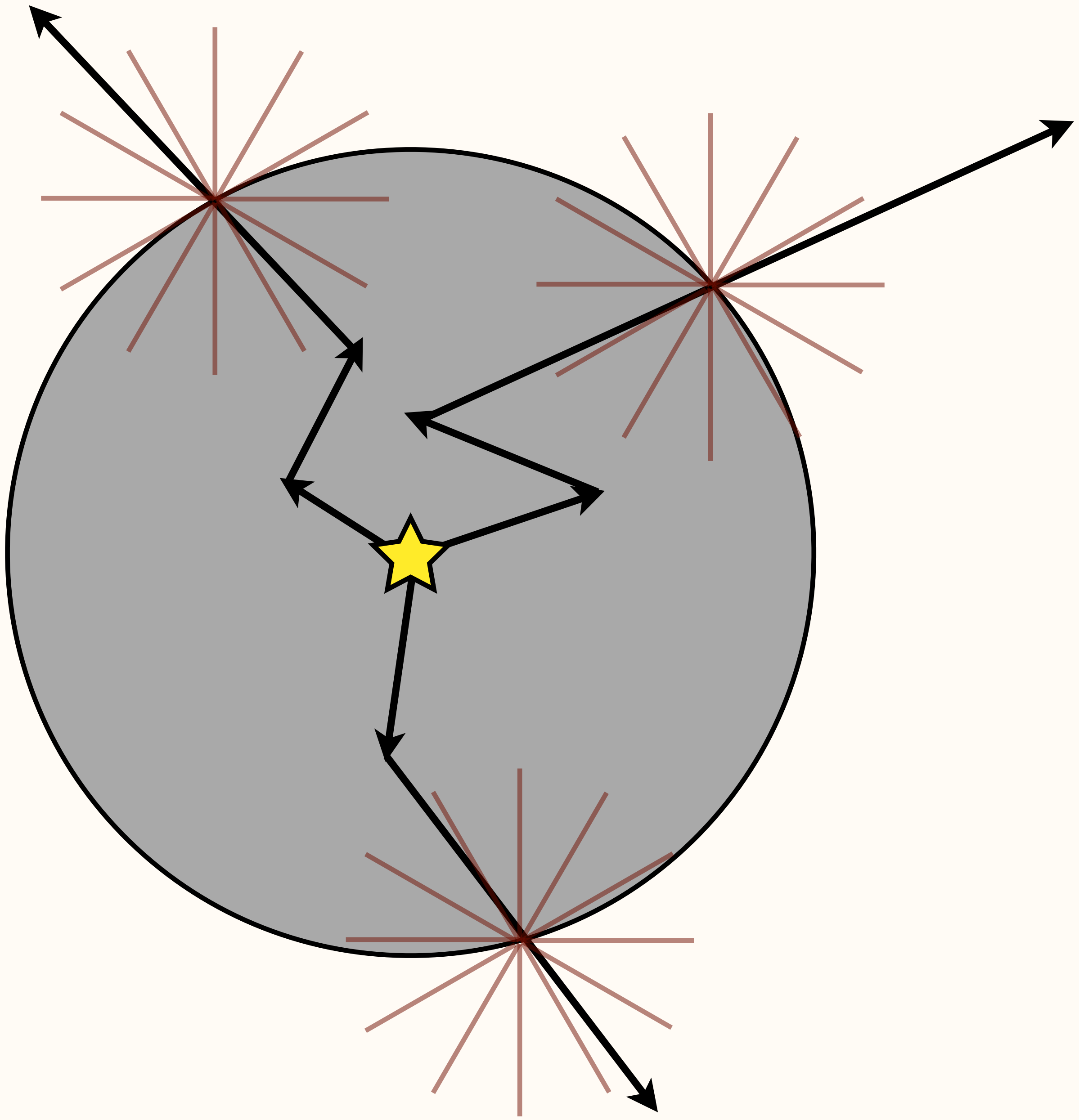
Photon binning



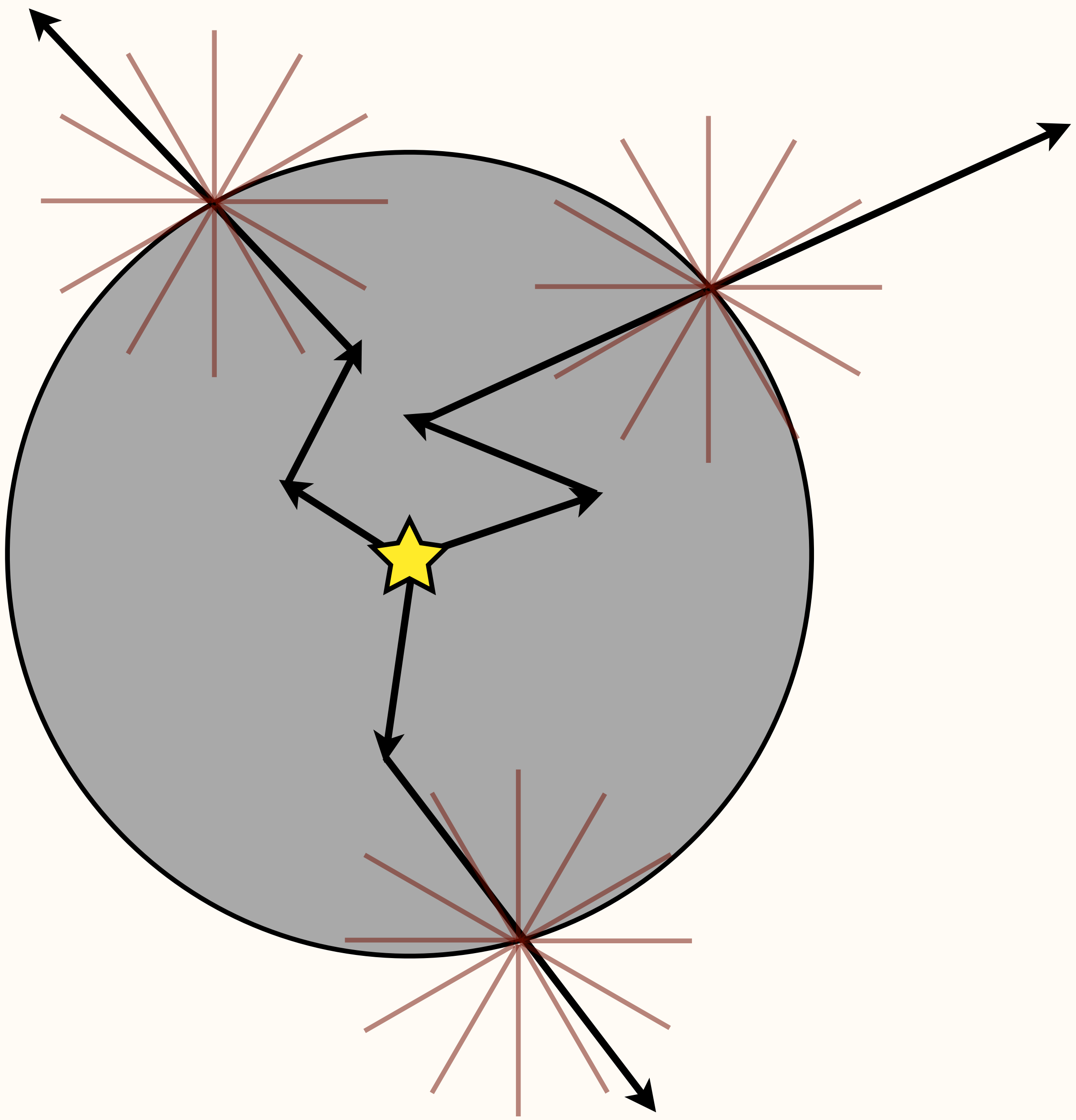
Photon binning



Photon binning

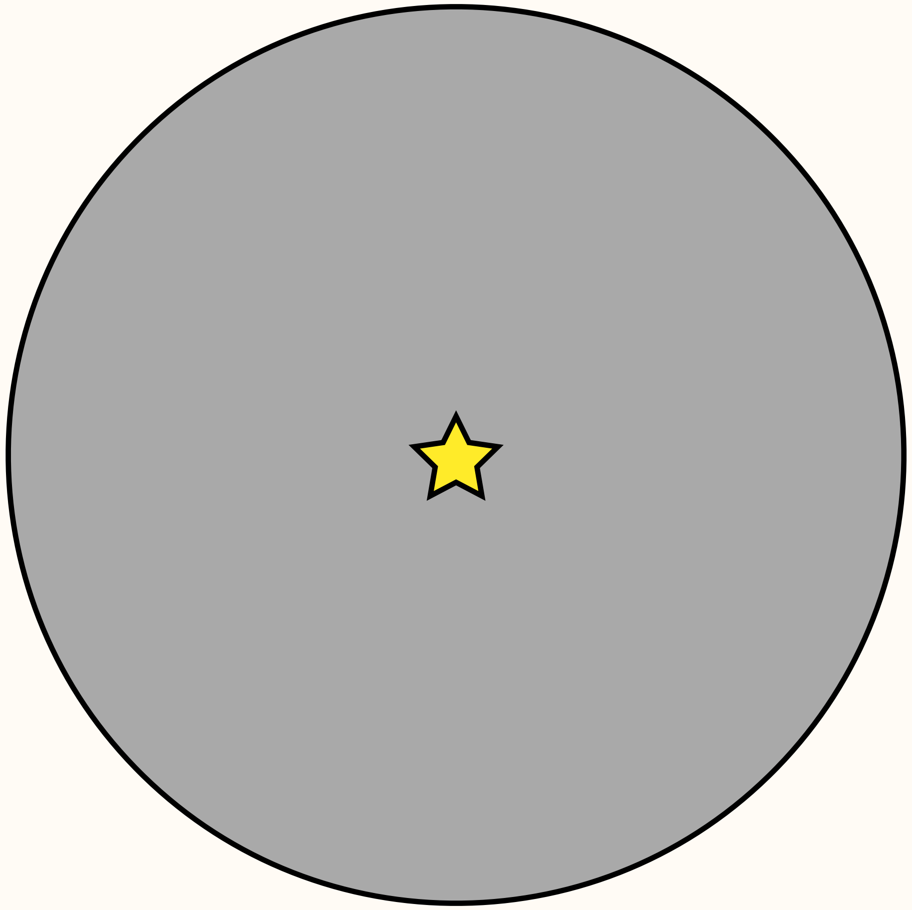


Photon binning

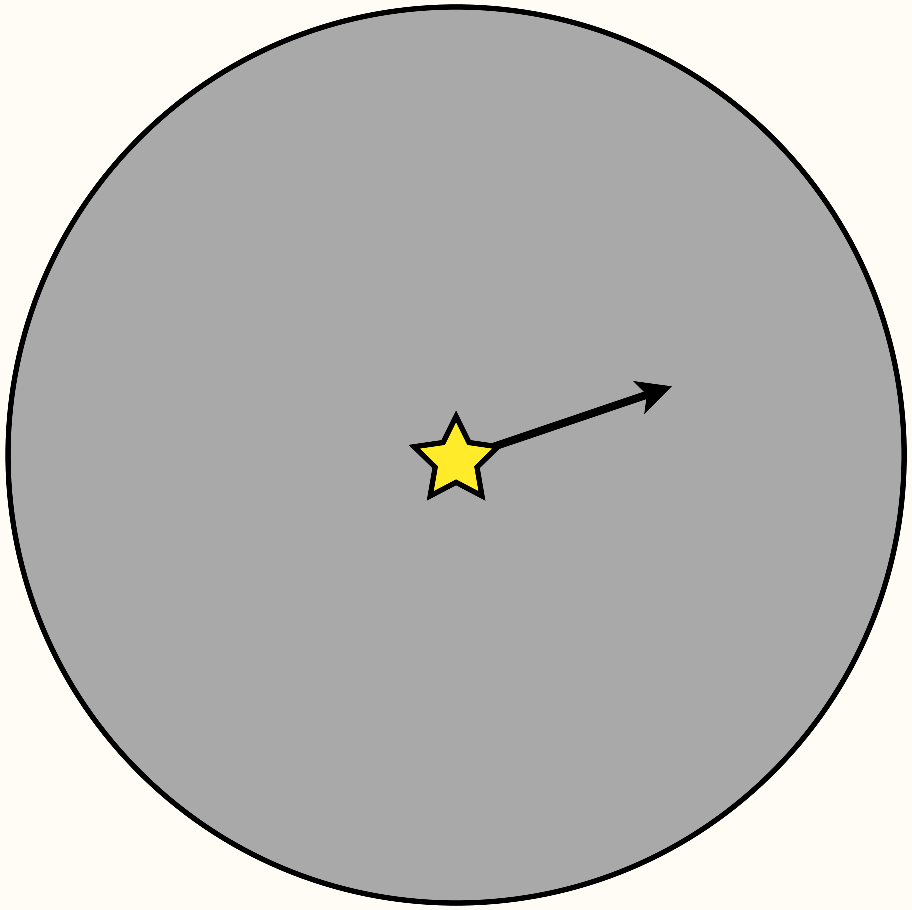


1 photon contributes to a single pixel, wavelength, and viewing angle

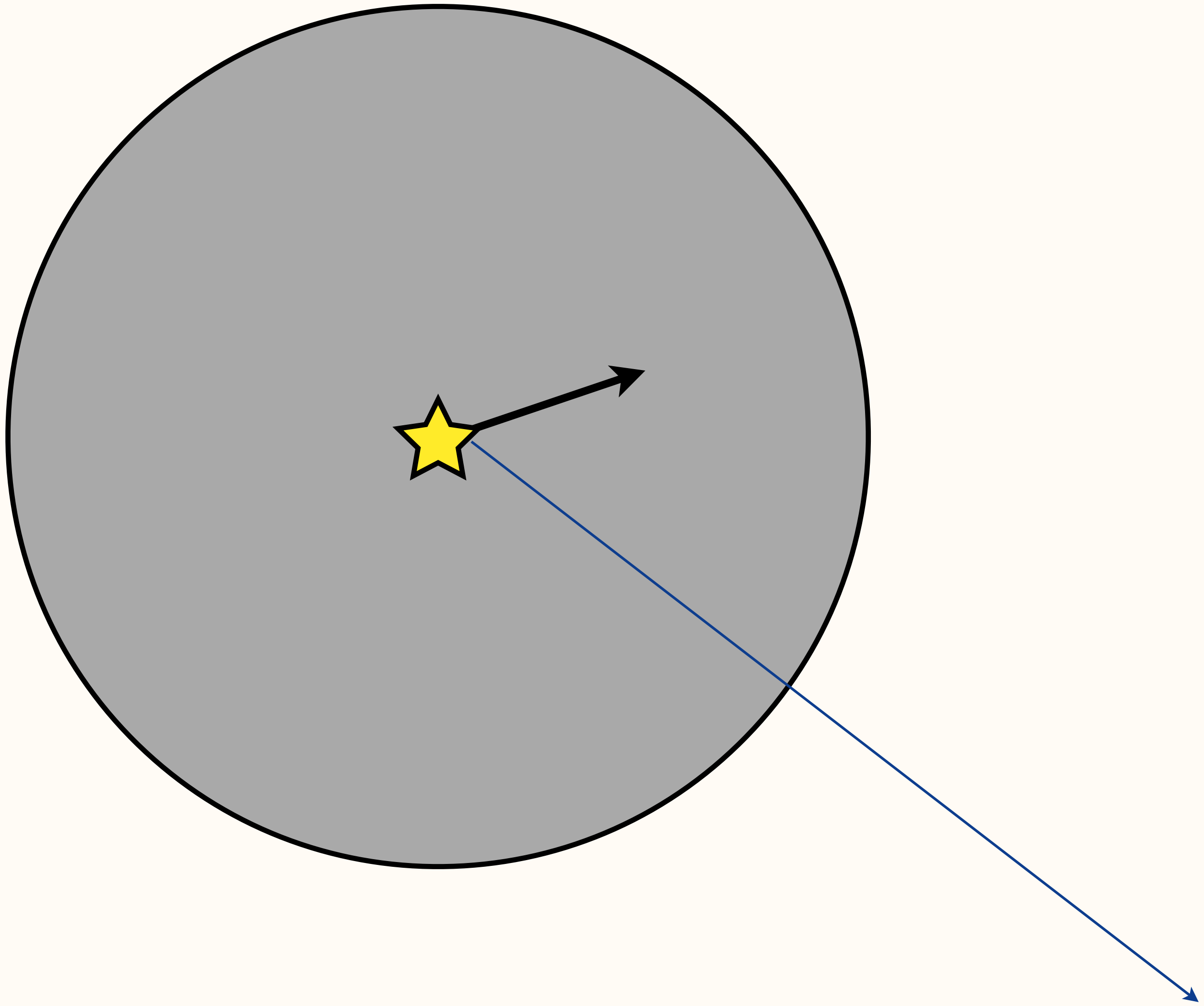
“Peeling-off”



“Peeling-off”

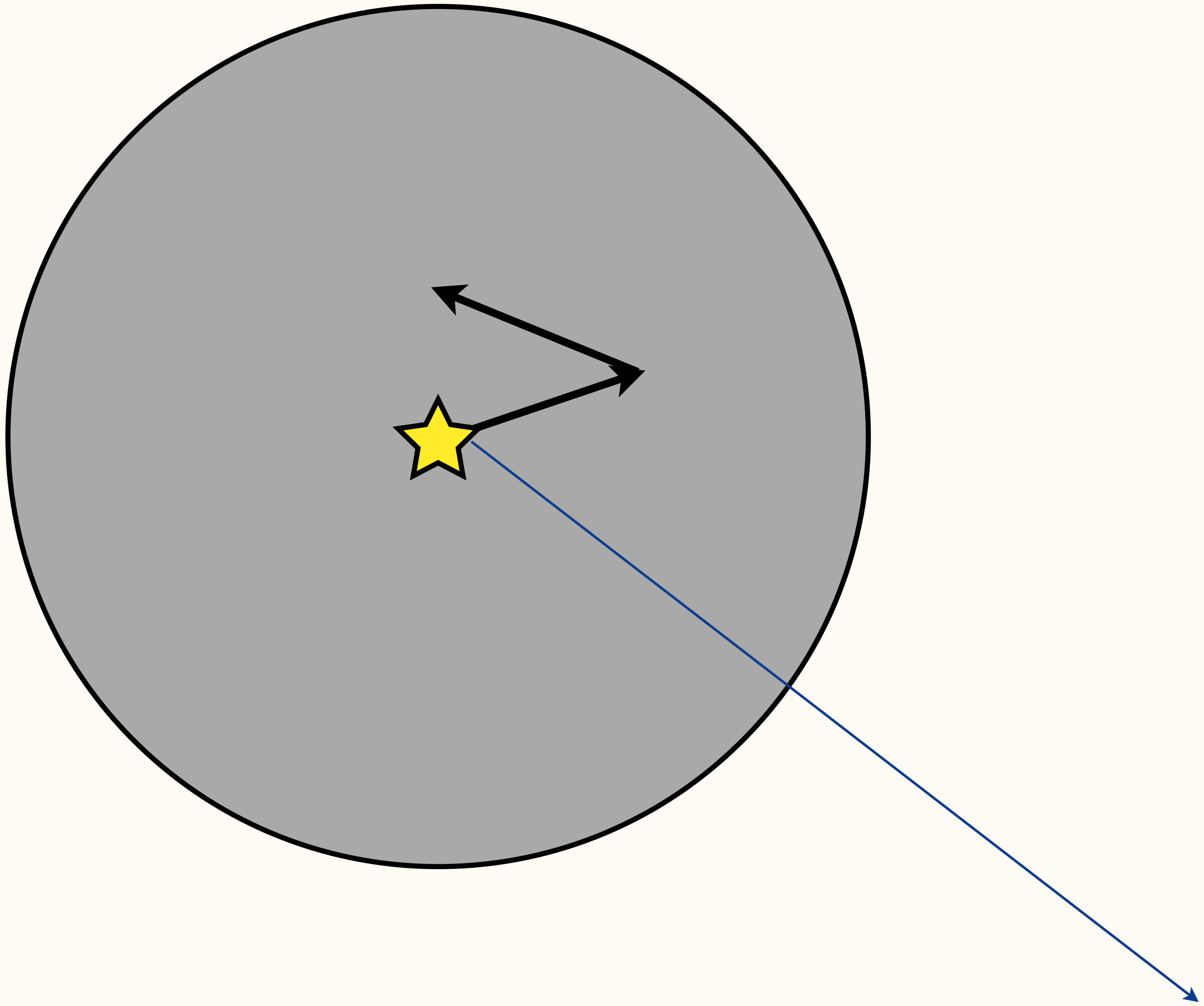


“Peeling-off”



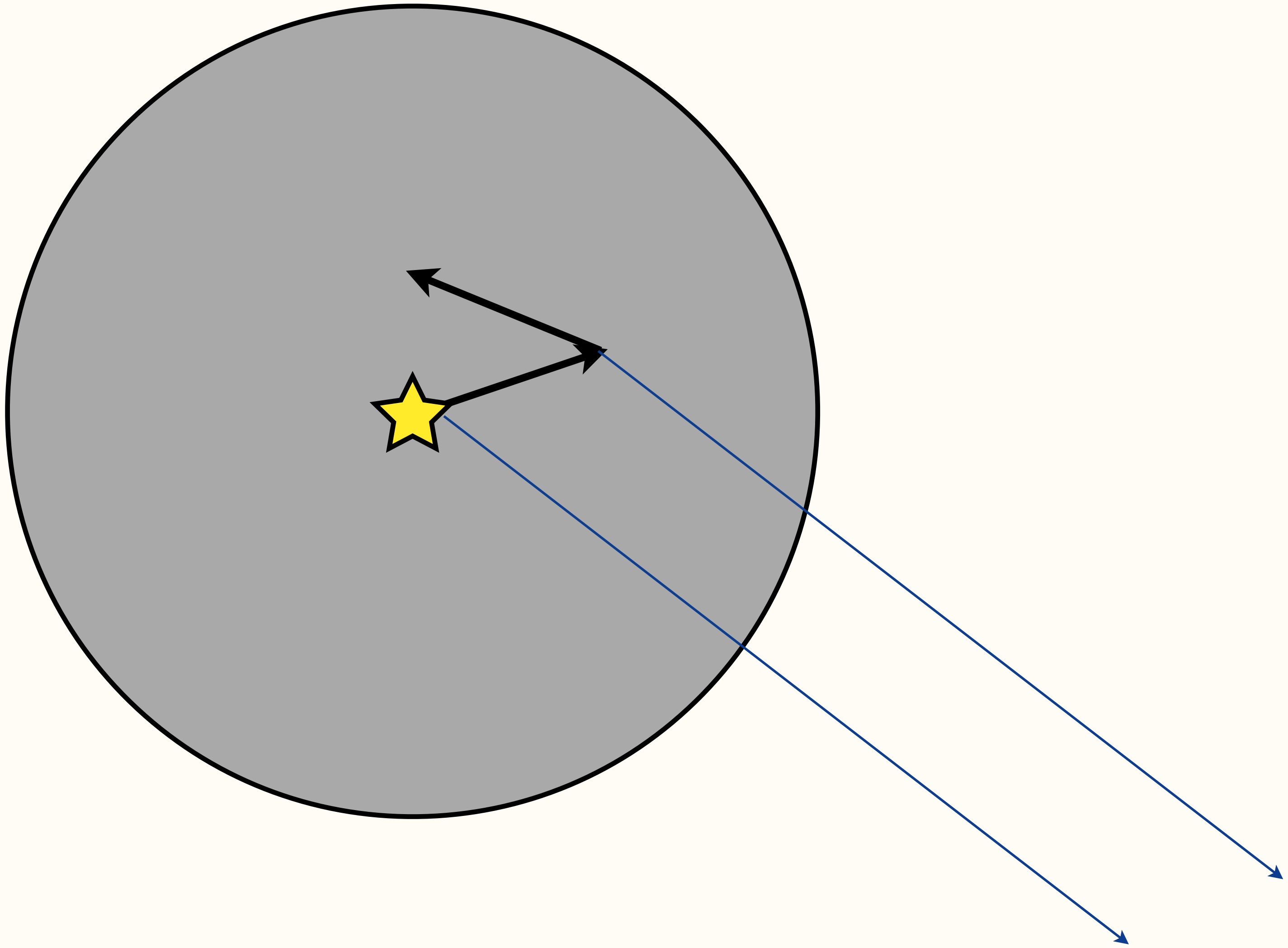
$$W = P(\theta, \phi) \cdot e^{-\tau_\mathrm{escape}}$$

“Peeling-off”



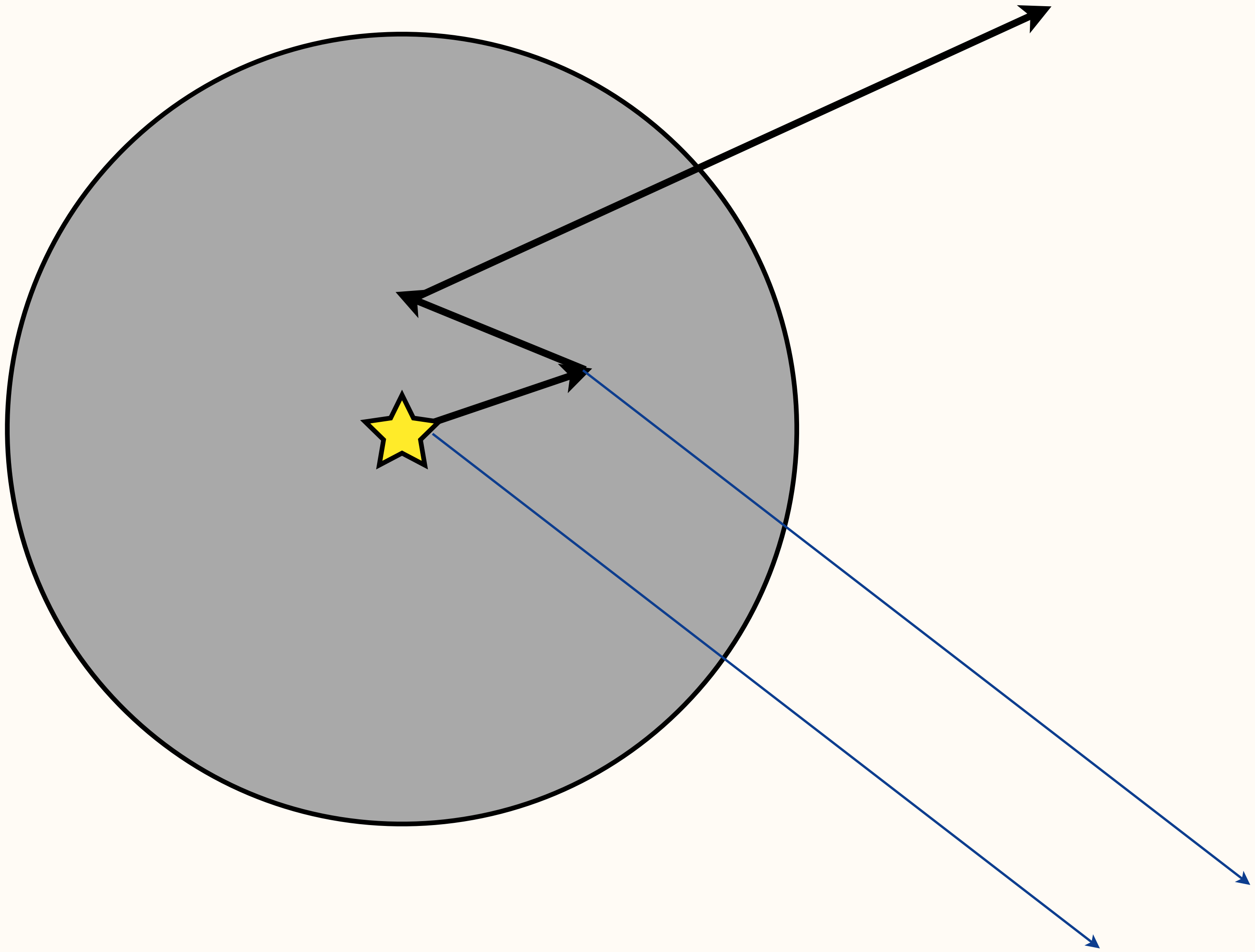
$$W = P(\theta, \phi) \cdot e^{-\tau_{\rm escape}}$$

“Peeling-off”



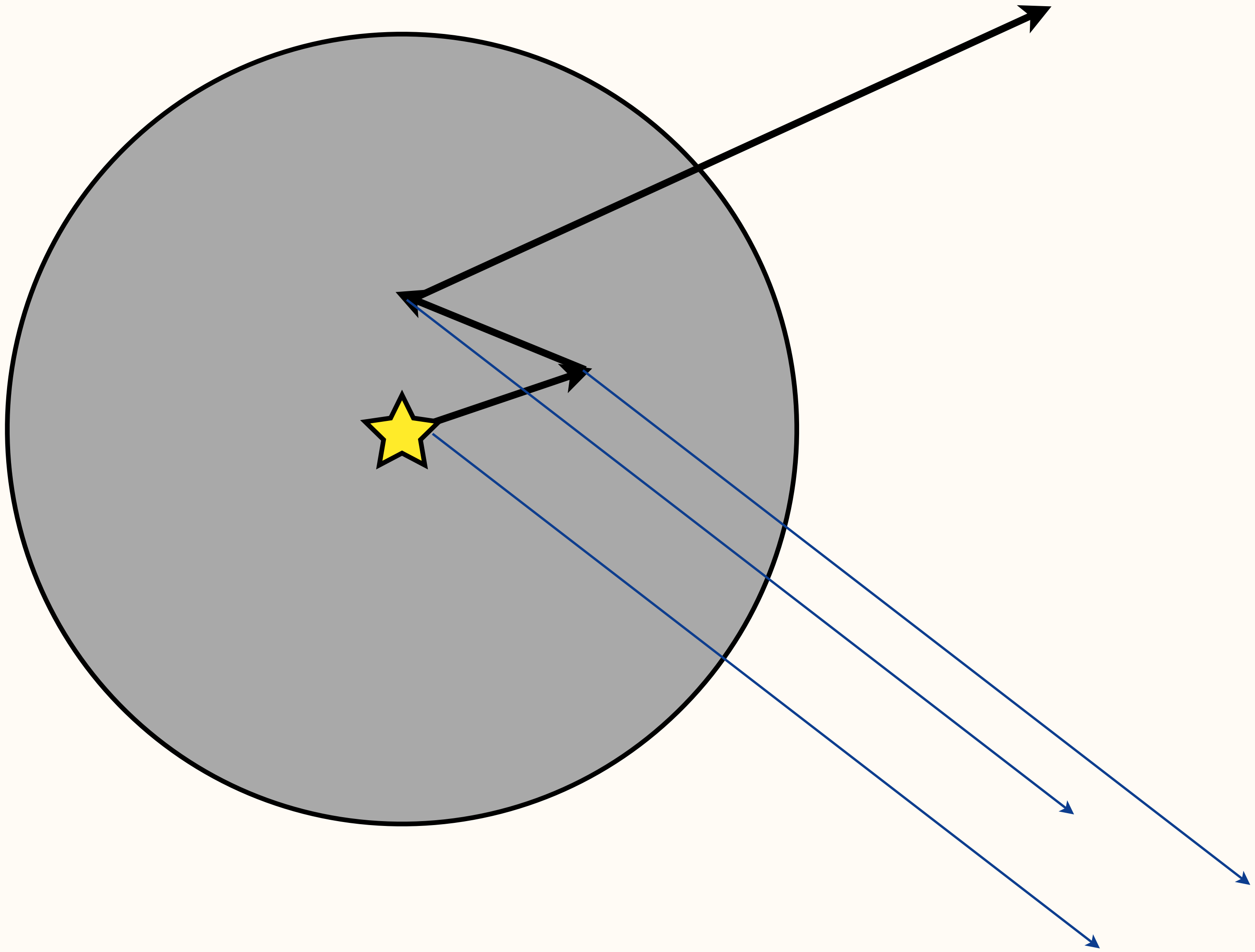
$$W = P(\theta, \phi) \cdot e^{-\tau_{\rm escape}}$$

“Peeling-off”



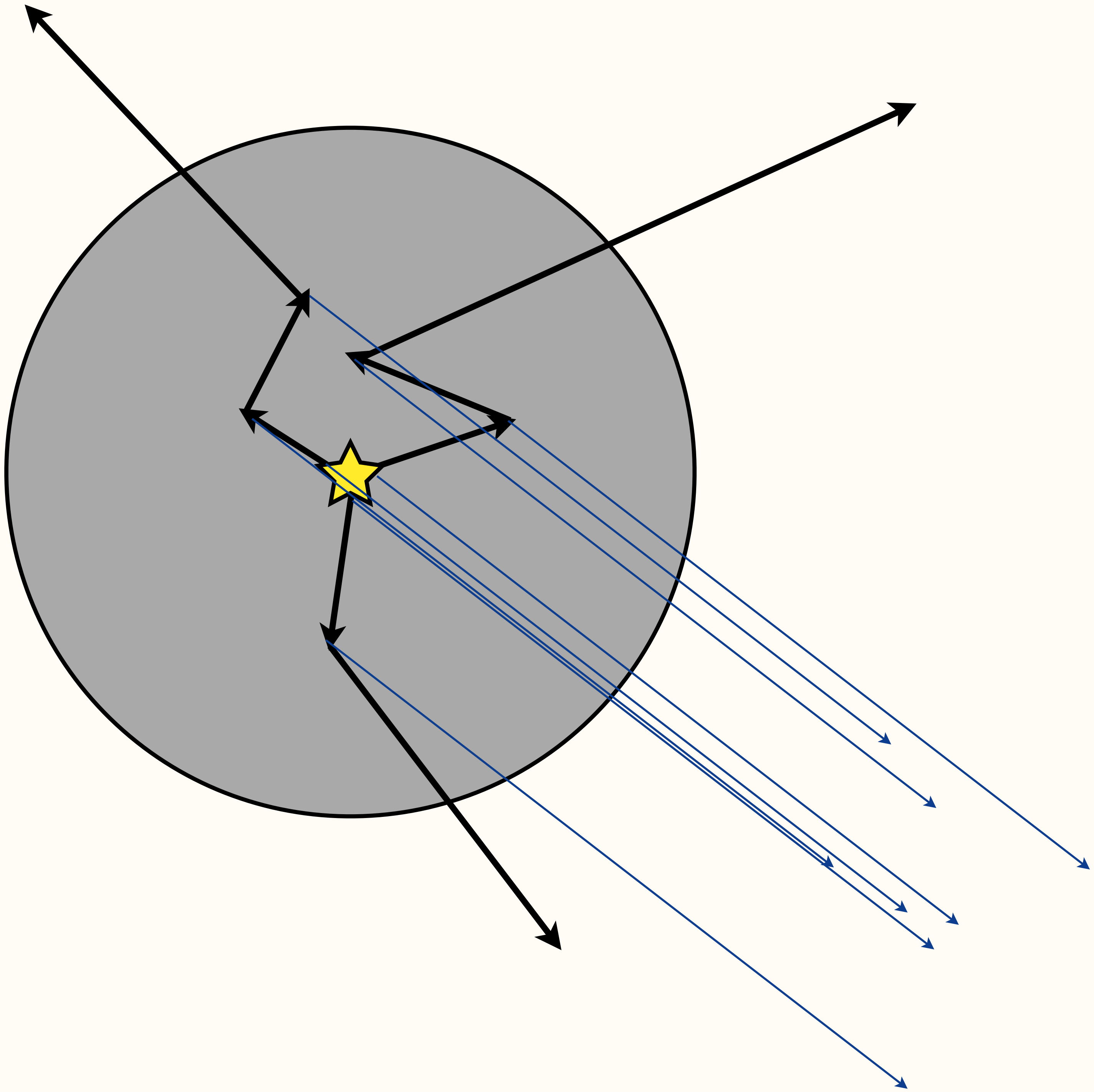
$$W = P(\theta, \phi) \cdot e^{-\tau_{\mathrm{escape}}}$$

“Peeling-off”

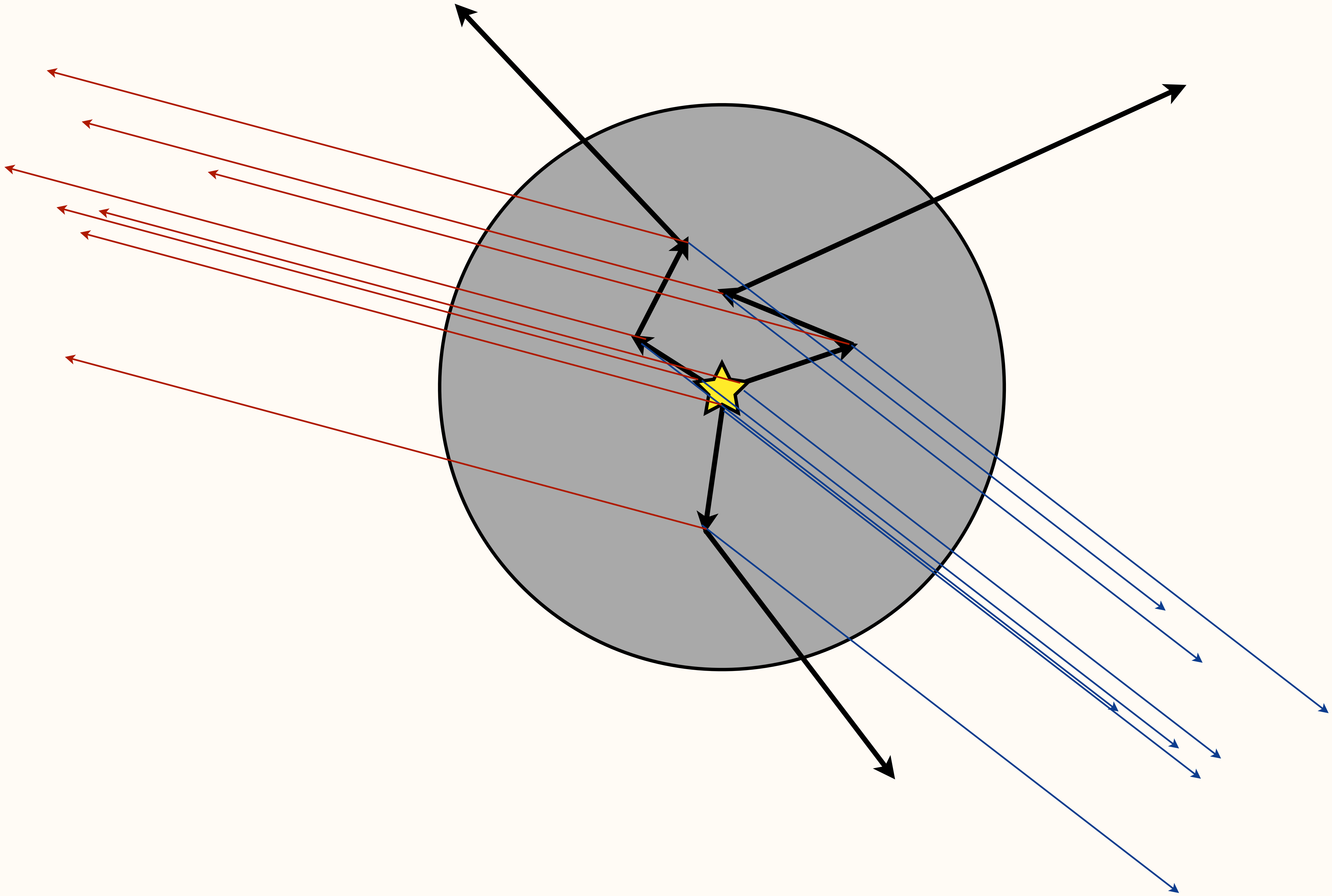


$$W = P(\theta, \phi) \cdot e^{-\tau_{\mathrm{escape}}}$$

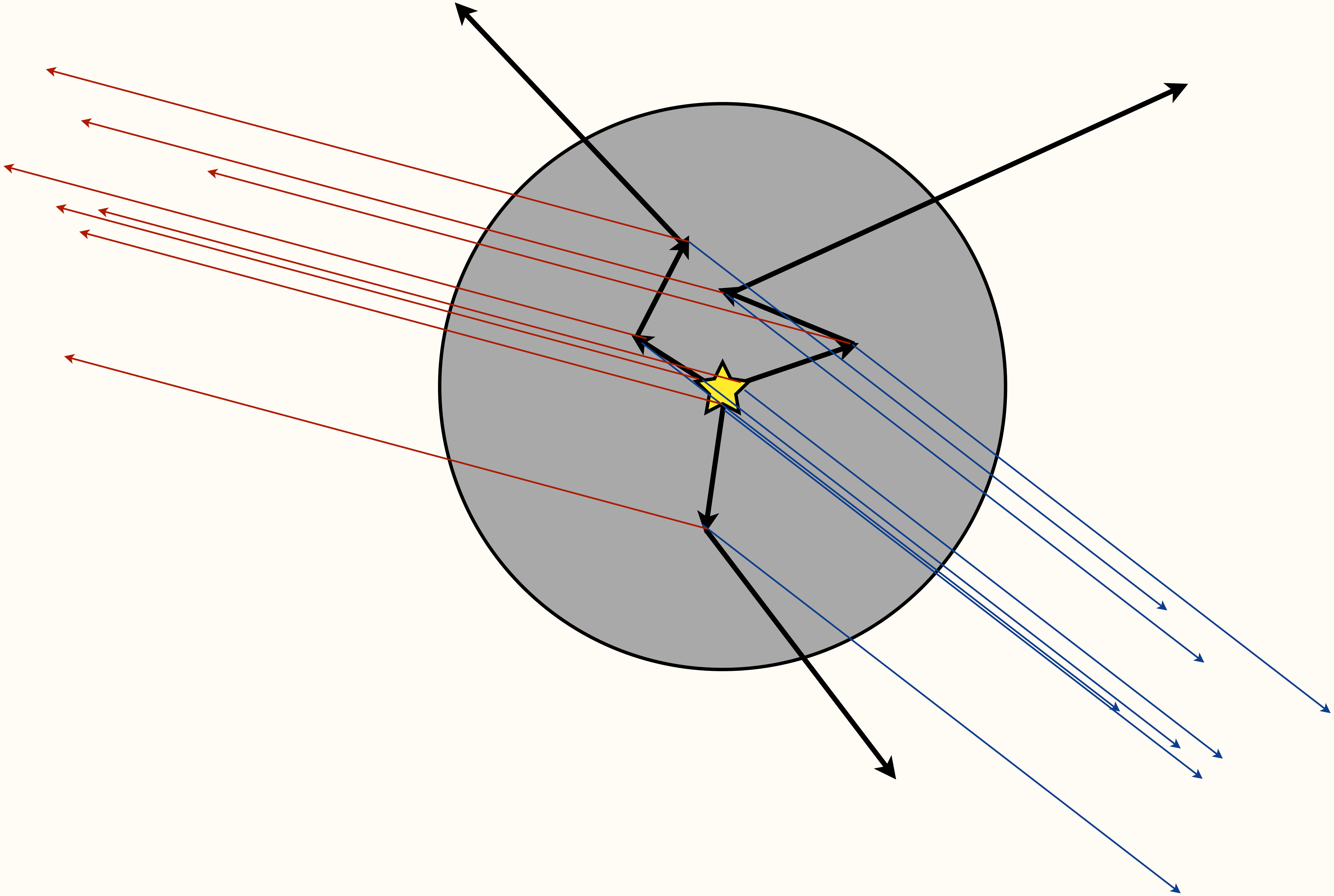
“Peeling-off”



“Peeling-off”



“Peeling-off”

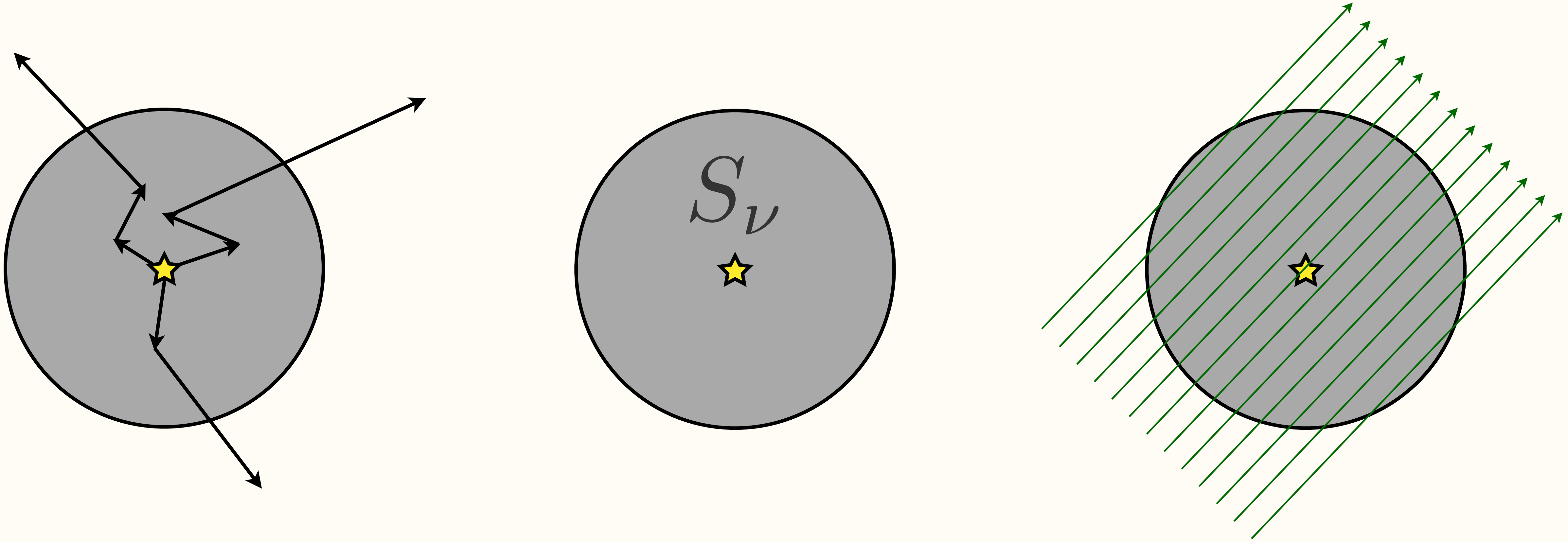


Every photon contributes multiple times to all viewing angles

SAMCSS Summer School 2015

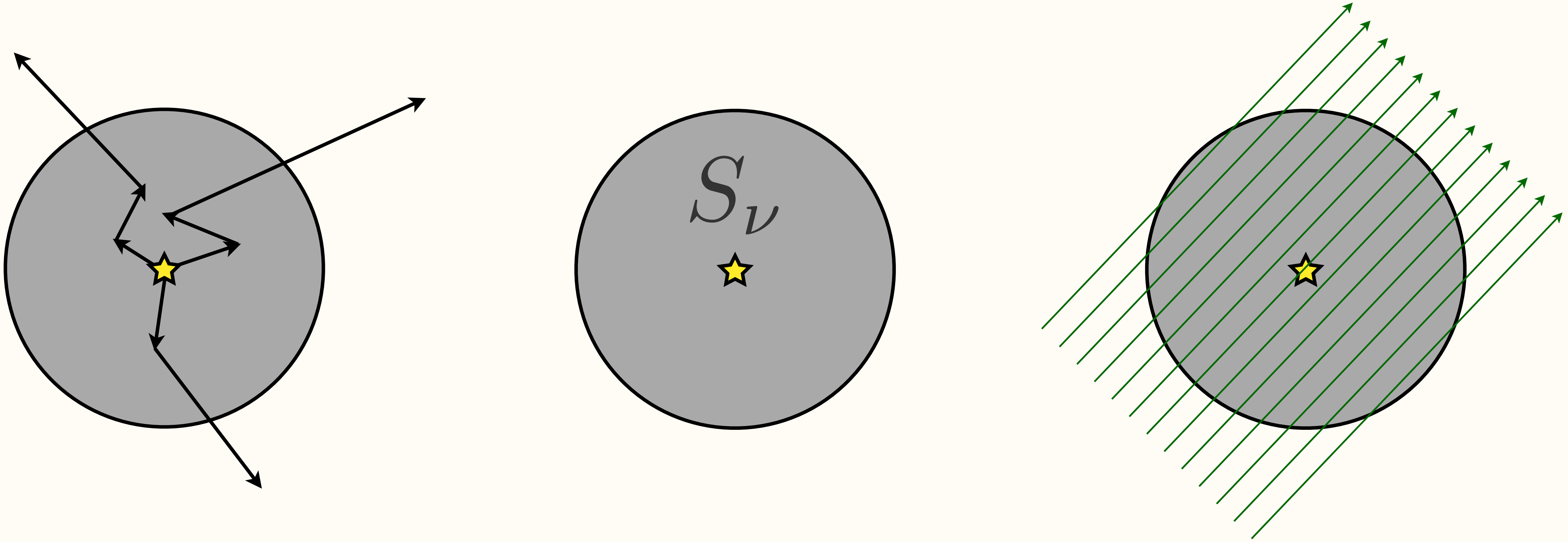
Thomas Robitaille

“Raytracing”



$$I_\nu(\tau_\nu) = I_\nu(0)e^{-\tau_\nu} + \int_0^{\tau_\nu} S_\nu(t_\nu)e^{-(\tau_\nu - t_\nu)}dt_\nu$$

“Raytracing”



$$I_\nu(\tau_\nu) = I_\nu(0)e^{-\tau_\nu} + \int_0^{\tau_\nu} S_\nu(t_\nu)e^{-(\tau_\nu - t_\nu)}dt_\nu$$

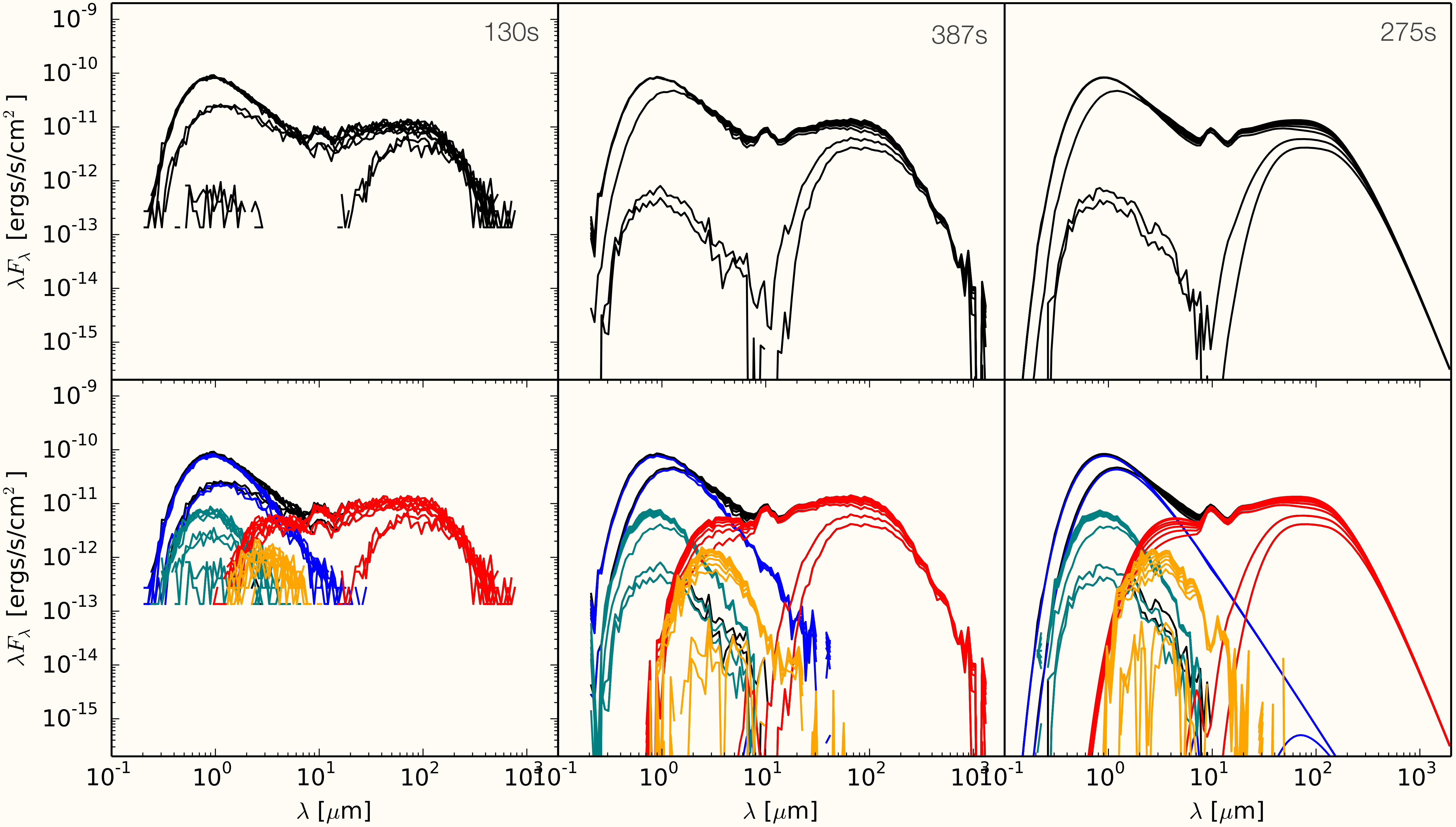
Can solve all wavelengths at the same time, very efficient!

“Raytracing”

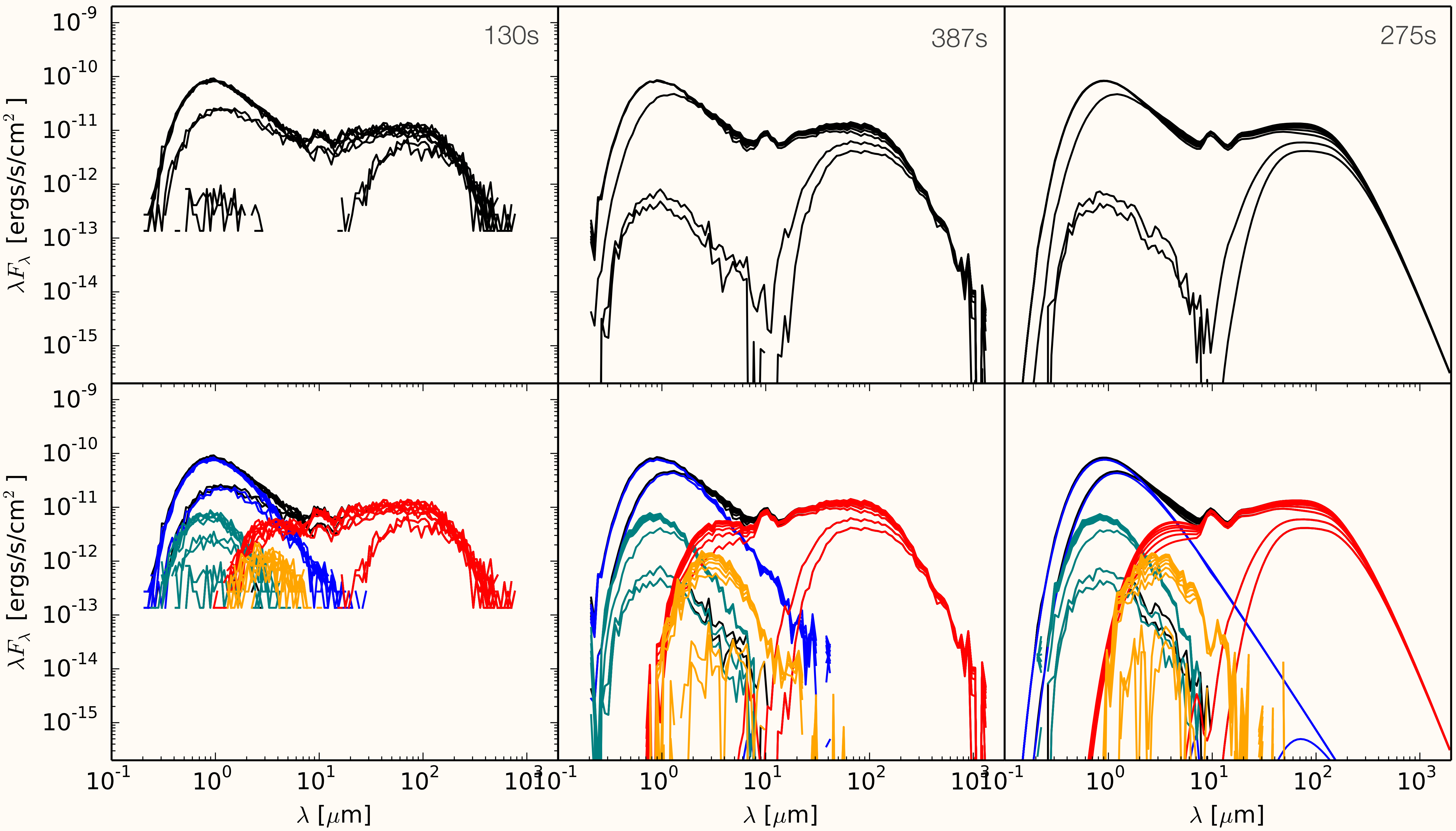
The downside of raytracing is that in the case of non-isotropic scattering, the source function can take up a LOT of memory:

$$S_\nu(\theta, \phi, x, y, z)$$

But for source and thermal emission, and for isotropic scattering, raytracing is **very** efficient.

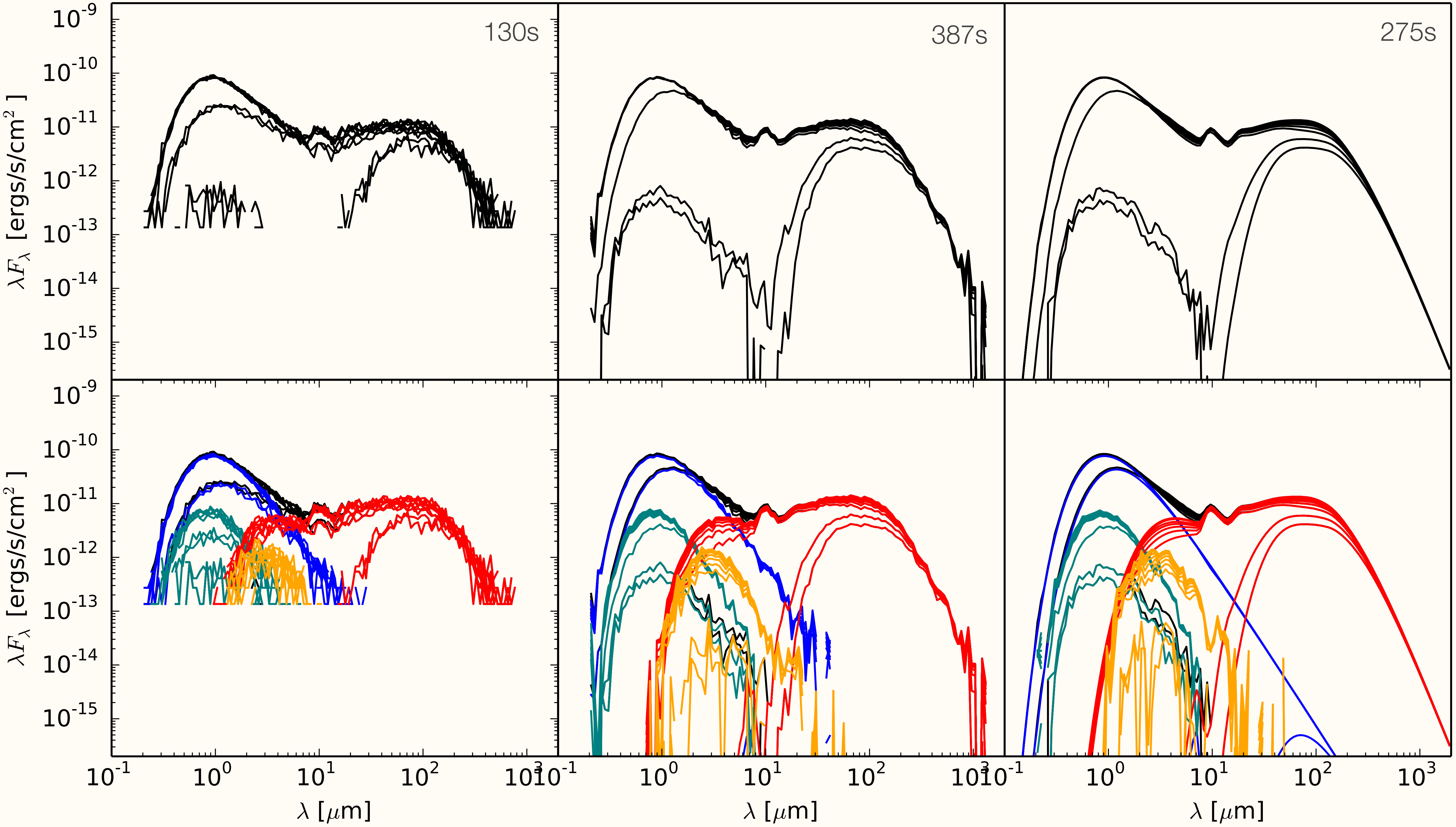

130s
387s
275s
λF_λ [ergs/s/cm^2]
10^{-9}
10^{-10}
10^{-11}
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
λ [μm]

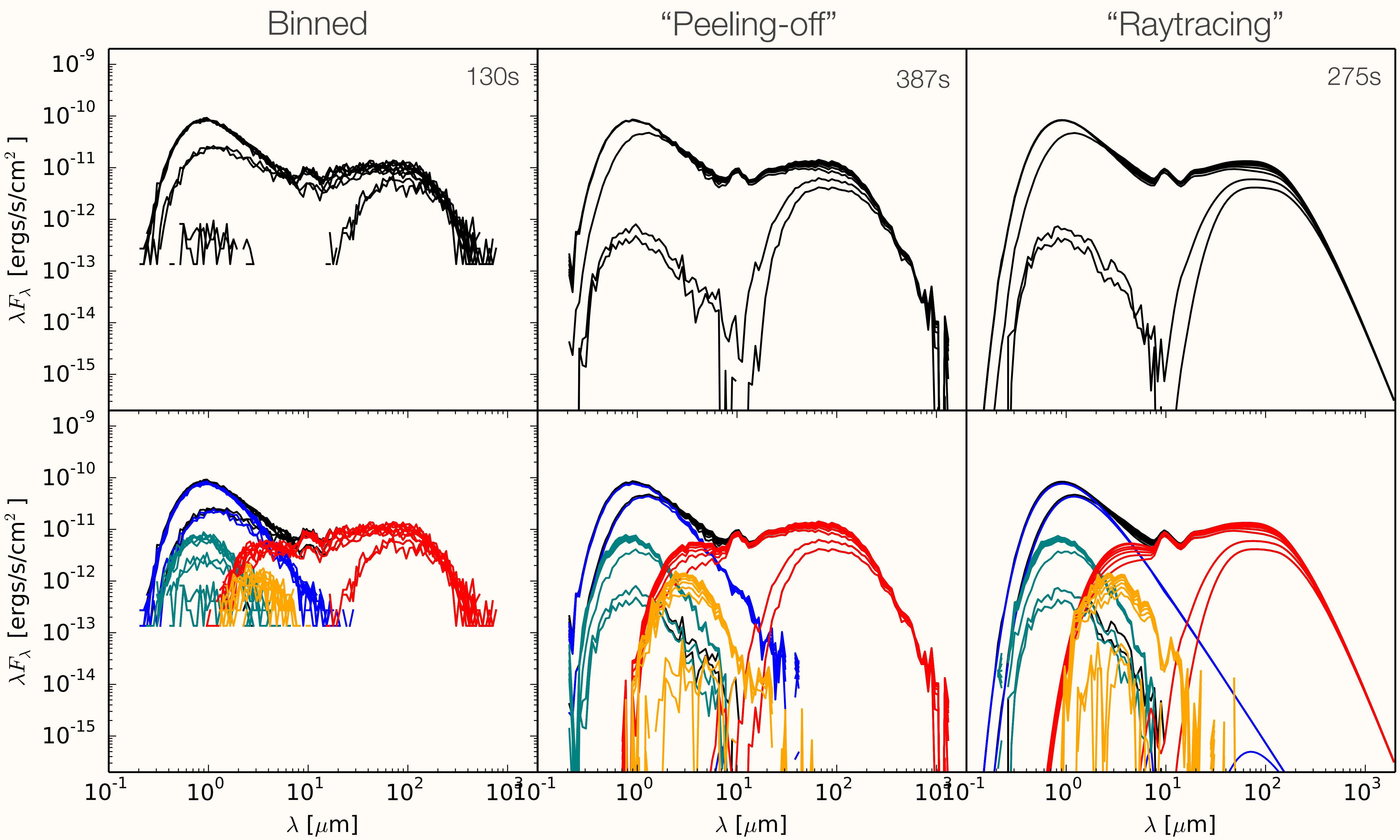
Binned


130s
387s
275s
λF_λ [ergs/s/cm^2]
10^{-9}
10^{-10}
10^{-11}
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
λ [μm]

Binned

“Peeling-off”

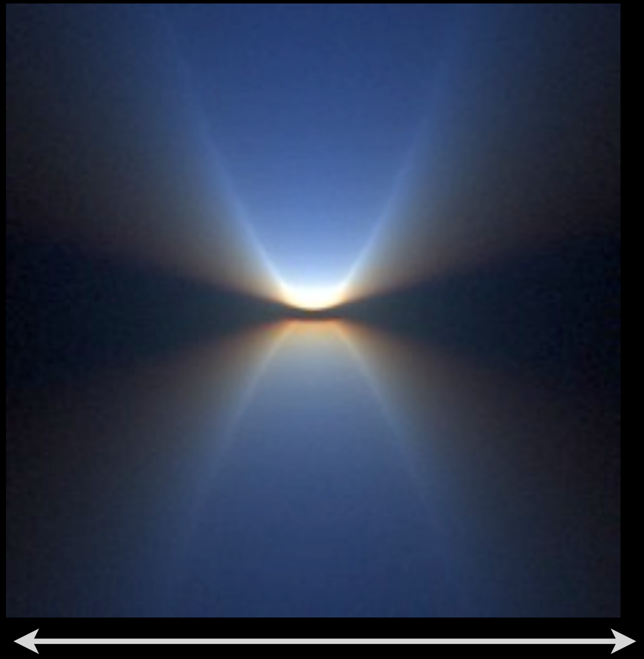



Binned
“Peeling-off”
“Raytracing”
130s
387s
275s
λF_λ [ergs/s/cm^2]
10^{-9}
10^{-10}
10^{-11}
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
λ [μm]

Analytical model of a Young Stellar Object

1 - 3 µm

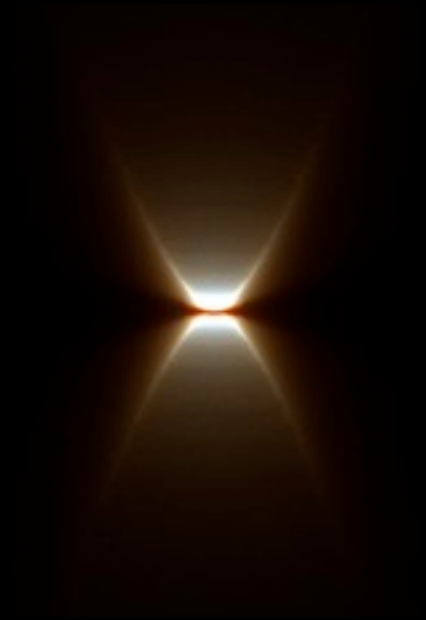
Scattered light + extinction



6×10^{14} m (4000 AU)

20 - 60 µm

Warm dust



400 - 800 µm

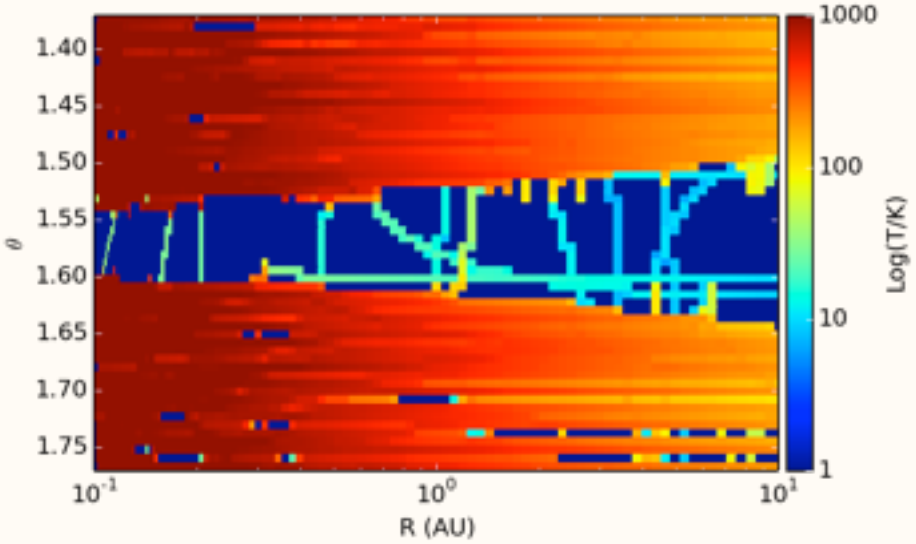
Cool dust

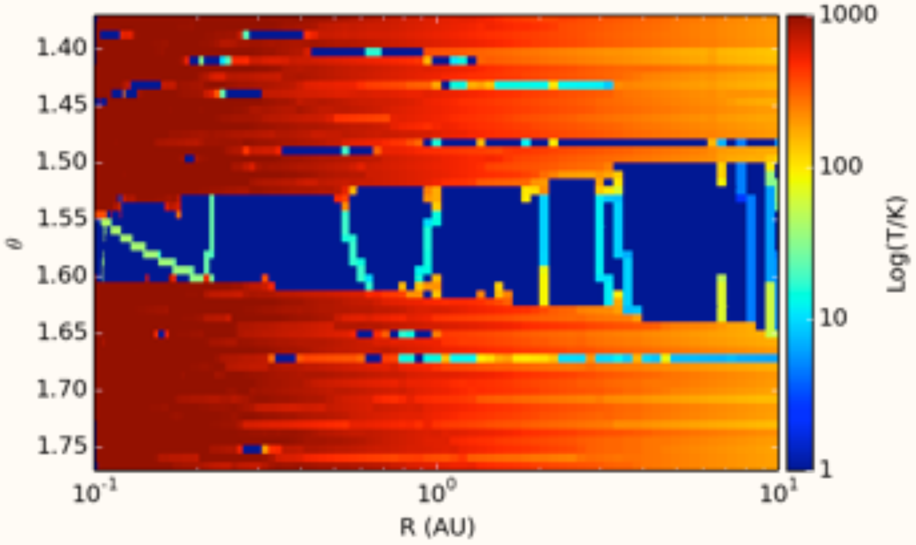
Note on Parallelization

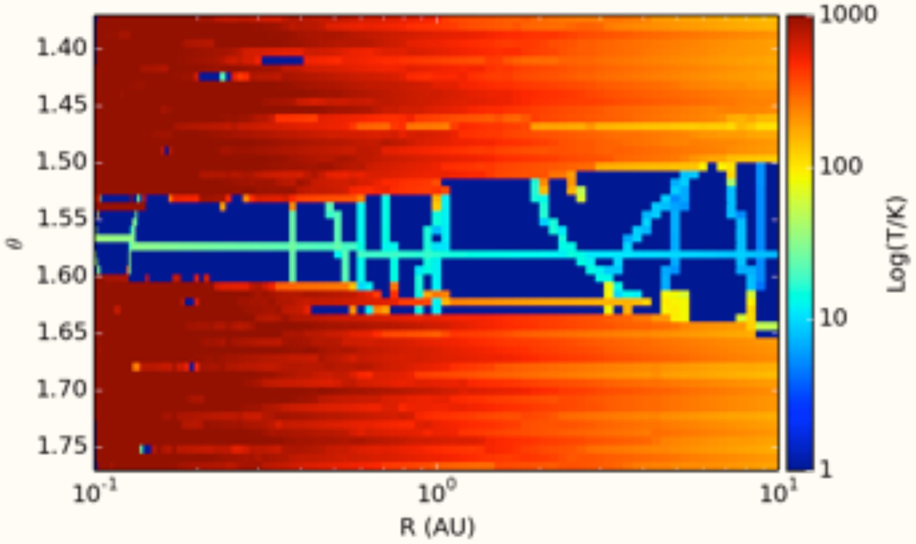
Monte-Carlo radiative transfer is extremely efficient in parallel (“embarrasingly parallel”)

Parallelization is easiest and most efficient with the Lucy (1999, A&A 344, 282) algorithm for temperature calculation, since the path lengths can be computed on separate cores and combined at the end of the iteration.

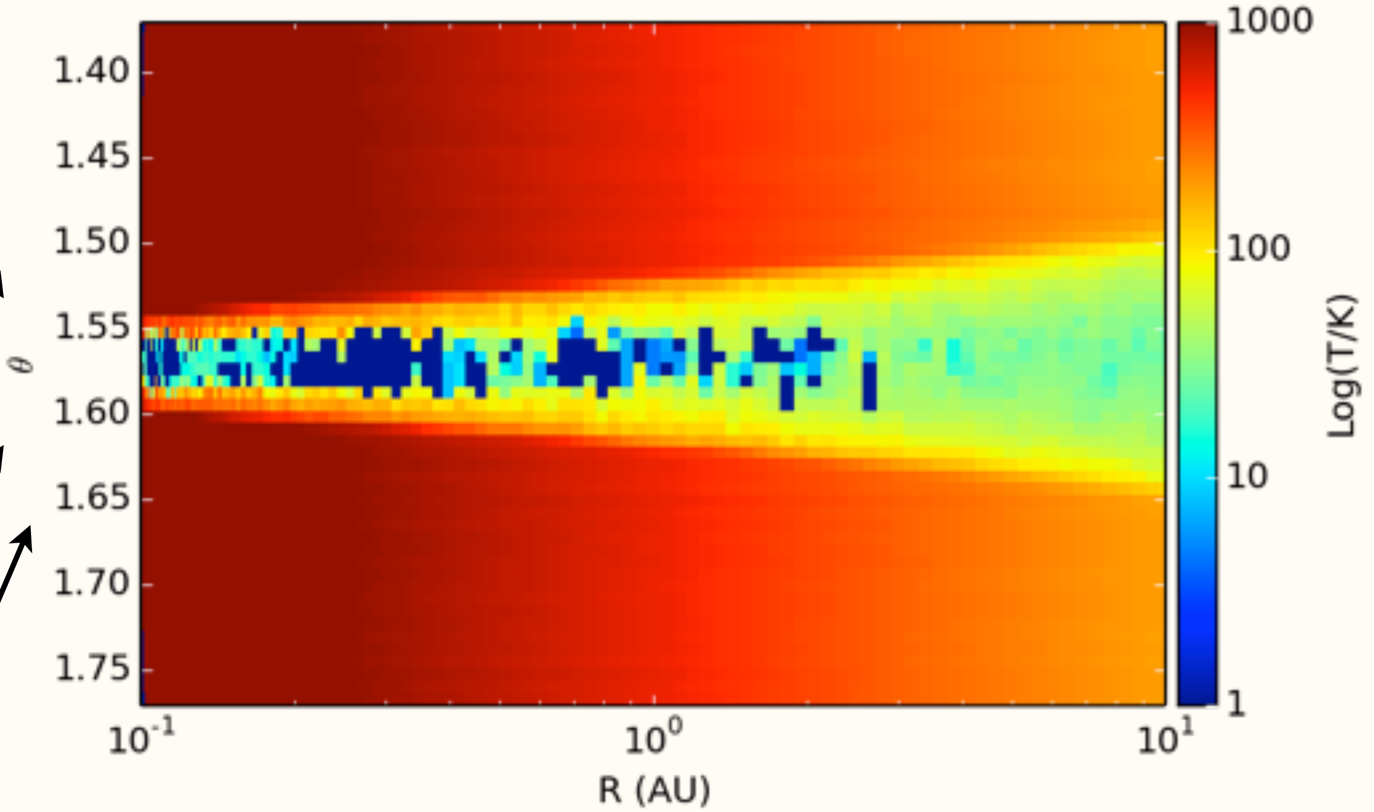
Simply split up photons by processes!







+97 more



and similarly for SEDs and images

Parallelization can provide **significant** speedups!