Jacopo Fritz, IRyA - UNAM

DataCubes in Astrophysics

Lesson 2

Model datacubes from radiative transfer

ACTIVE GALACTIC NUCLEI

An historical overview

ACTIVE GALACTIC NUCLEI

An historical overview

- It all started with the discovery of a new kind of objects with very intense emission lines;

ACTIVE GALACTIC NUCLEI

An historical overview

- It all started with the discovery of a new kind of objects with very intense emission lines;
- identification of these objects as galaxies;

ACTIVE GALACTIC NUCLEI

An historical overview

- It all started with the discovery of a new kind of objects with very intense emission lines;
- identification of these objects as galaxies;
- identification of the center of these objects as the “emitter”;

ACTIVE GALACTIC NUCLEI

An historical overview

- It all started with the discovery of a new kind of objects with very intense emission lines;
- identification of these objects as galaxies;
- identification of the center of these objects as the “emitter”;
- realization that the emitting region must be “small” (images of the center of these galaxies were point-like/unresolved);

ACTIVE GALACTIC NUCLEI

An historical overview

- It all started with the discovery of a new kind of objects with very intense emission lines;
- identification of these objects as galaxies;
- identification of the center of these objects as the “emitter”;
- realization that the emitting region must be “small” (images of the center of these galaxies were point-like/unresolved);
- mass-radius constraints led to the hypothesis of very high energy densities!

ACTIVE GALACTIC NUCLEI

The breakthrough

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;
- other where unknown, unidentified sources;

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;
- other where unknown, unidentified sources;
- using a lunar eclipse, in 1963 M. Schmidt pinpointed the position of a powerful radio source;

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;
- other where unknown, unidentified sources;
- using a lunar eclipse, in 1963 M. Schmidt pinpointed the position of a powerful radio source;
- it turned out to have broad intense emission lines;

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;
- other where unknown, unidentified sources;
- using a lunar eclipse, in 1963 M. Schmidt pinpointed the position of a powerful radio source;
- it turned out to have broad intense emission lines;
- the estimated redshift was z=0.158;

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;
- other where unknown, unidentified sources;
- using a lunar eclipse, in 1963 M. Schmidt pinpointed the position of a powerful radio source;
- it turned out to have broad intense emission lines;
- the estimated redshift was z=0.158;
- the derived luminosity was ~100 higher than a “normal” galaxy.

ACTIVE GALACTIC NUCLEI

The breakthrough

- The 3C and 3CR catalogues included powerful radio-sources (survey limit of 9 Jy);
- many were nearby radio galaxies;
- other where unknown, unidentified sources;
- using a lunar eclipse, in 1963 M. Schmidt pinpointed the position of a powerful radio source;
- it turned out to have broad intense emission lines;
- the estimated redshift was z=0.158;
- the derived luminosity was ~100 higher than a “normal” galaxy.

ACTIVE GALACTIC NUCLEI

Radio Galaxies

Fanaroff Riley classes

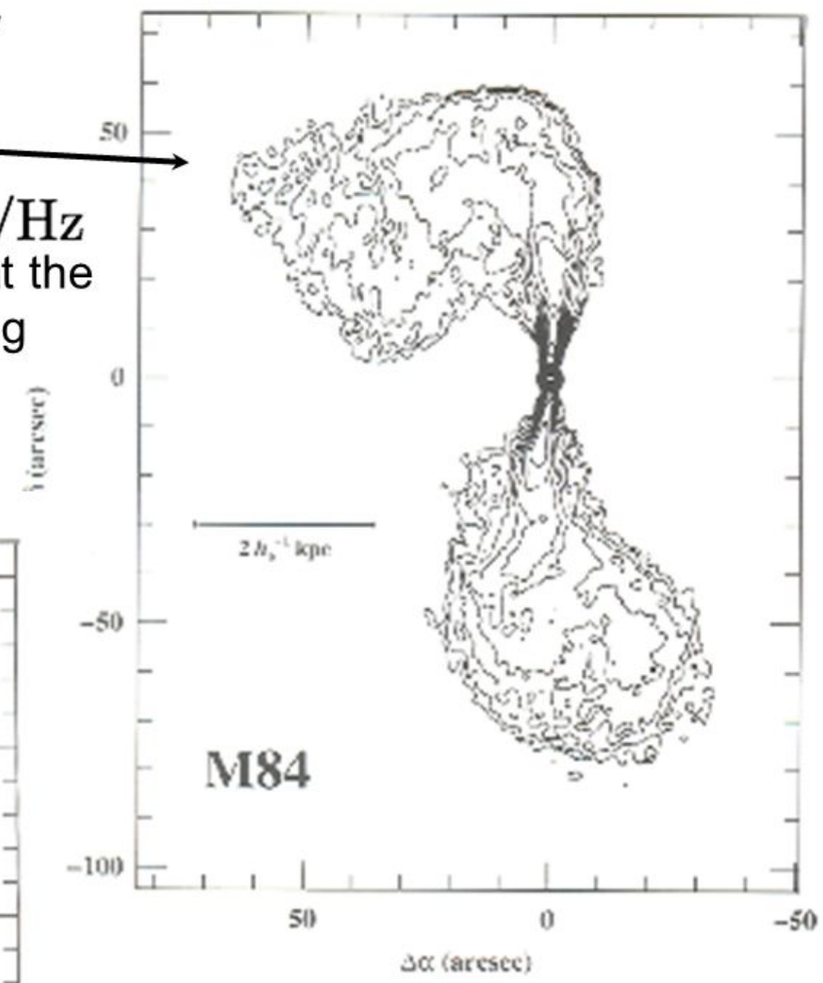
Fanaroff Riley 1974

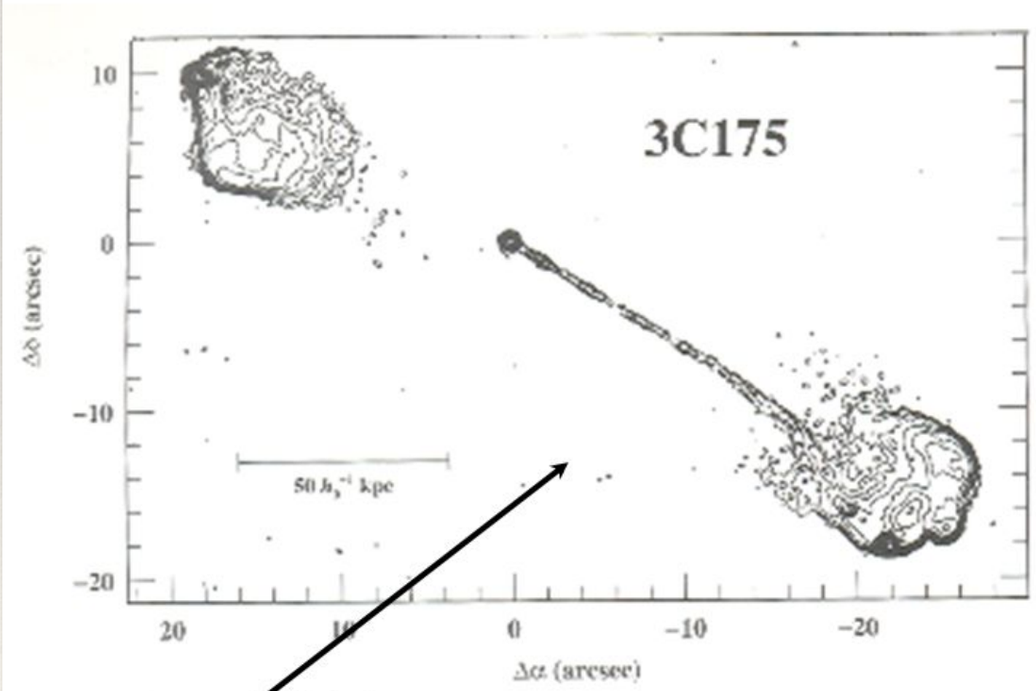
FR I

low power

$L_\nu(1.4\mathrm{GHz}) \lesssim 10^{32}$ erg/s/Hz

extended component brighter at the center, and gradually decreasing towards borders





FR II

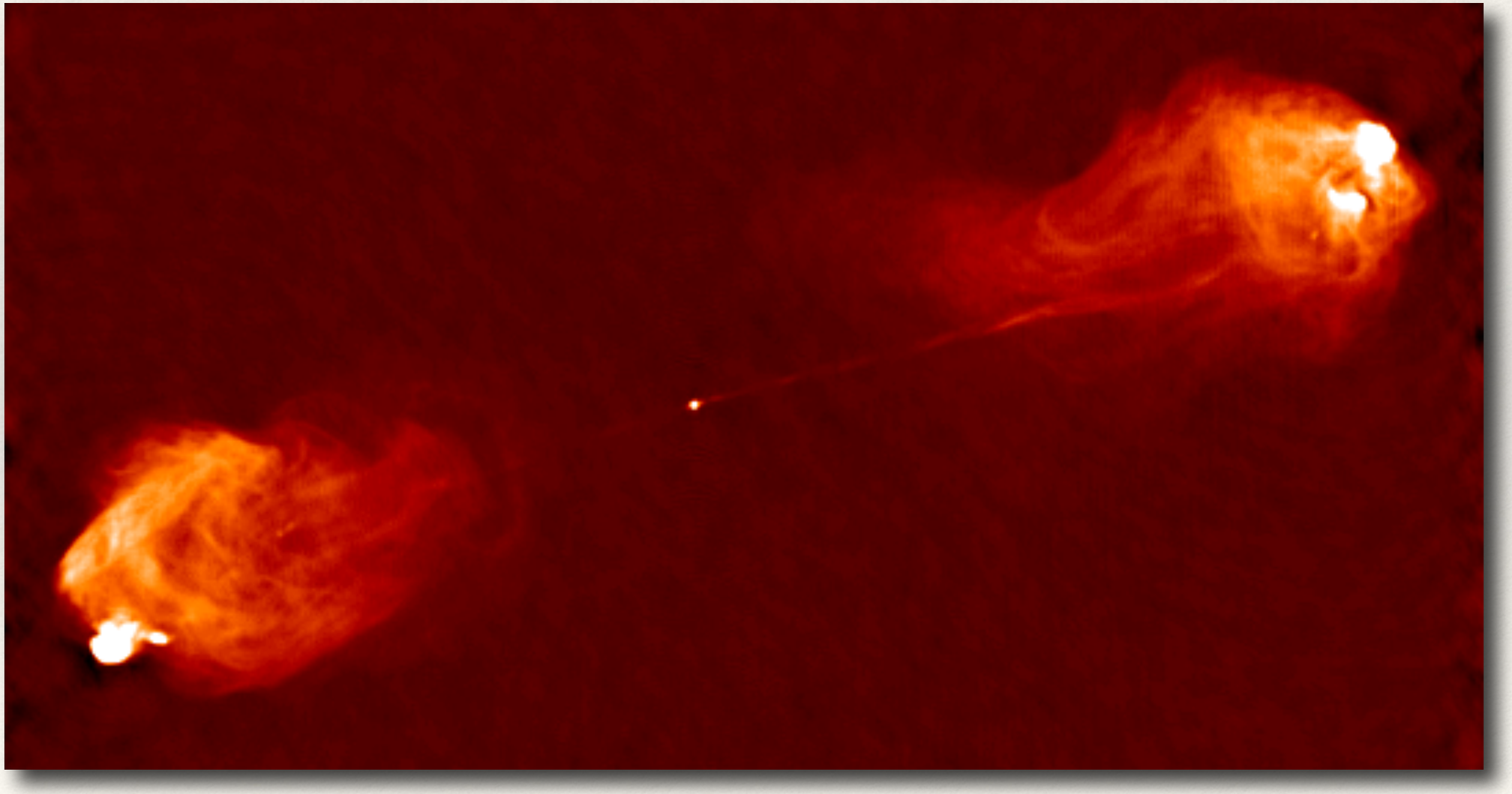
high power

extended component brighter at border

quasars are FR II sources

CYGNUS -A

@ 4.8 GHz



ACTIVE GALACTIC NUCLEI

We find different kind of objects under the AGN classification:

- Seyfert Galaxies;
- Radio Galaxies;
- QSO: Quasi-Stellar Objects;
- Quasar;
- OVV: Optically Violently Variables;
- BL Lac.

ACTIVE GALACTIC NUCLEI

We find different kind of objects under the AGN classification:

- Seyfert Galaxies;
- Radio Galaxies;
- QSO: Quasi-Stellar Objects;
- Quasar;
- OVV: Optically Violently Variables;
- BL Lac.

- Blazars

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a “normal” galaxy);

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a “normal” galaxy);
- this production happens in a volume less than 1 pc^3;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a “normal” galaxy);
- this production happens in a volume less than 1 pc^3;
- the energy emission covers a wide range of frequencies;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a “normal” galaxy);
- this production happens in a volume less than 1 pc^3;
- the energy emission covers a wide range of frequencies;
- mechanical energy is released as well;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a "normal" galaxy);
- this production happens in a volume less than 1 pc^3;
- the energy emission covers a wide range of frequencies;
- mechanical energy is released as well;
- extremely intense emission lines, in some cases very broad;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a "normal" galaxy);
- this production happens in a volume less than 1 pc^3;
- the energy emission covers a wide range of frequencies;
- mechanical energy is released as well;
- extremely intense emission lines, in some cases very broad;
- great variety in the radio emission;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- Very high luminosity (up to 10^4 times than a "normal" galaxy);
- this production happens in a volume less than 1 pc^3;
- the energy emission covers a wide range of frequencies;
- mechanical energy is released as well;
- extremely intense emission lines, in some cases very broad;
- great variety in the radio emission;
- can be very strongly variable.

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- The range of observed physical phenomena is quite broad;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- The range of observed physical phenomena is quite broad;
- not all of the AGN display ALL of these characteristics;

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- The range of observed physical phenomena is quite broad;
- not all of the AGN display ALL of these characteristics;
- this indicates a high degree of complexity in the mechanism(s) of energy production.

ACTIVE GALACTIC NUCLEI

OBSERVATIONAL FACTS:

- The range of observed physical phenomena is quite broad;
- not all of the AGN display ALL of these characteristics;
- this indicates a high degree of complexity in the mechanism(s) of energy production.

Classification might be extremely difficult!

ACTIVE GALACTIC NUCLEI

They all have in common:

- Strong non-thermal emission;
- coming from the galactic nucleus;
- produced in a very small scale region;
- highly ionising;
- often variable in short time-scales.

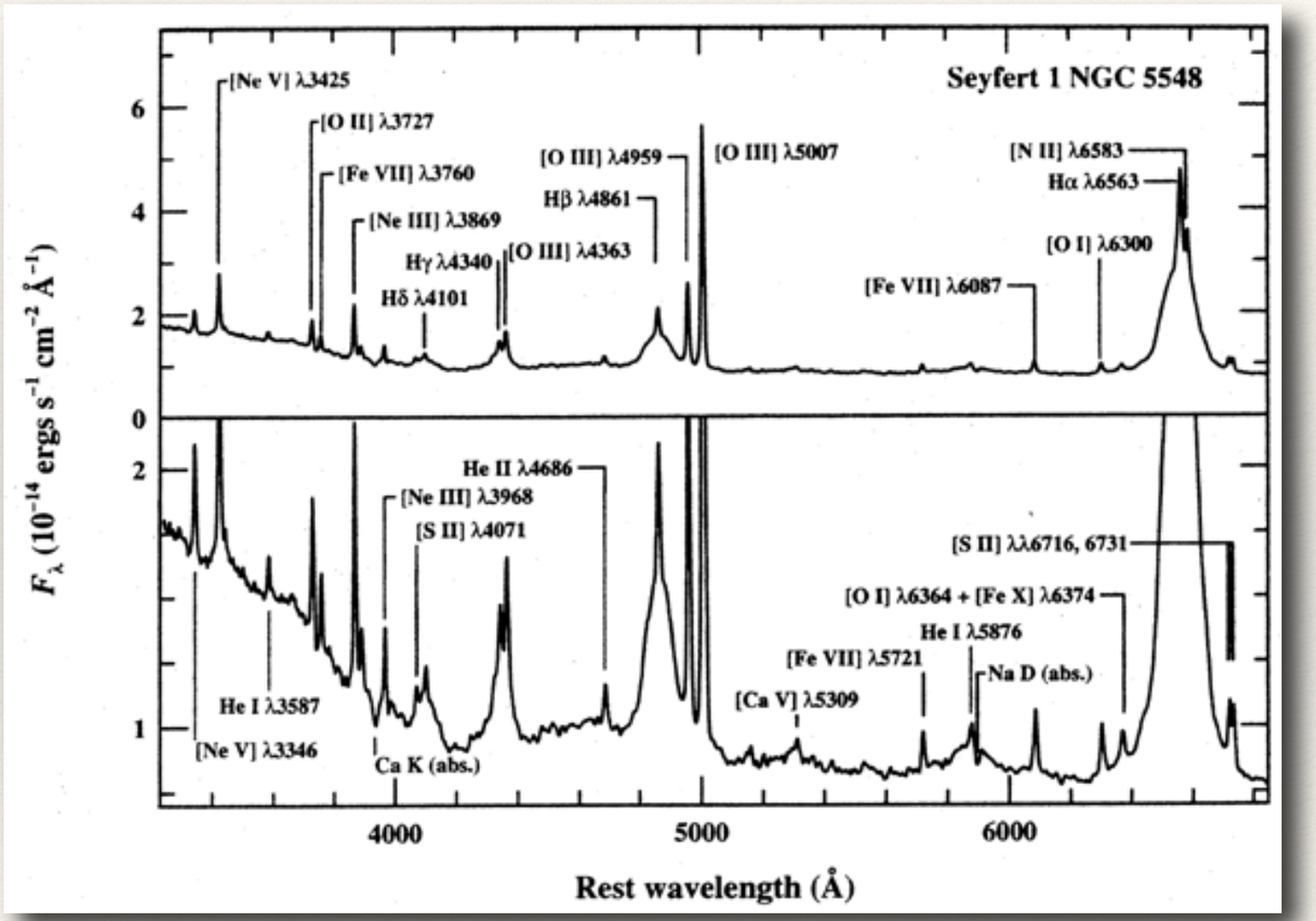
AGN: Observed features

- Bolometric luminosity between 10^{42} and 10^{48} erg s^{-1};
- this is from 0.01 to 10^4 more than a **typical** galaxy;
- typical luminosity variation are of about 10% within few years (and the period is wavelength-dependent);
- relatively "flat" (in υF_υ units) SED, ad odds with "normal" (star —powered) galaxies).

Seyfert Galaxies

- Low-luminosity AGNs;
- first to be studied in depth;
- first found in very nearby galaxies;
- identified in spiral galaxies with bright cores;
- characterized by very intense lines from highly ionized gas.

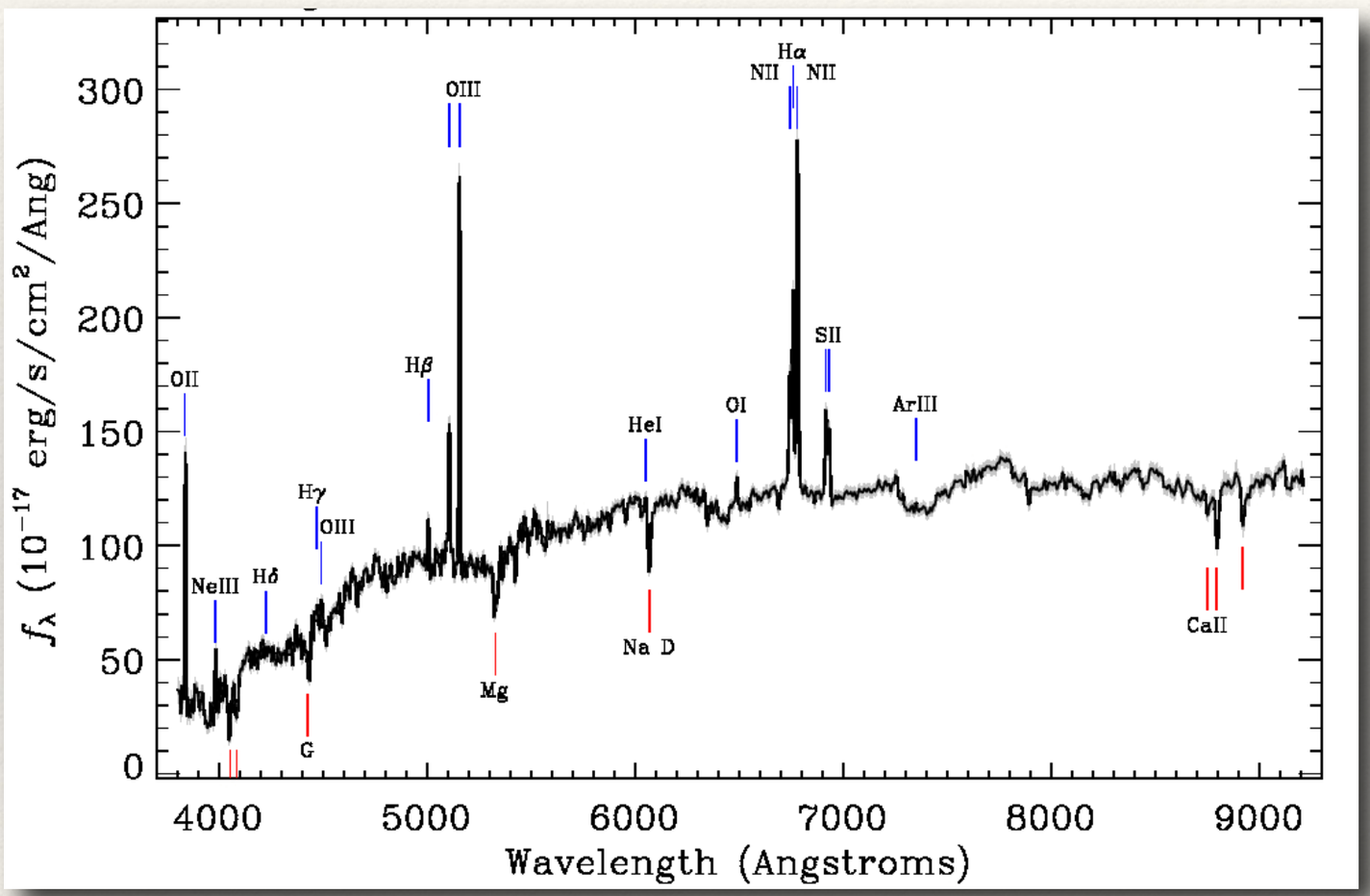
Seyfert Type 1



- broad emission lines;
- narrow emission lines as well;
- lines from highly ionized elements;
- extremely blue continuum;

They are the low-luminosity counterparts of the QSOs

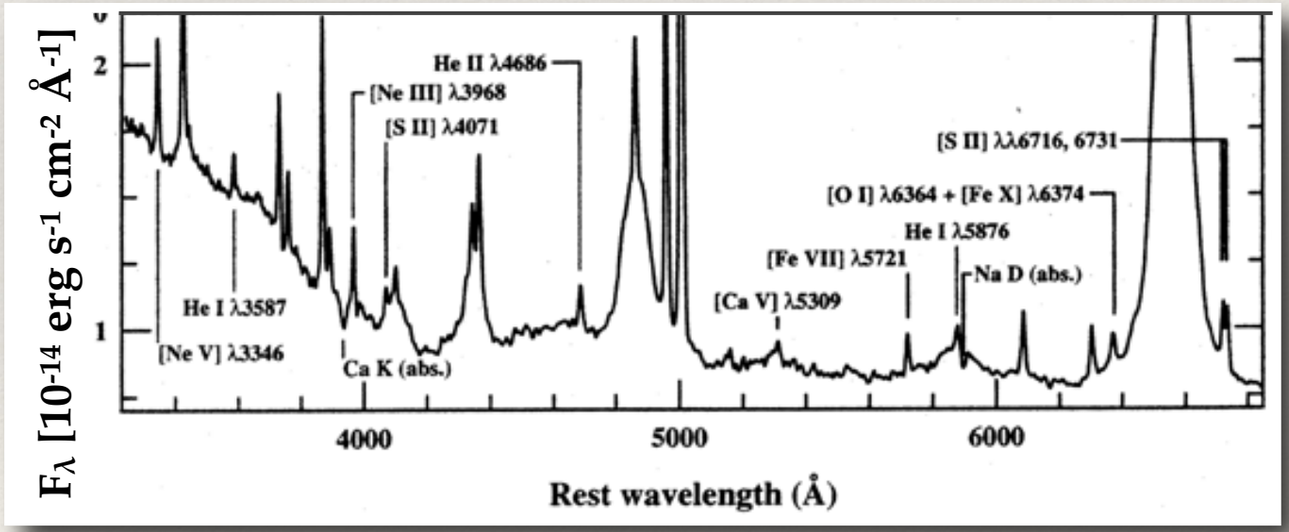
Seyfert Type 2

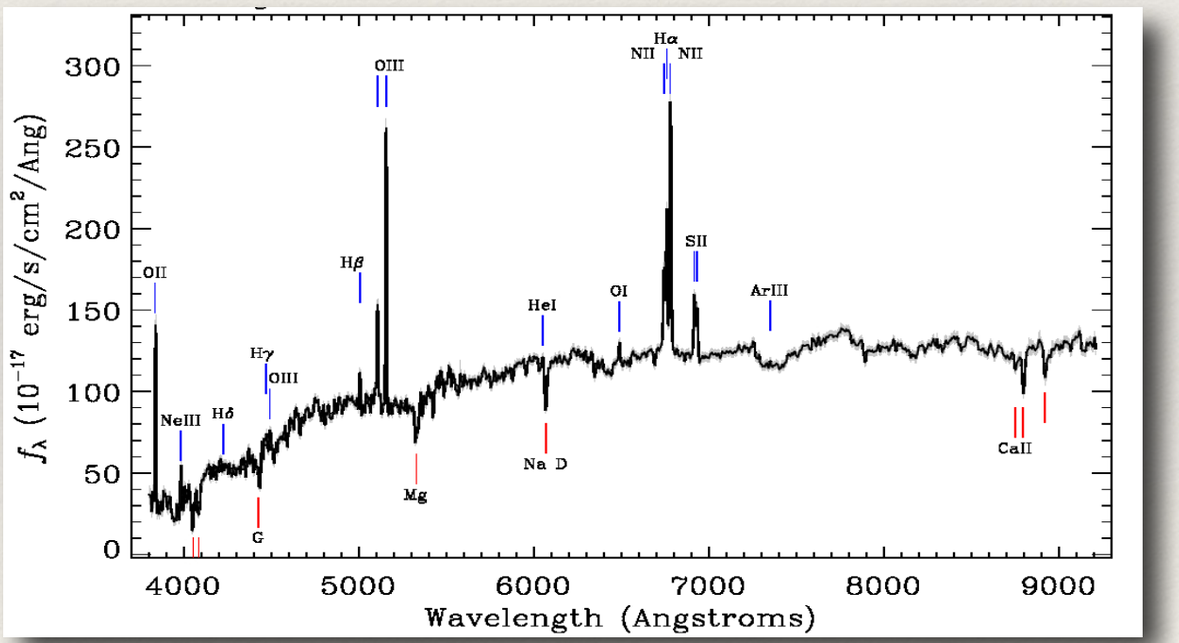


- no broad lines;
- lines from highly ionized elements;
- fairly "normal" continuum (i.e. stellar dominated);

It took a while for type-2 QSO to be discovered

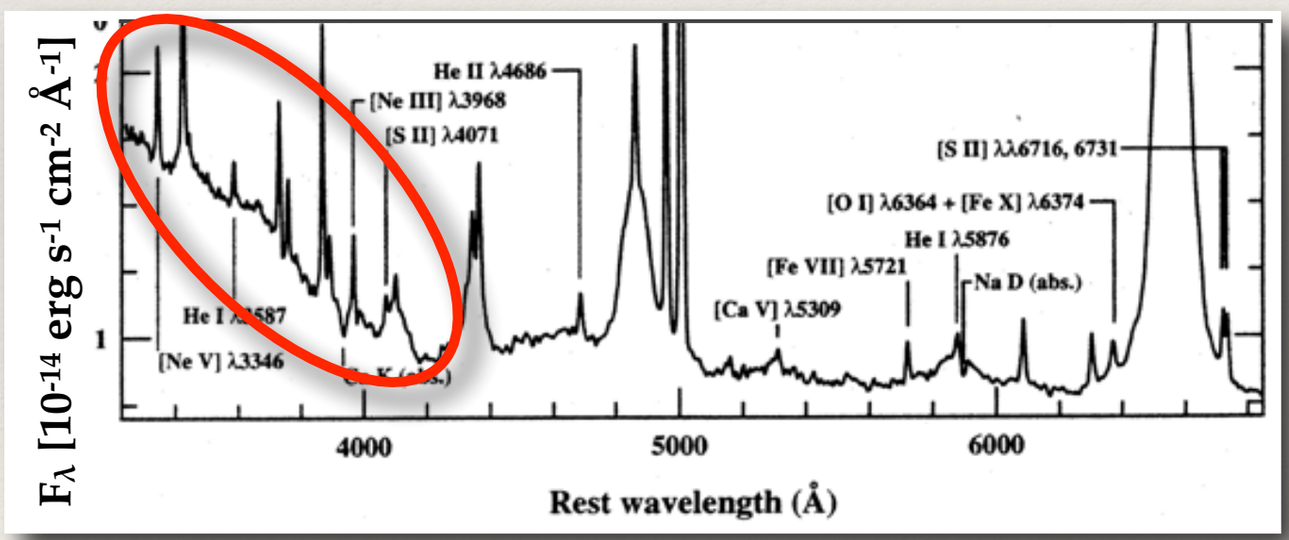
Type 1 vs Type 2





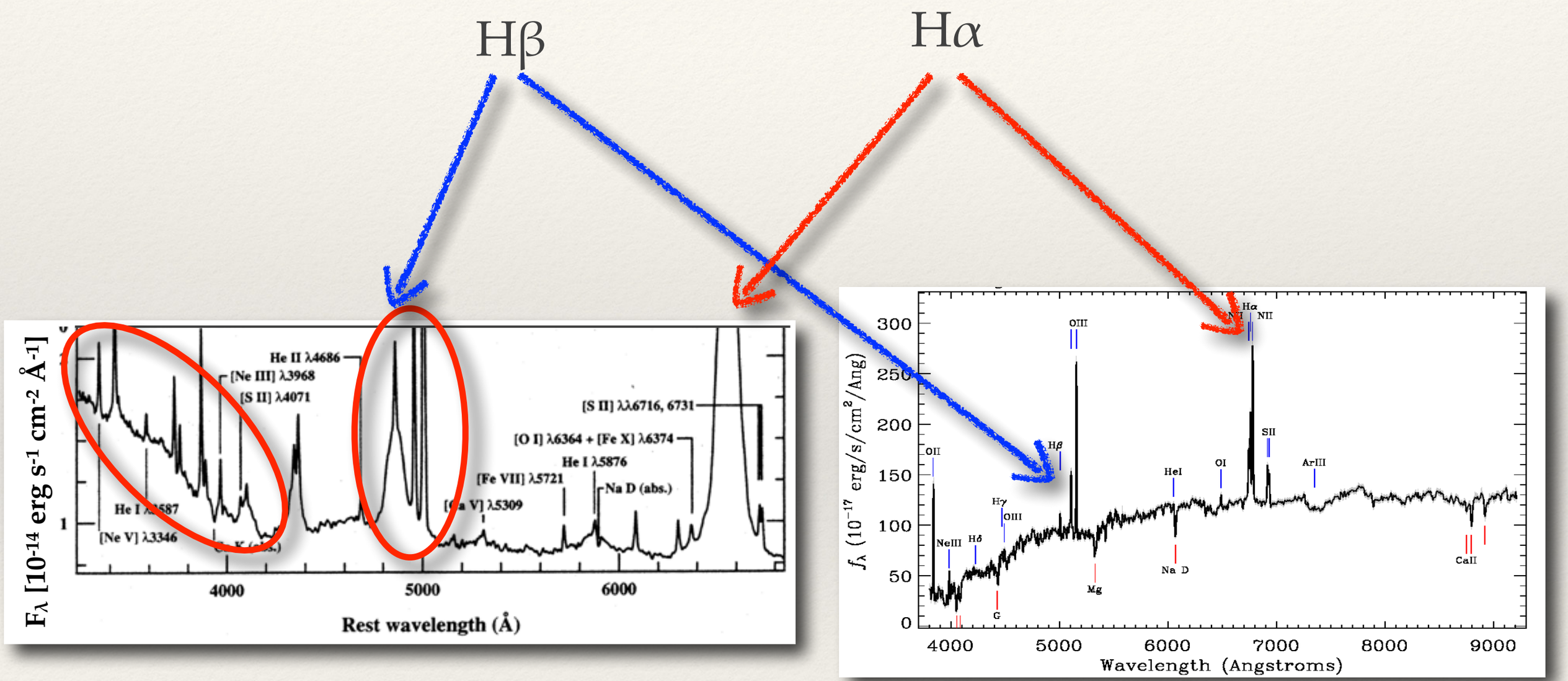
Plus: [OI], [OII], [OIII], [NII], [SII], [NeIV], …

Type 1 vs Type 2



Plus: [OI], [OII], [OIII], [NII], [SII], [NeIV], …

Type 1 vs Type 2



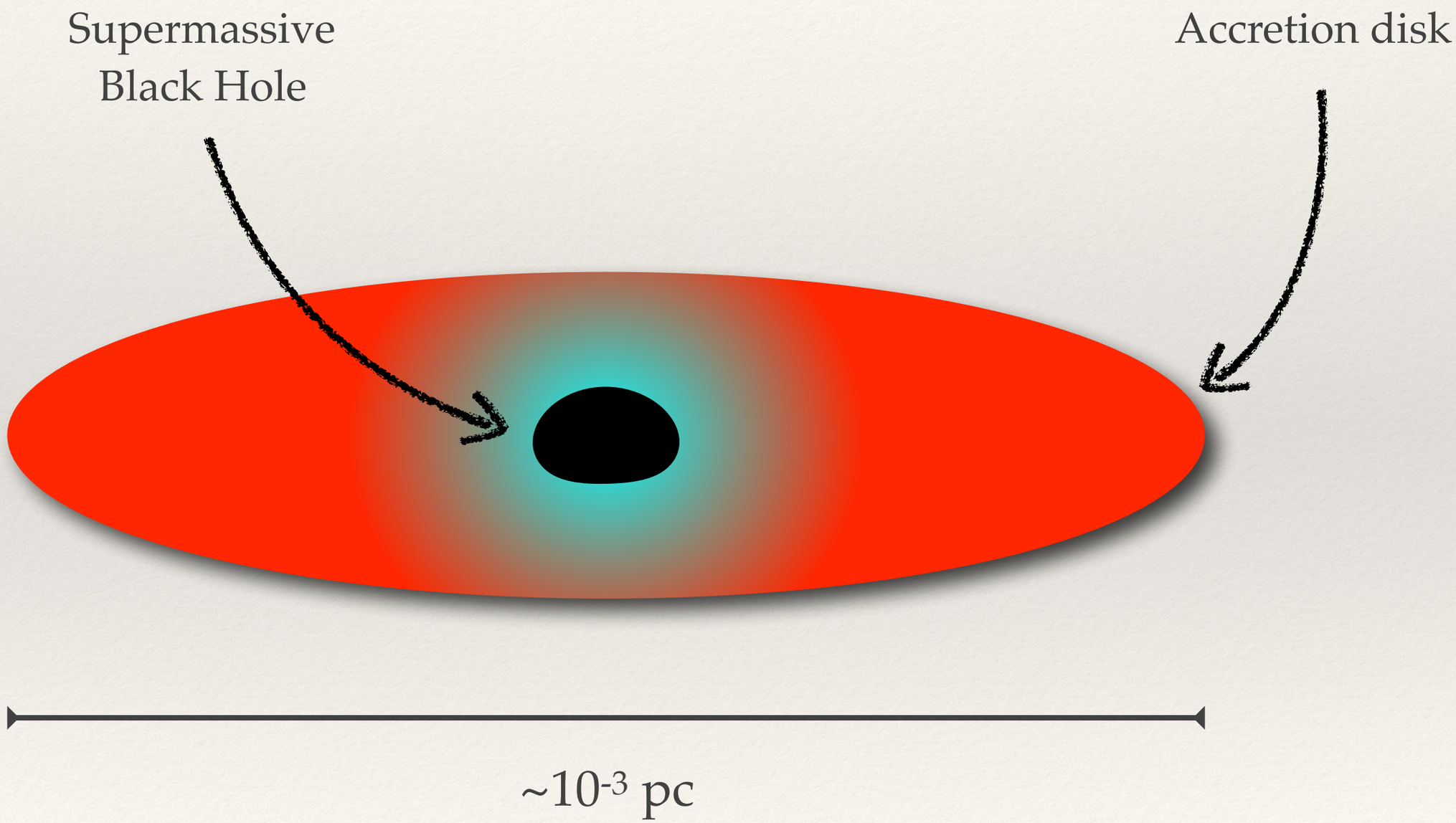
Plus: [OI], [OII], [OIII], [NII], [SII], [NeIV], …

The AGN "Unified Model"

- Both similarities and differences were spotted in this class of objects;
- focussing on the characteristics of optical spectra, and on the MIR emission, a unification scheme was proposed:

These objects represent the same kind of physical phenomena, with optically thick medium obscuring the powerful central source: <u>different class are simply explained as different viewing angles</u>.

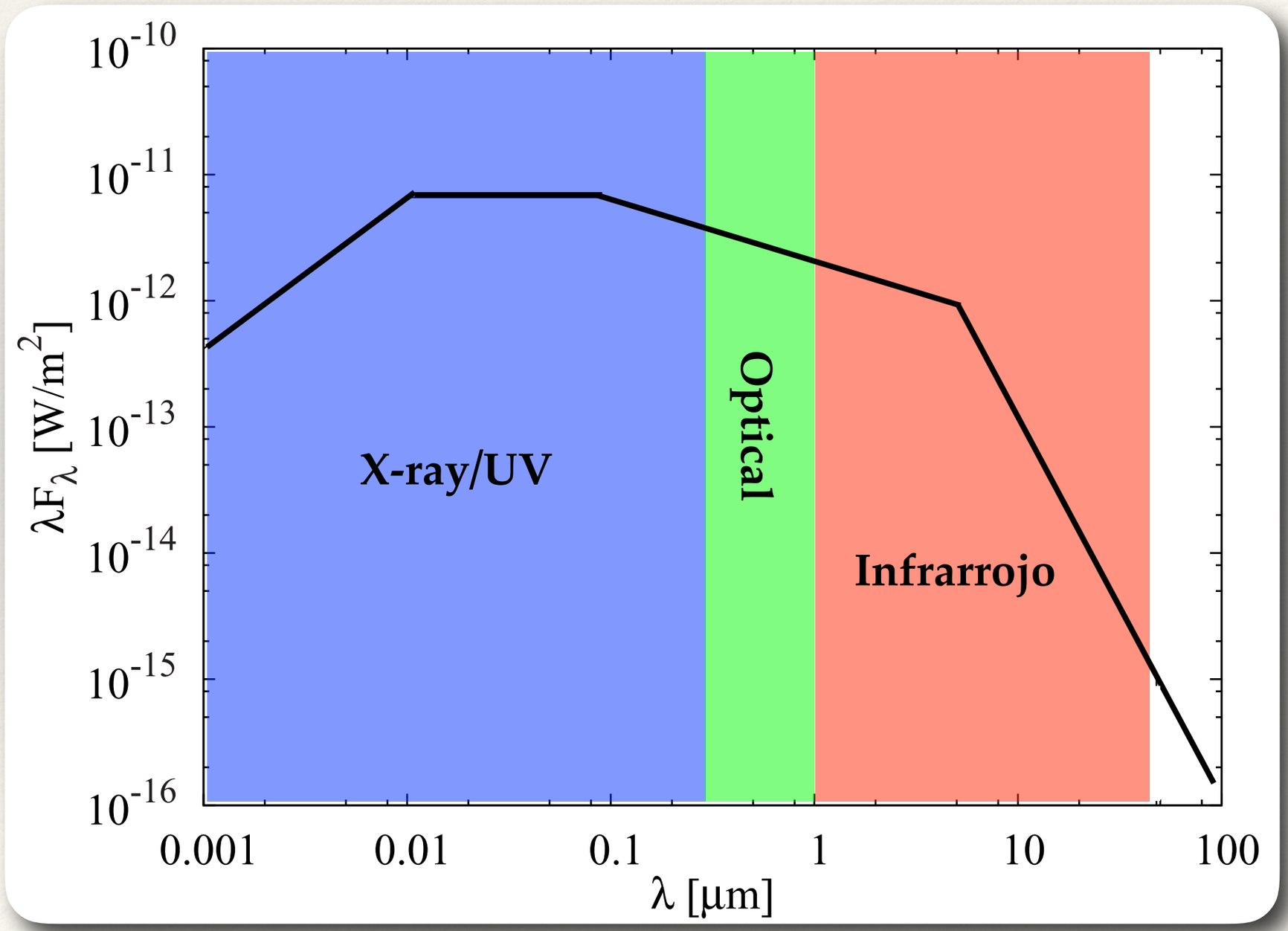
The AGN "Unified Model"



The AGN “Unified Model”

The AGN "Unified Model"

The accretion disk SED

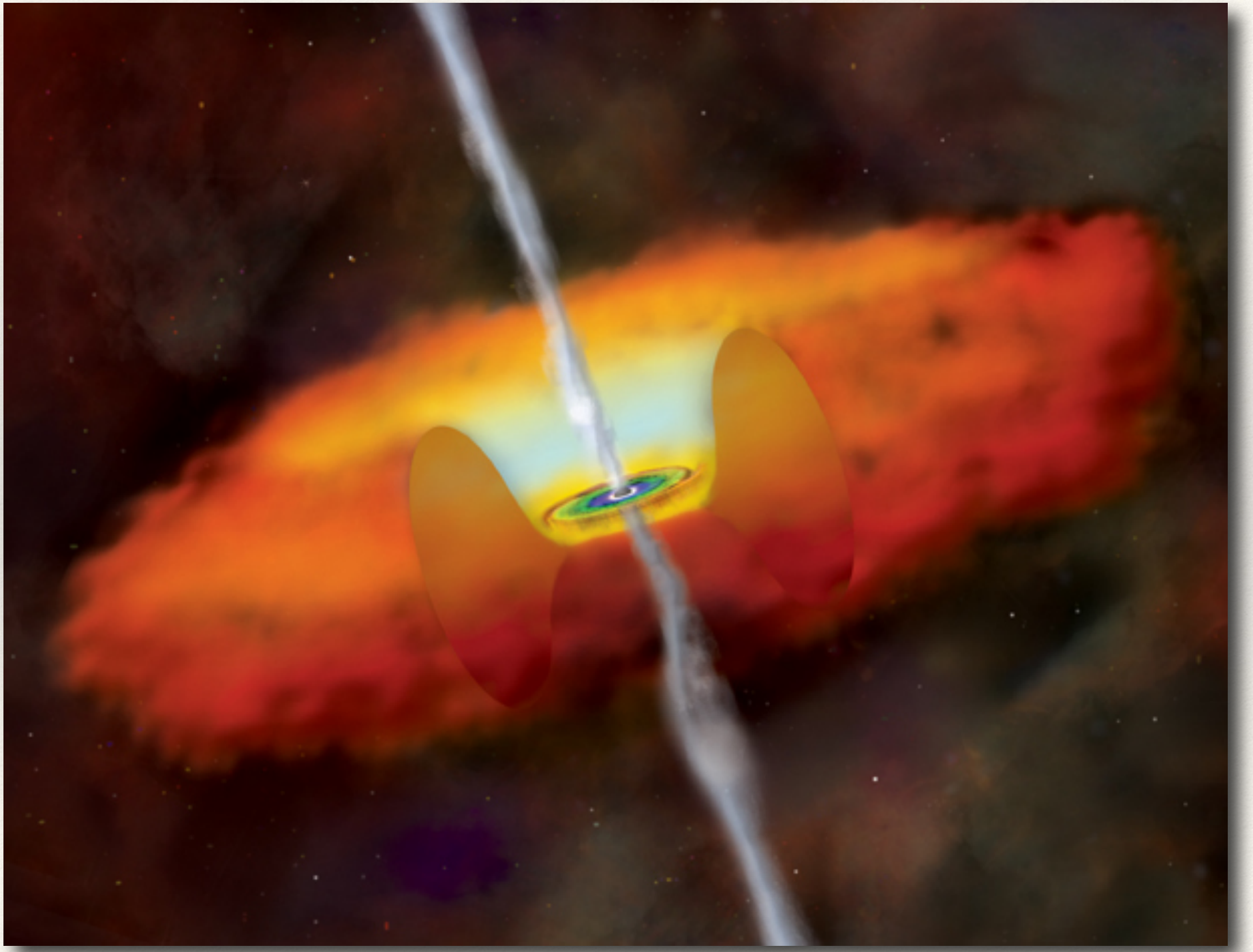


Stalevski et al. 2012

The Dusty Torus

- A further rising of the luminosity is observed towards the NIR/MIR;
- this is believed to be thermal (i.e. black body-like) emission from very hot dust;
- this emission comes from the more external regions of the accretion disk ($d \propto L^{1/2}$);
- the characteristics of this region are significantly different from those of the disk.

The Dusty Torus

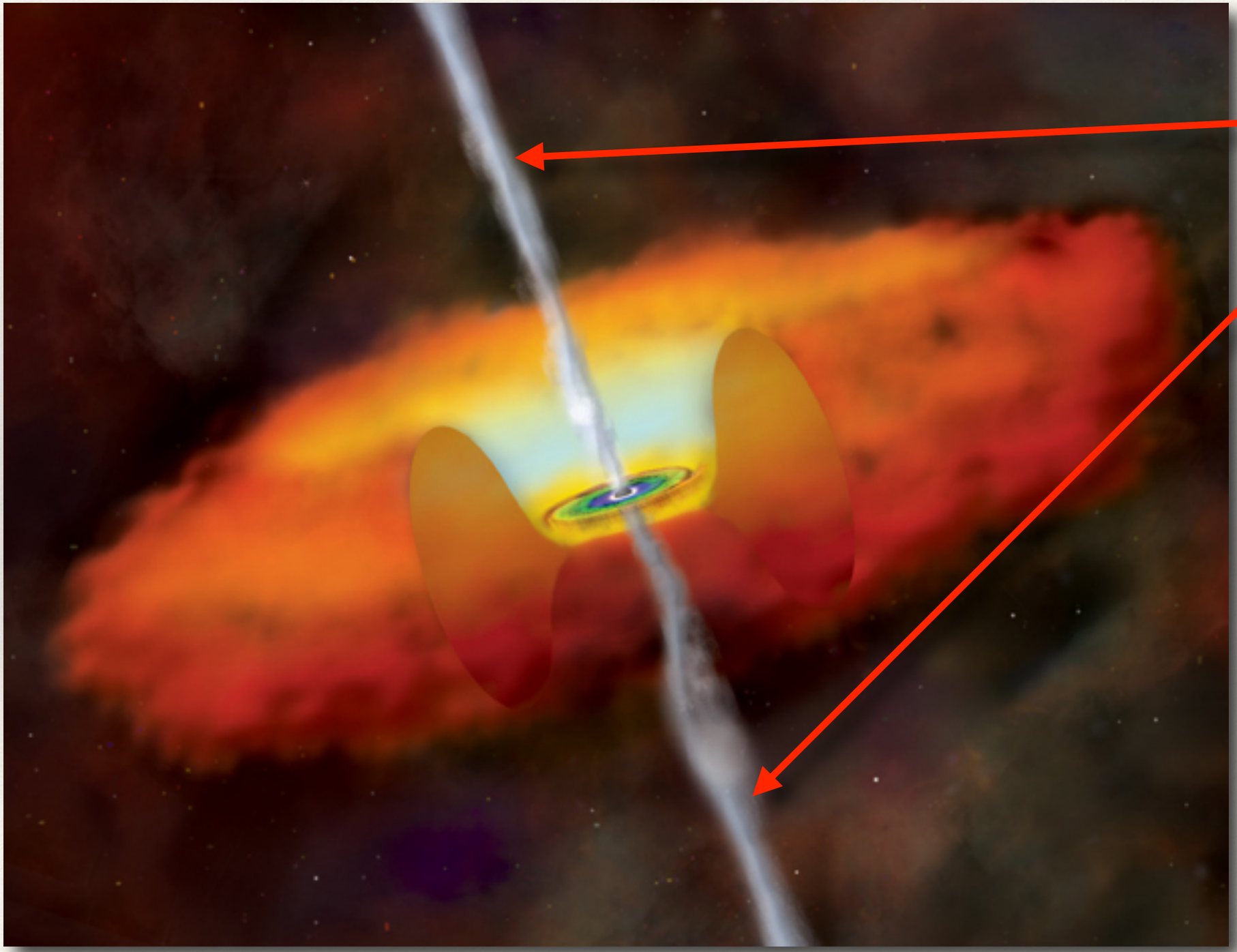


Jets

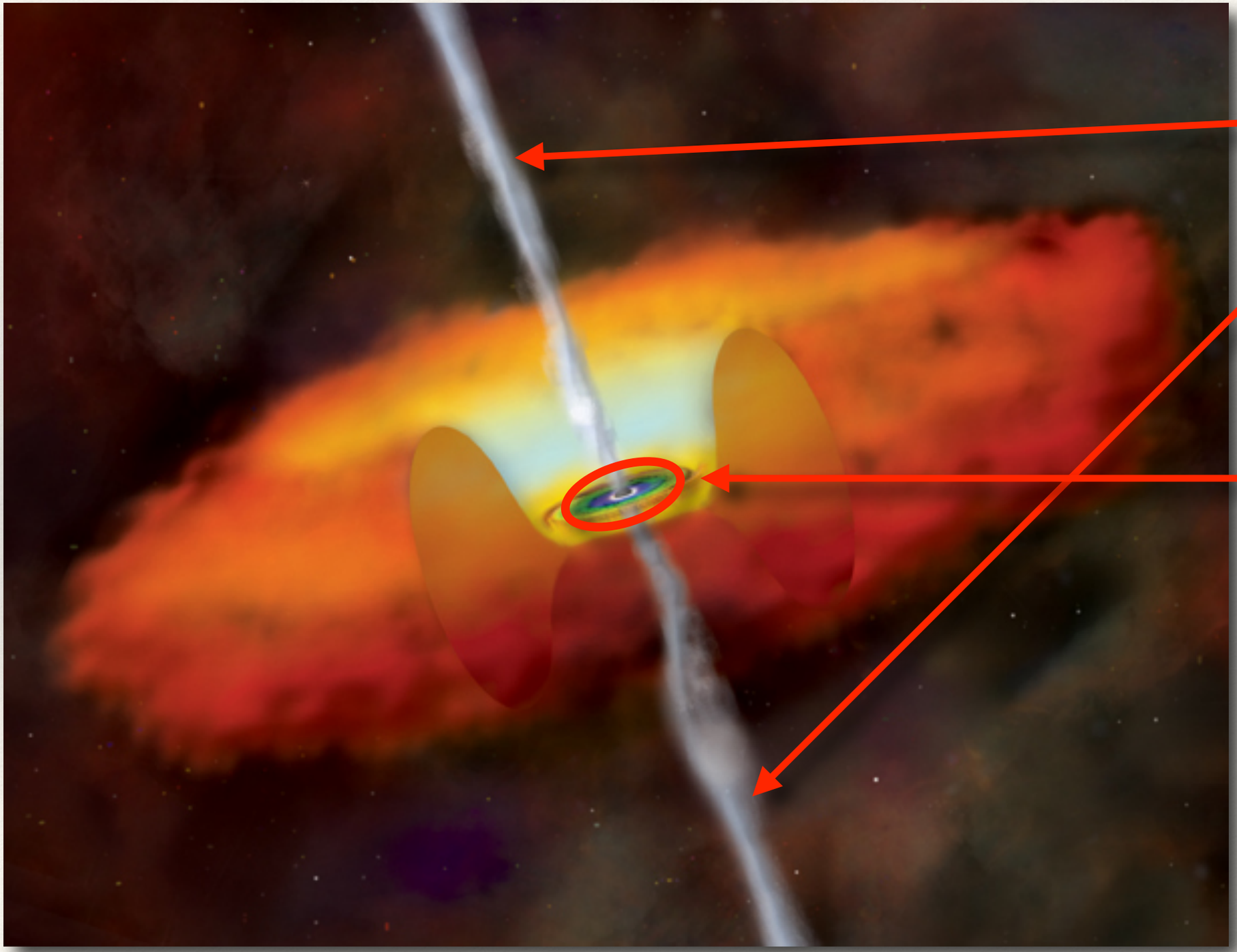
Accretion Disk

Dusty torus

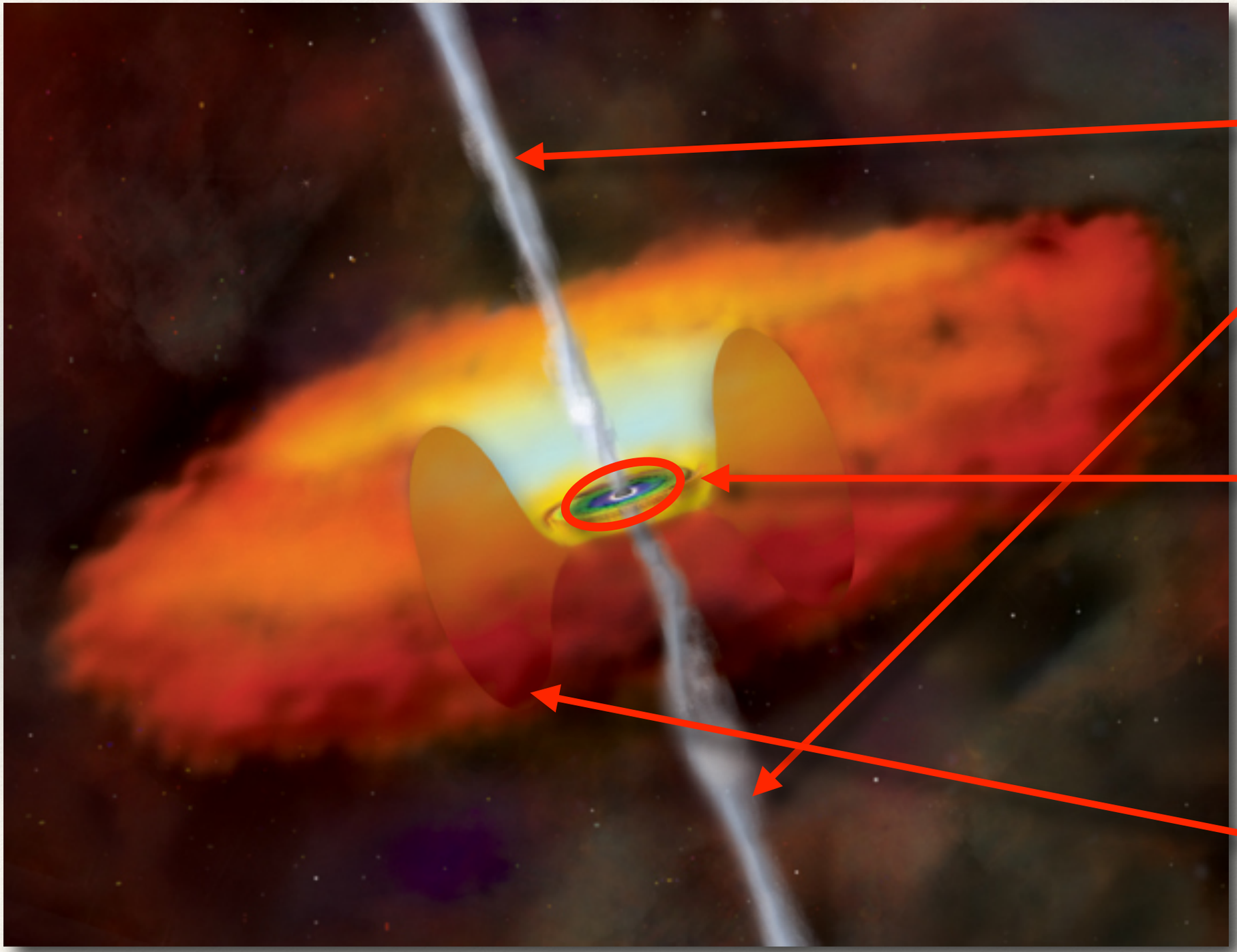
The Dusty Torus



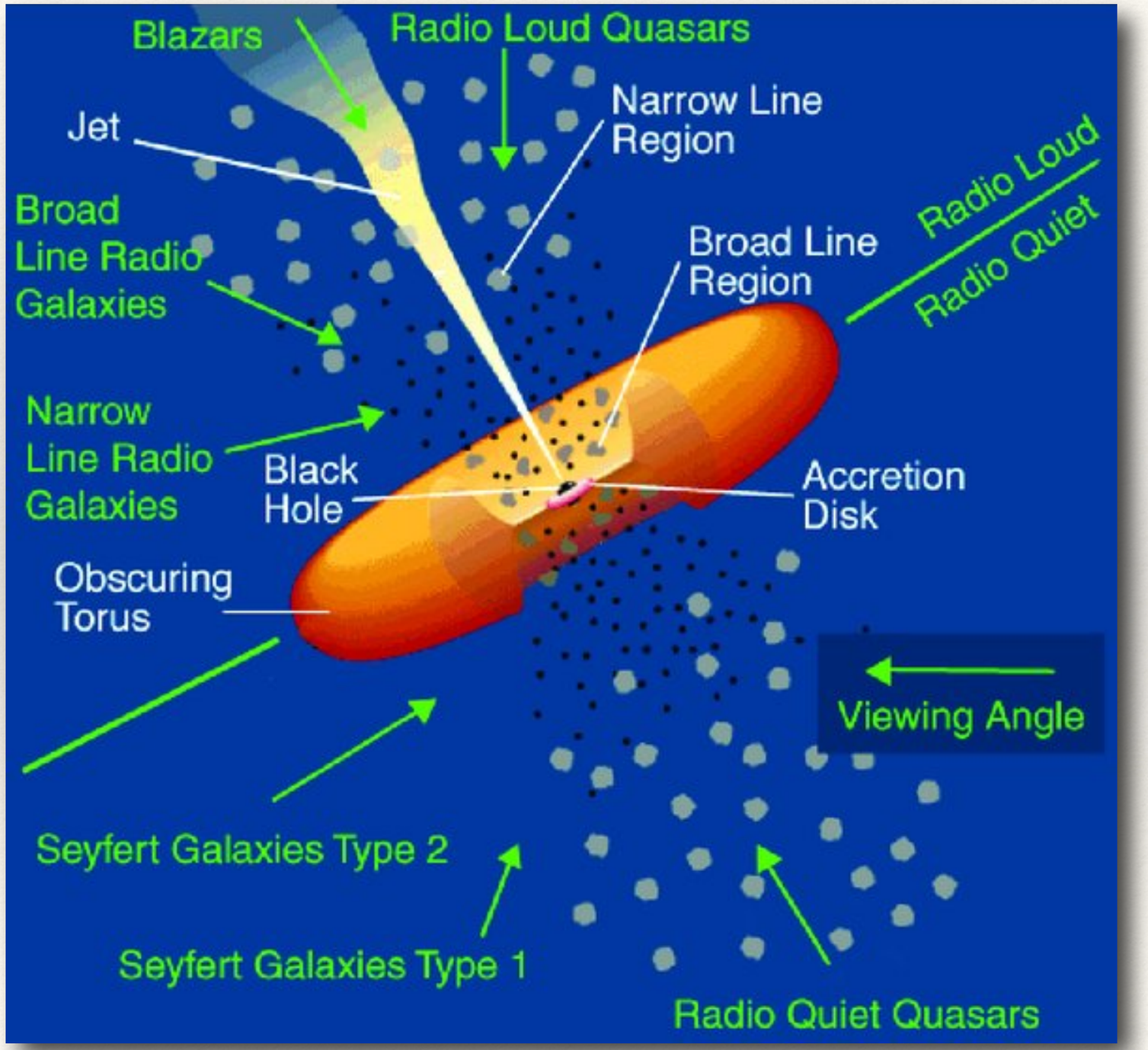
The Dusty Torus



The Dusty Torus



The Unified Model



The Dusty Torus: consequences

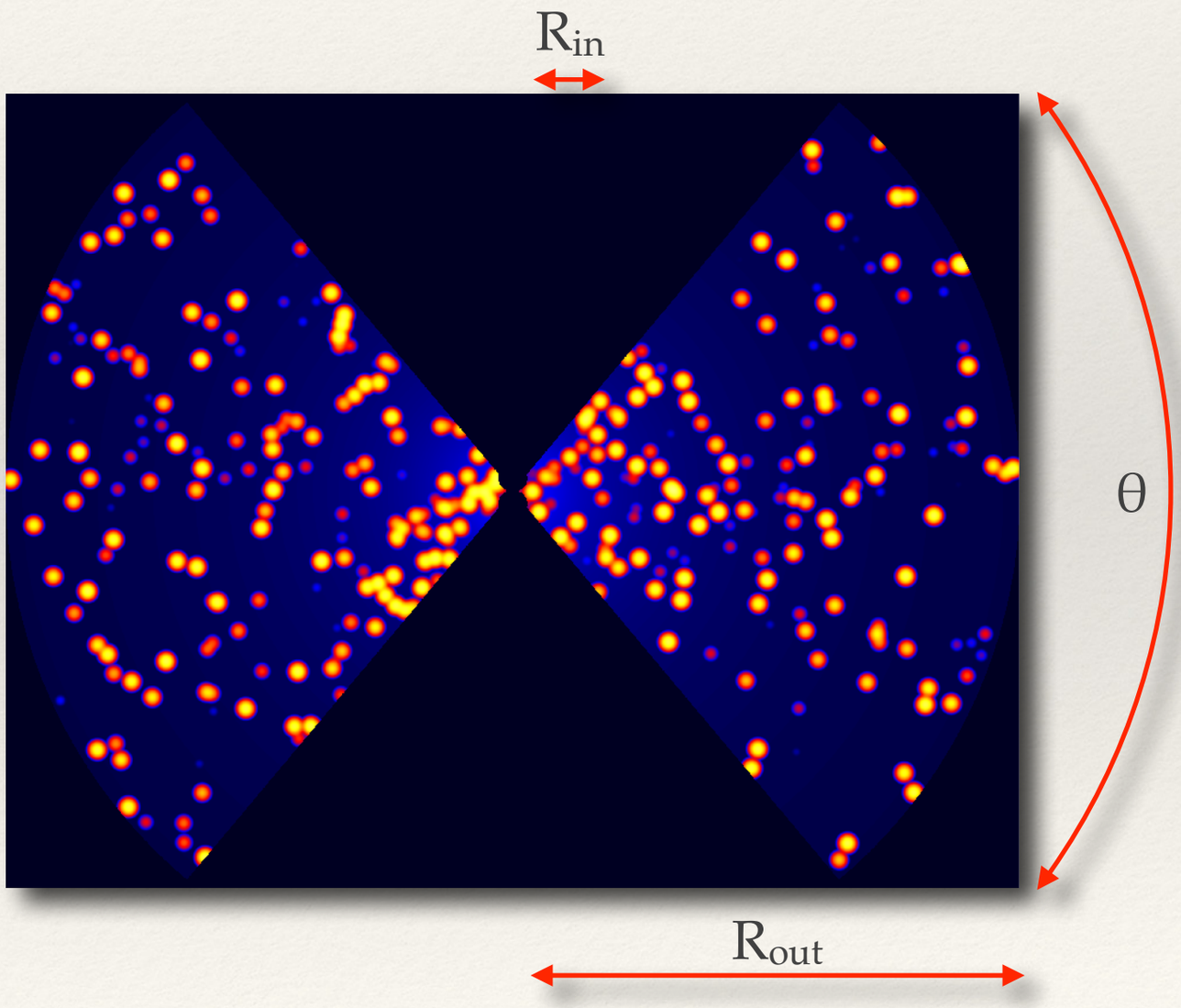
- Explains the NIR-MIR emission;
- naturally arises from a “wind scenario”;
- explains differences between the type-1 and type 2;
- explains polarimetry observations;
- probably depends on the disk luminosity (disappearing at $L<10^{42}$ erg/s and receding torus evidences).

The Dusty Torus

Characteristics

Geometrically defined by:

- inner radius;
- outer radius (questionable);
- torus opening angle;
- filling factor.



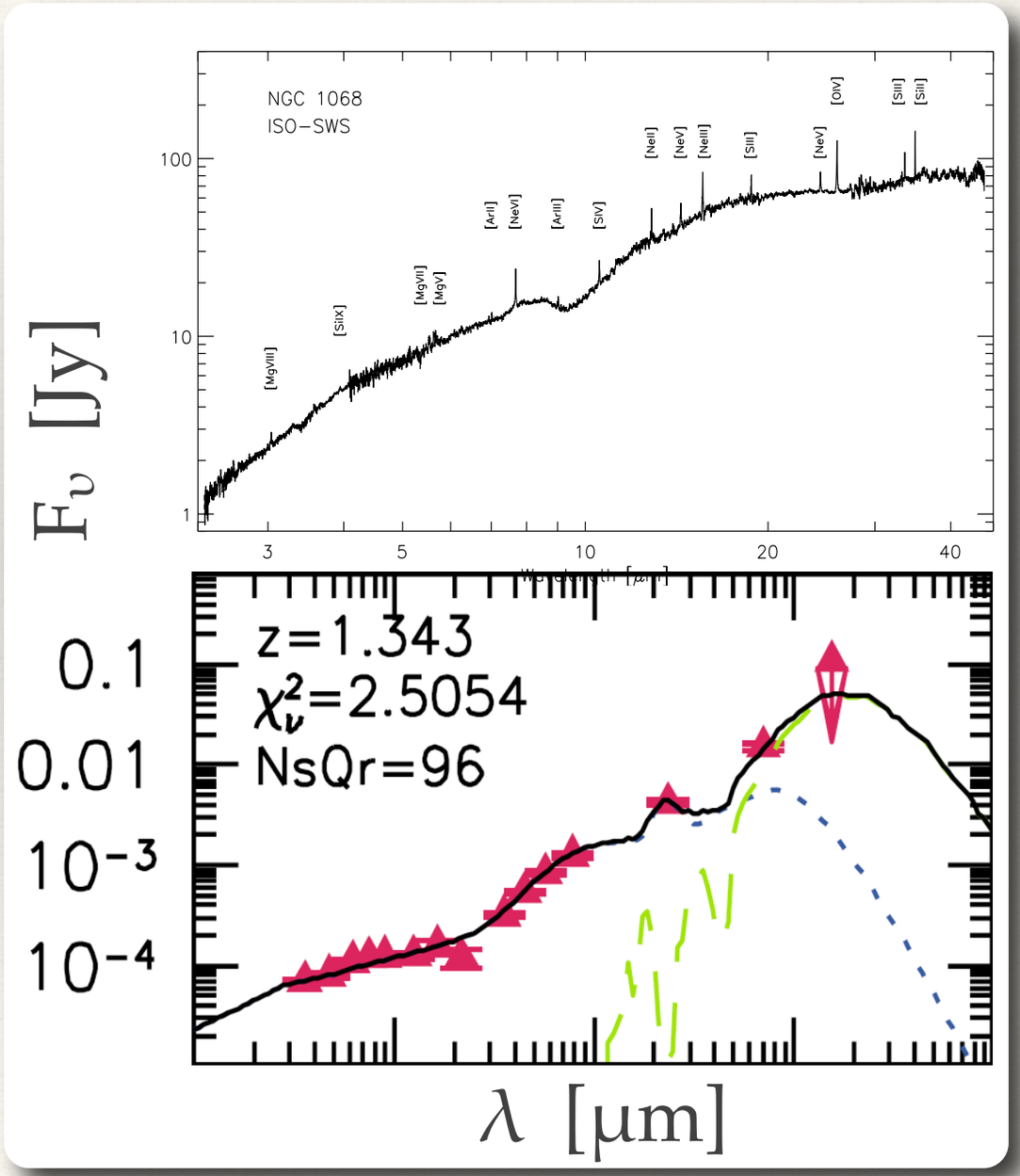
Schematic Representation

The Dusty Torus

Characteristics

Dust

- Silicate / Graphite;
- hot temperatures (up to 1500 K) ;
- very high optical depths;
- dust masses from few tens to several thousands of $M_\odot$;
- inhomogeneous distribution.

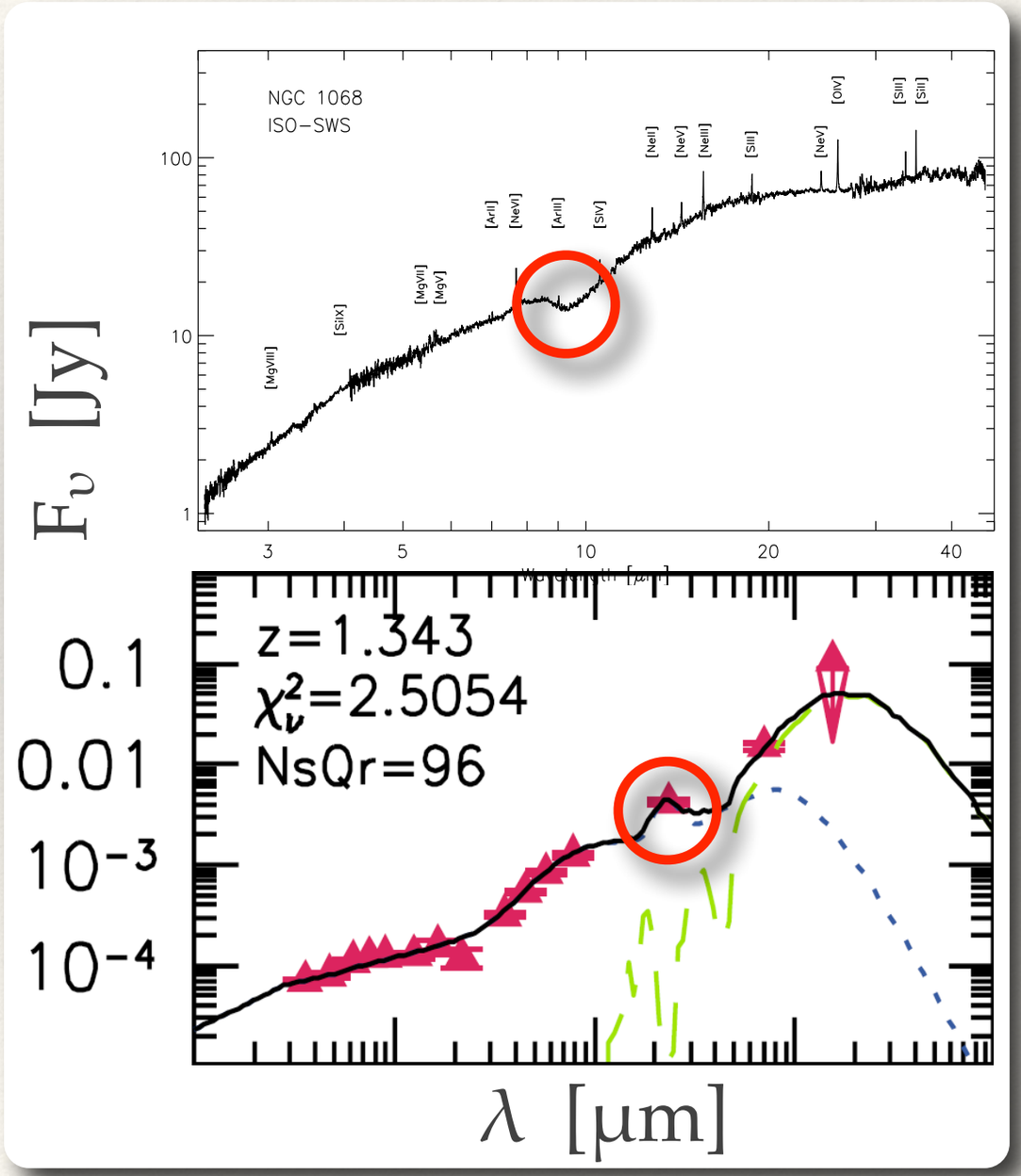


The Dusty Torus

Characteristics

Dust

- Silicate / Graphite;
- hot temperatures (up to 1500 K) ;
- very high optical depths;
- dust masses from few tens to several thousands of $M_\odot$;
- inhomogeneous distribution.

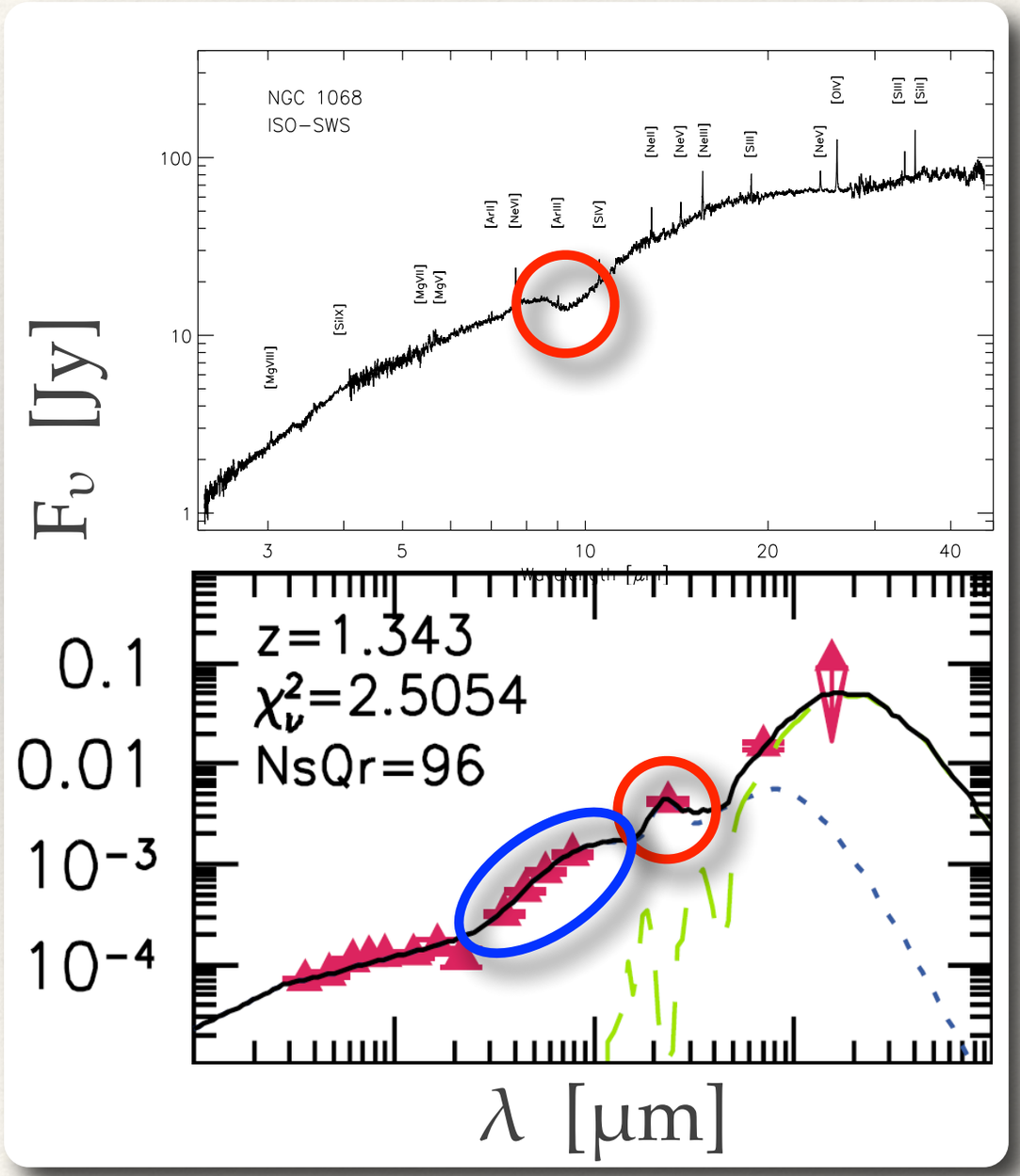


The Dusty Torus

Characteristics

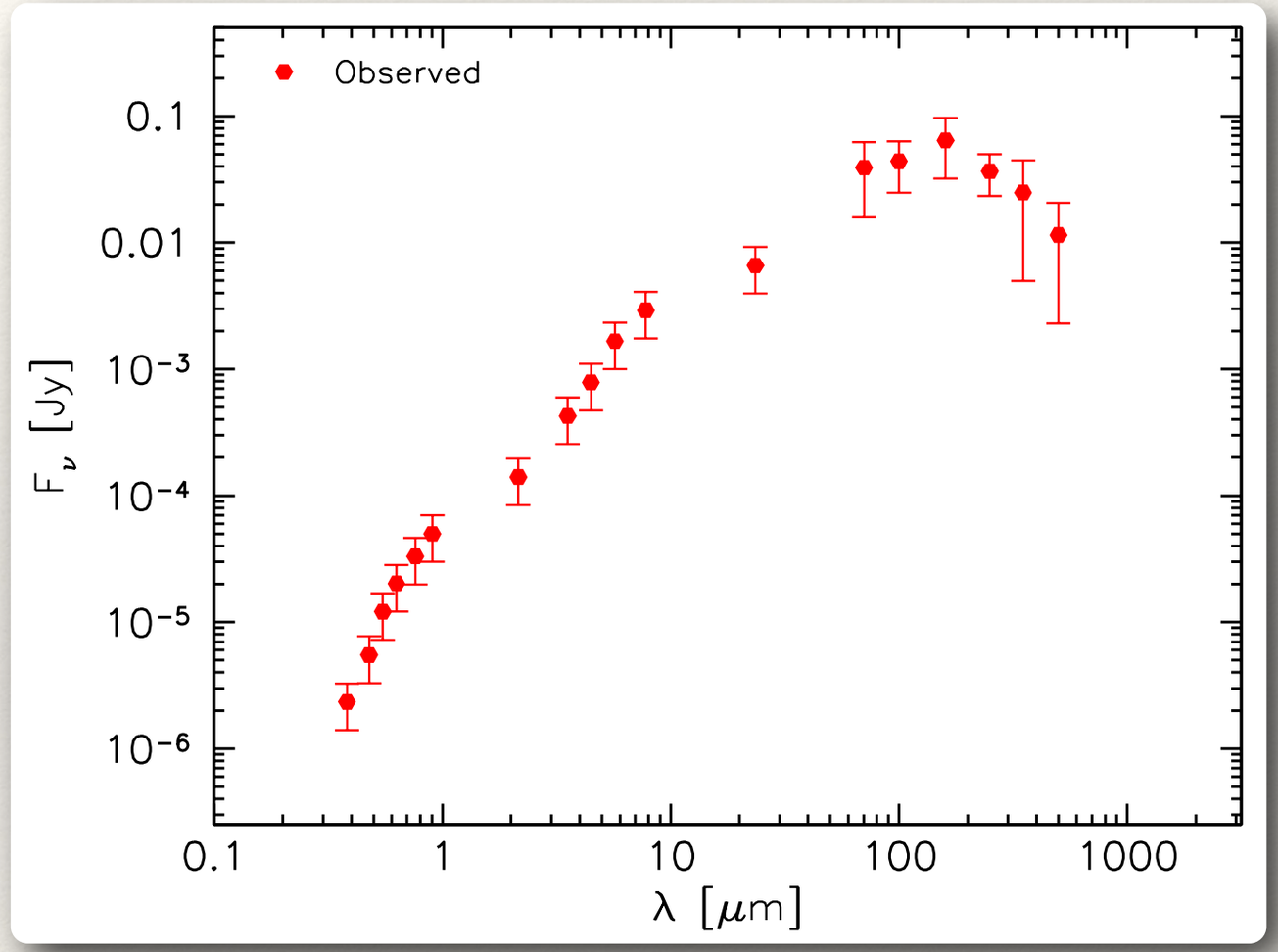
Dust

- Silicate / Graphite;
- hot temperatures (up to 1500 K) ;
- very high optical depths;
- dust masses from few tens to several thousands of $M_\odot$;
- inhomogeneous distribution.



On SED Fitting

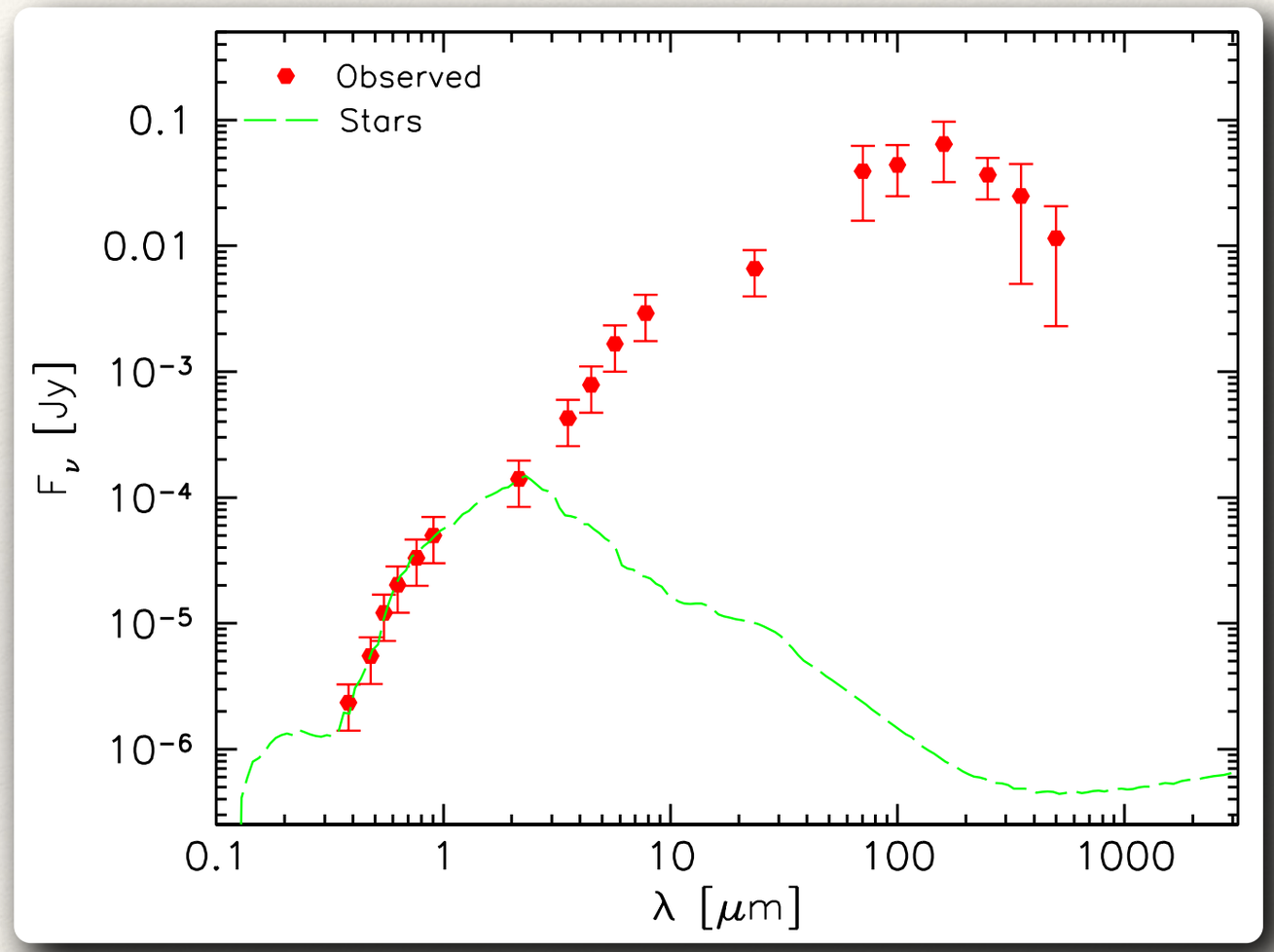
"Composite" objects:



On SED Fitting

"Composite" objects:

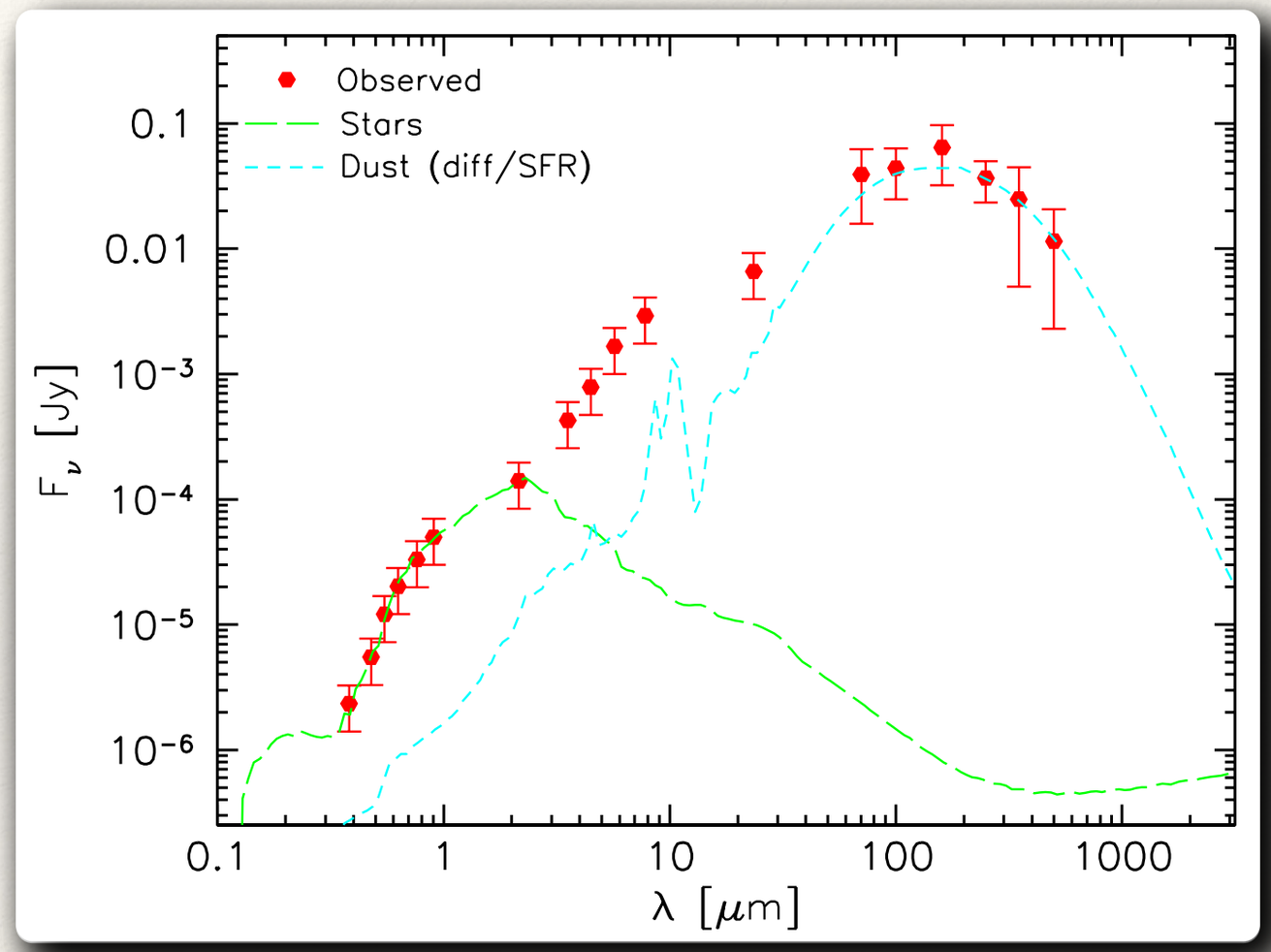
- Stars (optical)



On SED Fitting

"Composite" objects:

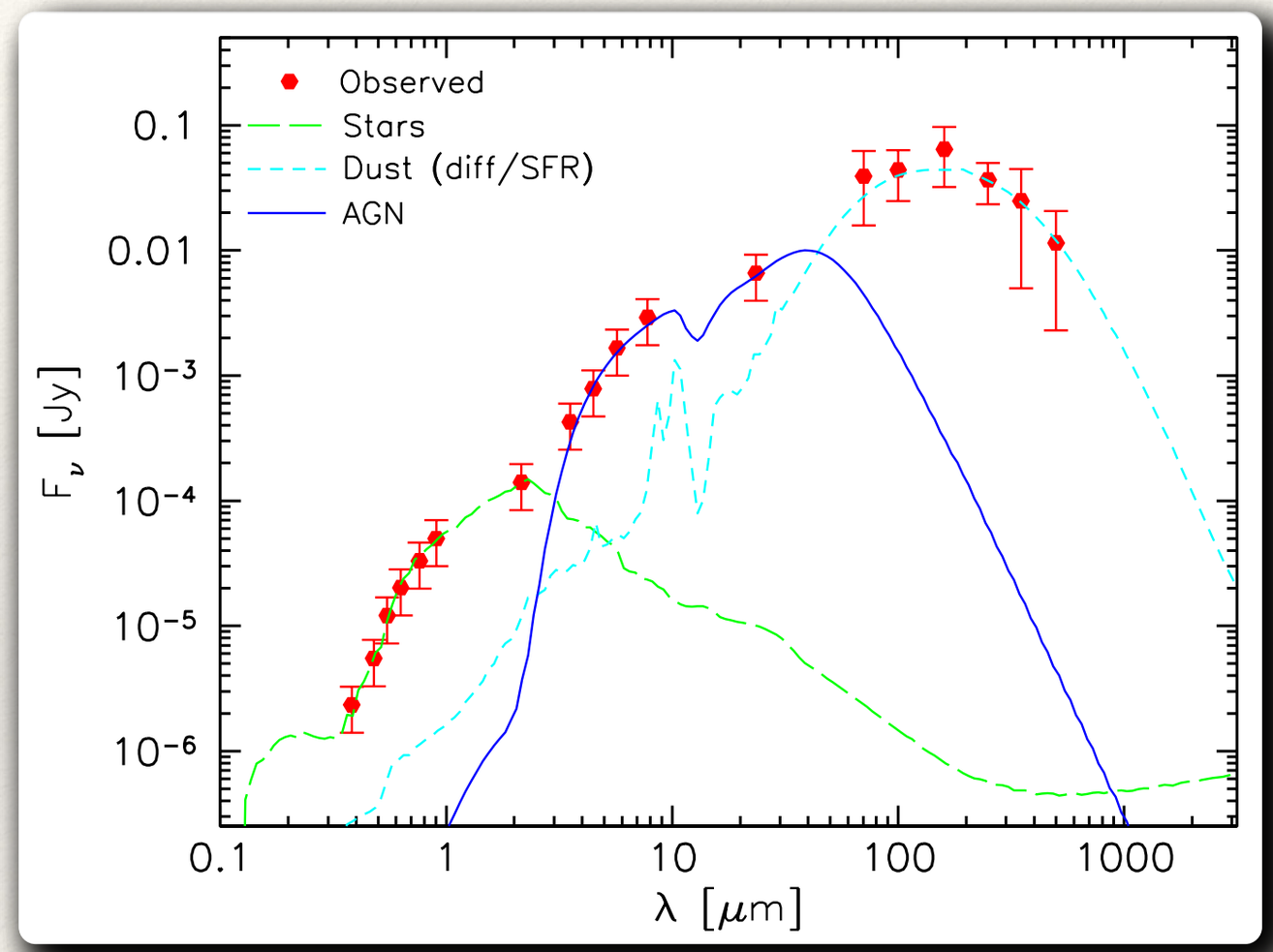
- Stars (optical)
- Dust (NIR to submm)
  - Warm/cold dust (stellar RF)



On SED Fitting

"Composite" objects:

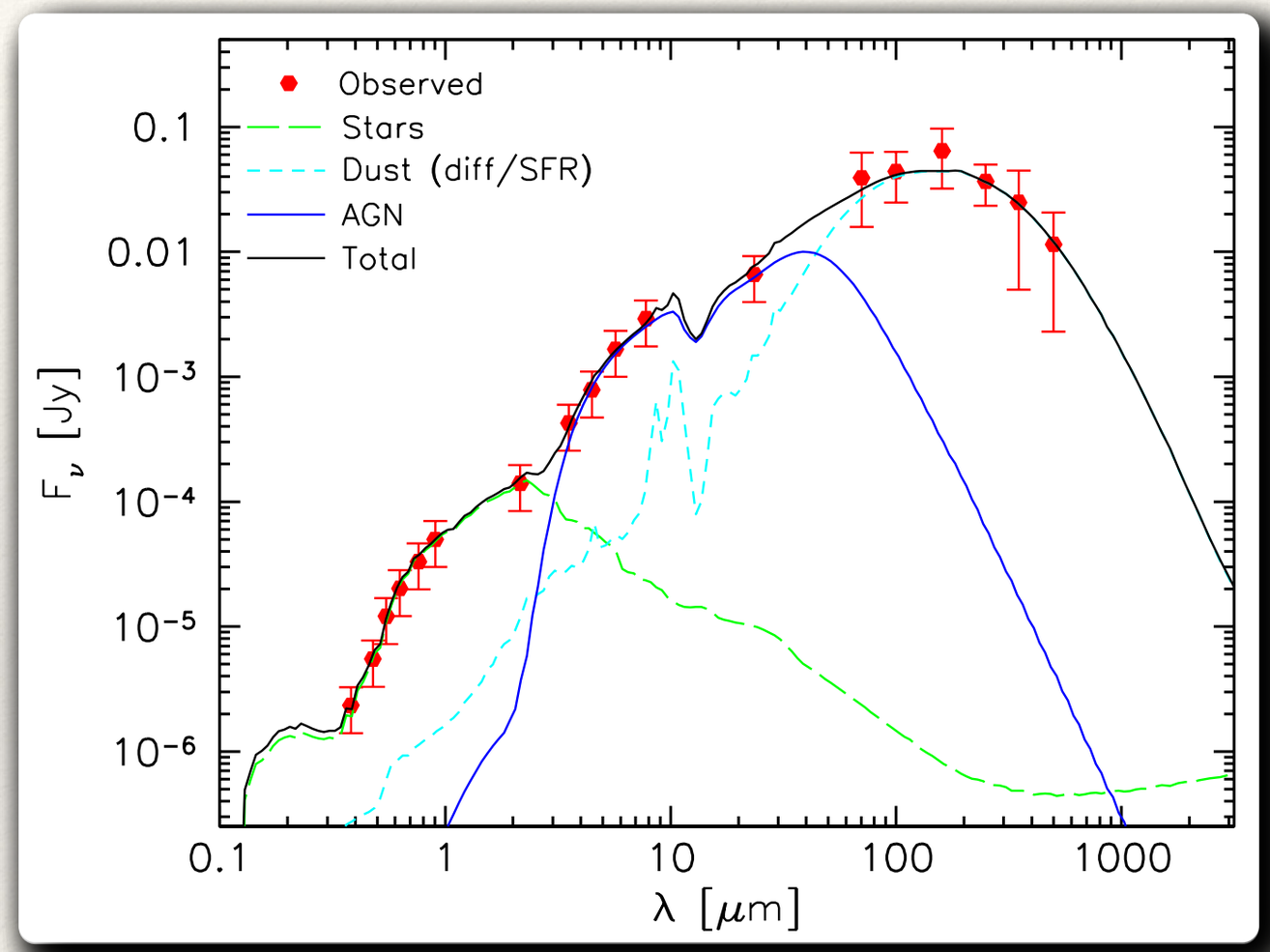
- Stars (optical)
- Dust (NIR to submm)
  - Warm/cold dust (stellar RF)
  - Hot dust (accretion disk-AGN)



On SED Fitting

"Composite" objects:

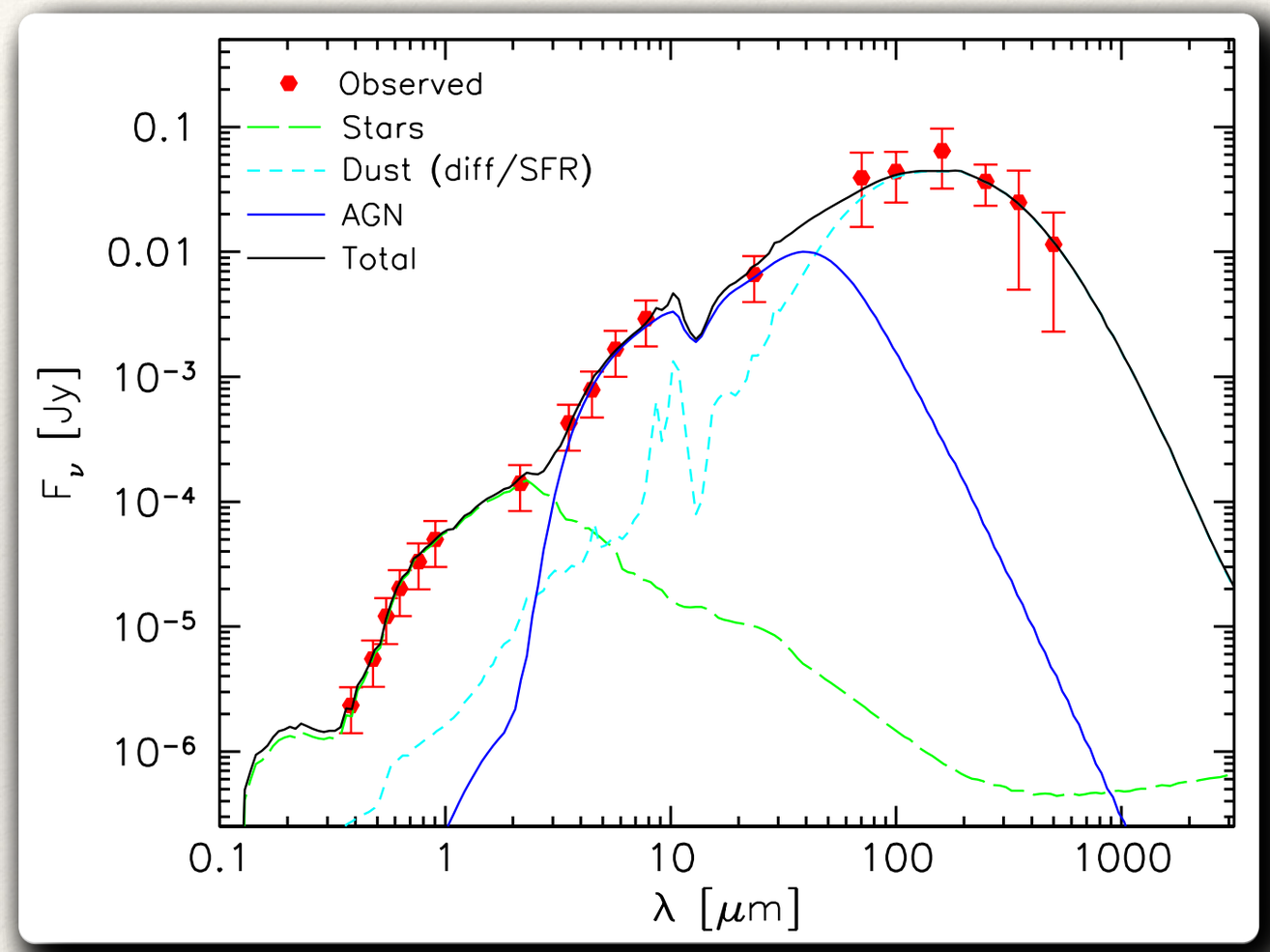
- Stars (optical)
- Dust (NIR to submm)
  - Warm/cold dust (stellar RF)
  - Hot dust (accretion disk-AGN)



On SED Fitting

"Composite" objects:

- Stars (optical)
- Dust (NIR to submm)
  - Warm/cold dust (stellar RF)
  - Hot dust (accretion disk-AGN)



Mandatory to explain the properties of many galaxies!

Understanding galaxies' properties

- What processes drive galaxy evolution?
- why do galaxies have the properties that we observe today?
- how and why did these properties evolve from what they were in the younger Universe?
- what are the relations connecting different components (stars, gas, dust, AGN, …) in a galaxy?

We use numerical models of the electromagnetic emission to try and answer these questions

RADIATIVE TRANSFER

RADIATIVE TRANSFER

- Describes the interaction between radiation and matter;

RADIATIVE TRANSFER

- Describes the interaction between radiation and matter;
- couples extinction (in the UV-optical) to emission (infrared) through <u>energy balance</u>;

RADIATIVE TRANSFER

- Describes the interaction between radiation and matter;
- couples extinction (in the UV-optical) to emission (infrared) through <u>energy balance</u>;
- can be used to simulate “real”, physical situations.

RADIATIVE TRANSFER

- Describes the interaction between radiation and matter;
- couples extinction (in the UV-optical) to emission (infrared) through <u>energy balance</u>;
- can be used to simulate “real”, physical situations.

It is possible to reconstruct the appearance of a galaxy as we would observed it at any wavelength

RADIATIVE TRANSFER

- It is a 6D problem: 3 (space) + 2 (directions) + 1 (wavelength). 100 points in each dimension yields 10^{12} points!!
- it is non linear;
- it is non-local with respect to each of the 6 dimensions:
  1. space: photons propagates in the spatial domain;
  2. direction: scattering couples intensity in all directions;
  3. wavelength: absorption/re-emission change the wavelengths.

Solving the RT problem

Extinction Term

Source Term

$$\frac{\mathrm{d}I}{\mathrm{d}s}(\mathbf{x},\mathbf{n},\lambda) = -\kappa_{\mathrm{ext}}(\mathbf{x},\lambda)\rho(\mathbf{x})I(\mathbf{x},\mathbf{n},\lambda) + j_*(\mathbf{x},\mathbf{n},\lambda) + \kappa_{\mathrm{sca}}(\mathbf{x},\lambda)\int_{4\pi}\mathbf{\Phi}(\mathbf{n},\mathbf{x}',\mathbf{x},\lambda)I(\mathbf{x},\mathbf{n}',\lambda)\mathbf{d}\Omega'$$

Scattering Term

+ dust emission
+ dust self-absorption
+ transiently-heated grains

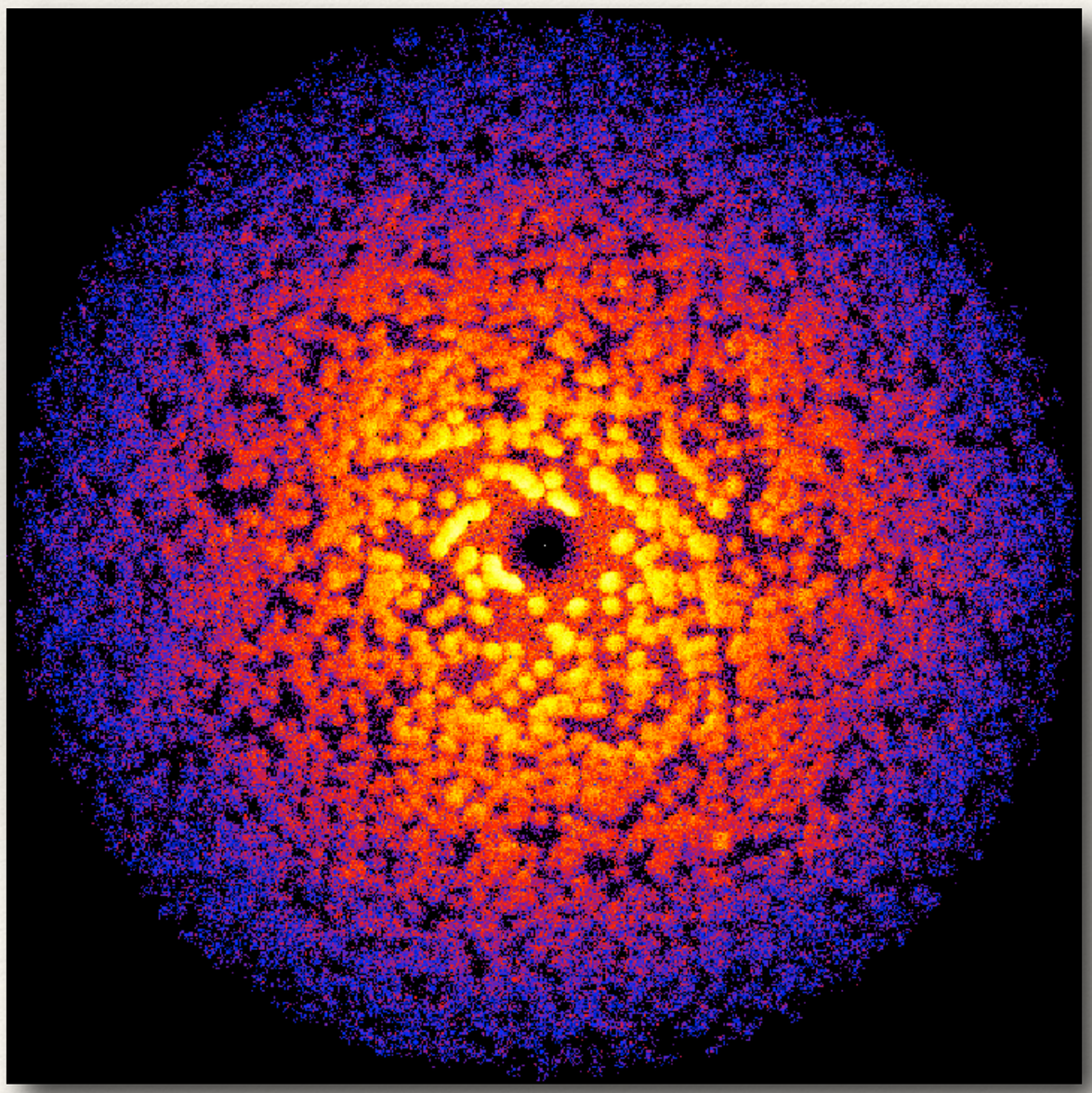
Solving the RT problem

- Probabilistic simulation method (RT equation is <u>NOT</u> solved);
- a large number of photon packages are followed individually through the dusty medium;
- the trajectory of each photon package is determined by random numbers;
- the radiation field is reconstructed by classifying photon packages by position, propagation direction, wavelength.

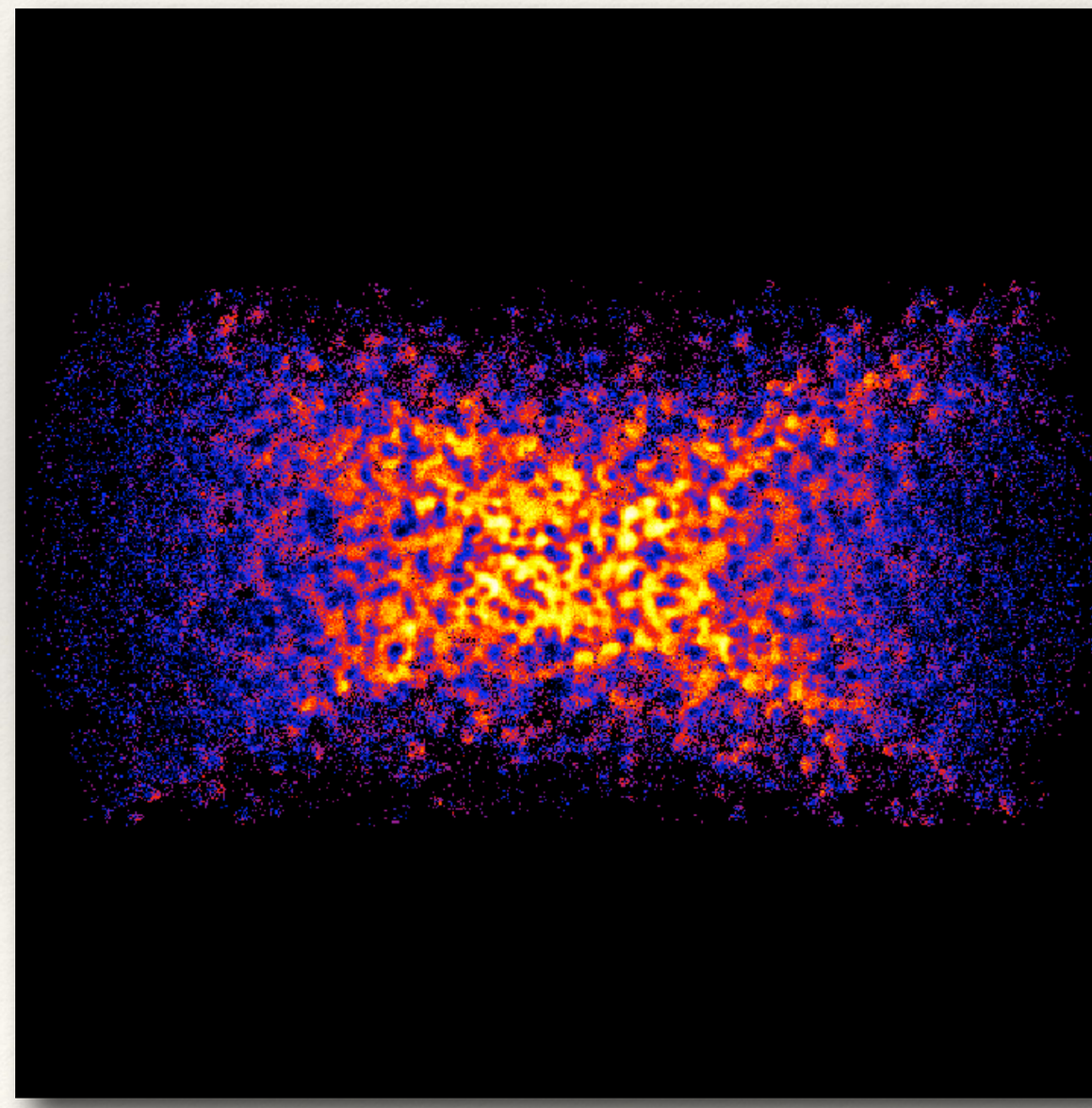
Dusty Torus Models

SKIRT MODELING

Type-1 view



Type-2 view



TASK 1

TORUS IR EMISSION MODEL

1. Use images calculated from RT models of the dusty torus emission (of a Type-1);
2. reconstruct, from the images:
    a. the total Spectral Energy Distribution;
    b. the SED of different parts of the torus.

TASK 1

- Take a type-1 torus model datacube from SKIRT;
- write a python routine to read in the model;
- calculate:
  1. the total SED (and compare to the one provided);
  2. the pure dust SED (and compare to the one provided);
  3. the AGN SED (and compare to the one provided);
  4. the dust SED of various rings.

On the models' name convention

Exponent of the angular density low

Outer-to-inner torus radius

Optical depth @ 9.7 μm

Viewing angle

`t5_p1_q1_oa30_R10_Mcl0.97_i0`

Exponent of the radial density low

Torus opening angle

Mass fraction of dust in clouds

The content of the files

t5_p1_q1_oa30_R10_Mcl0.97_i0 +

_ds_tempxy.fits

_ds_tempxz.fits

_ds_tempyz.fits

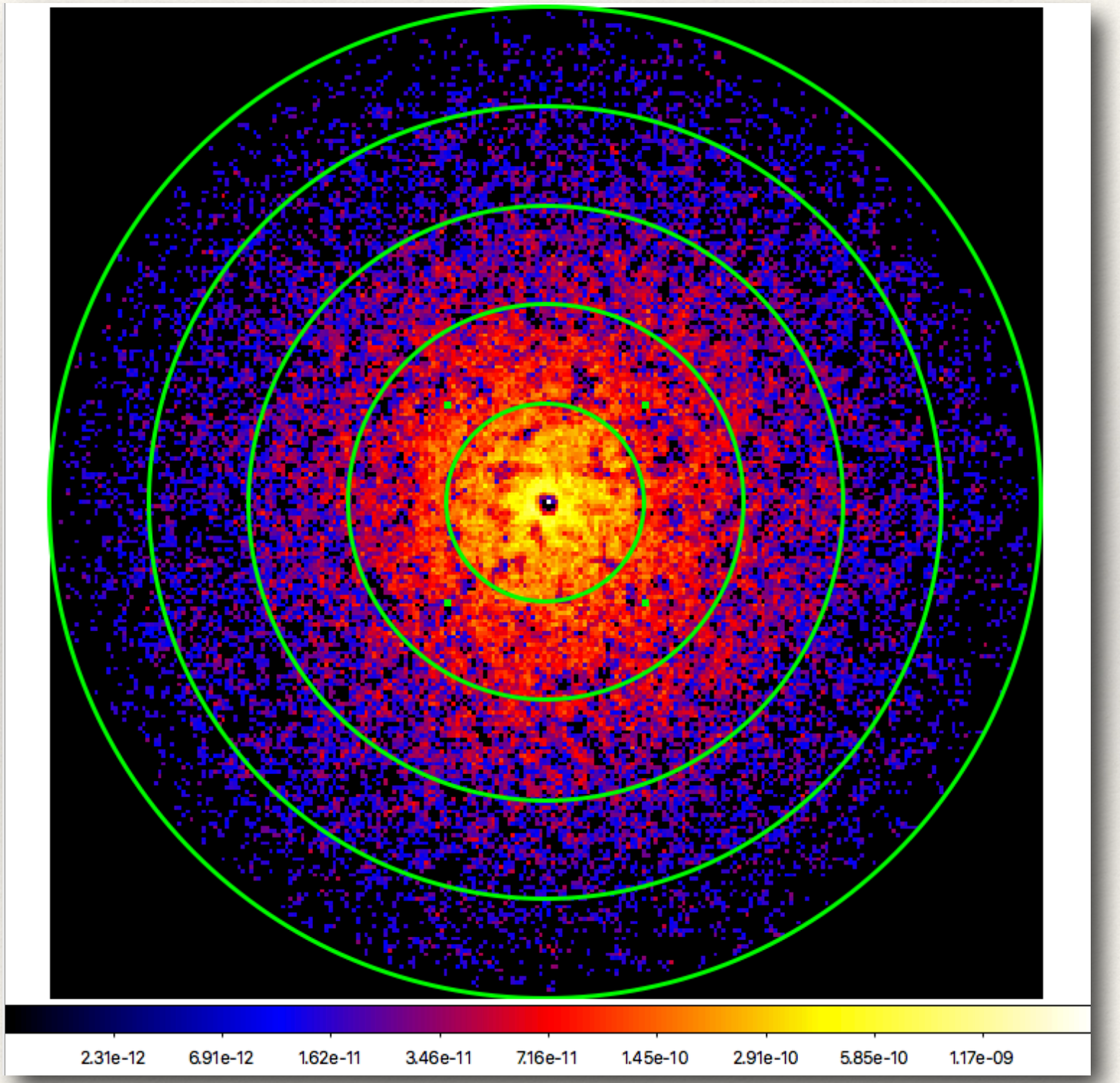
_i0_total.fits

_i0_sed.dat

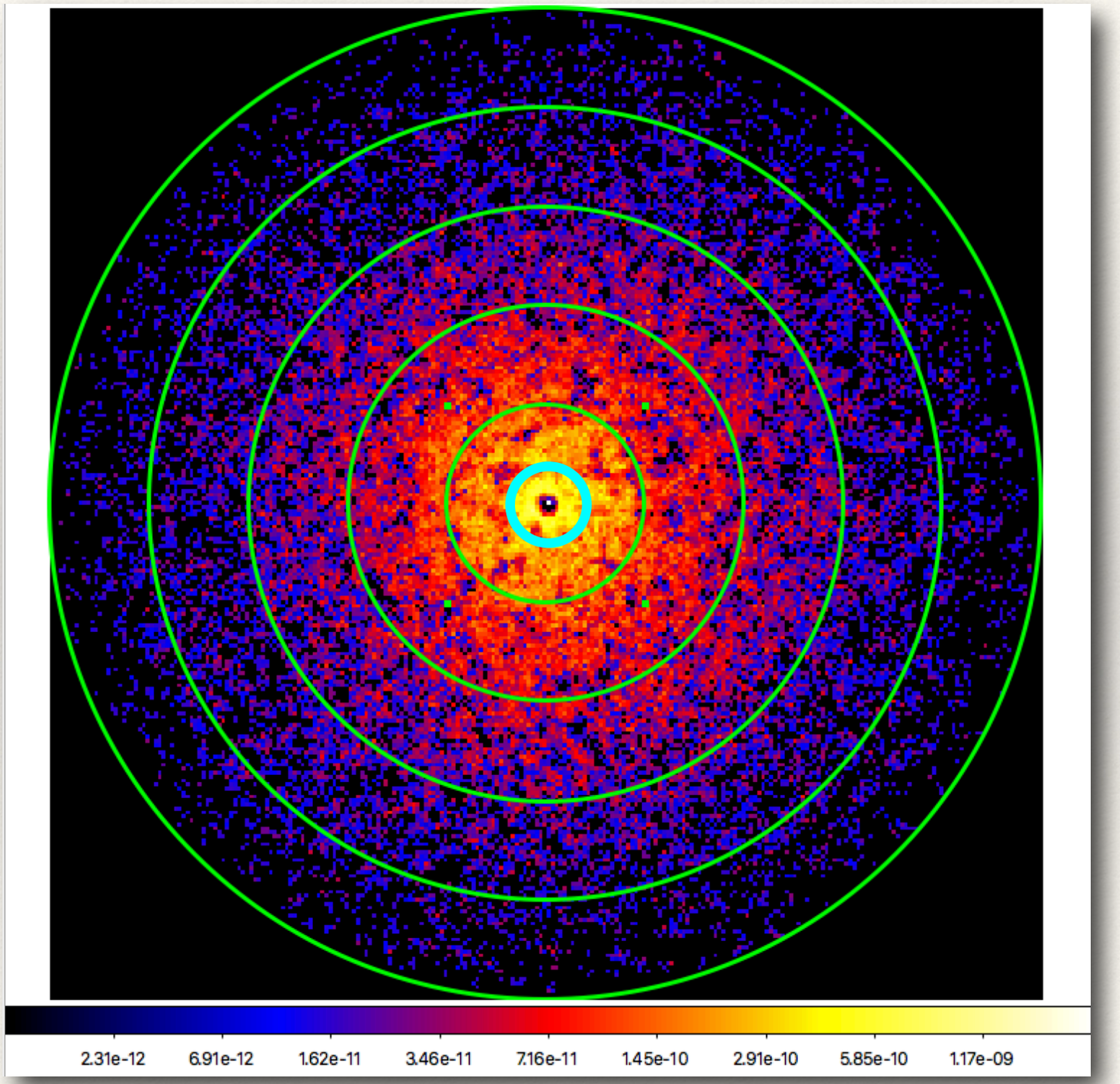
Let's get started!

- Open the "`total`" fits file within DS9;
- choose an appropriate scale (*log* or *sqrt* should do) and color;
- also, go to a wavelength plane with high enough emission;
- using *regions*, and adopting a circular region, enclose the full circle of emission;
- define other concentric regions at different radii;
- define a region for the innermost, dust free, region (disk);
- save all the regions in *image coordinates*.

Regions Definition

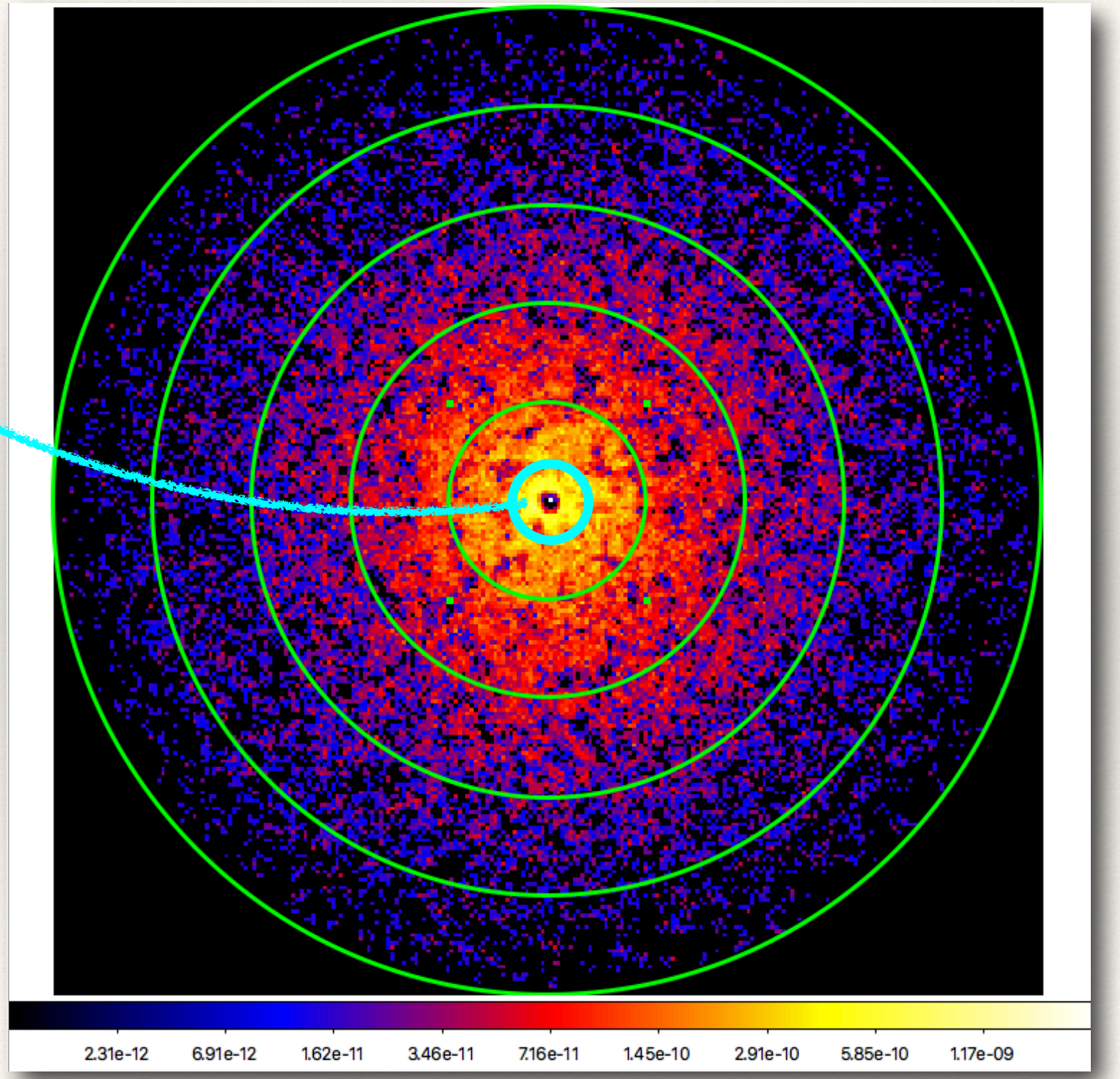


Regions Definition



Regions Definition

Accretion
disk emission



Open a datacube and extract infos

```
#!/usr/bin/env python
# -*- coding: utf8 -*-
import numpy as np
import astropy.io.fits as fits
import pyregion
import pyregion._region_filter as filter

modelname = “filename and path here”
hdulist = fits.open(modelname)
header = (hdulist[0].header)
cube = (hdulist[0].data[:,:,:])
nx = header["NAXIS1"]
ny = header["NAXIS2"]
nl = header["NAXIS3"]
```

Use to run python directly from the terminal

Need modules

Loads the fits structure onto a variable

Saves the header only

Saves the data only

This is the way we extract header information

Open a datacube and extract infos

```
#!/usr/bin/env python
# -*- coding: utf8 -*-
import numpy as np
import astropy.io.fits as fits
import pyregion
import pyregion._region_filter as filter
```

Use to run python directly from the terminal

Need modules

```
modelname = “filename and path here”
hdulist = fits.open(modelname)
header = (hdulist[0].header)
cube = (hdulist[0].data[:,:,:])
nx = header["NAXIS1"]
ny = header["NAXIS2"]
nl = header["NAXIS3"]
```

Loads the fits structure onto a variable

Saves the header only

Saves the data only

This is the way we extract header information

The structure of the fits file can be seen with: `hdulist.info()`

Calculating the SED (total+regions)

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
    sflx = 0.0
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
    sflx = 0.0
    for m in range(nl):
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
    sflx = 0.0
    for m in range(nl):
        for i in range(nx):
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
    sflx = 0.0
    for m in range(nl):
        for i in range(nx):
            for n in range(ny):
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
    sflx = 0.0
    for m in range(nl):
        for i in range(nx):
            for n in range(ny):
                if mask[n,i] == True:
```

Calculating the SED (total+regions)

```
source =  pyregion.open("regions.reg")
nreg = len(source)
size = (ny, nx)
tot_sed = np.zeros((nl,nreg))
for k in range(nreg):
    print "Region ",k
    source =  pyregion.open("regions.reg")
    del source[0:k]
    del source[k+1:nreg]
    mask = source.get_mask(shape=size)
    sflx = 0.0
    for m in range(nl):
        for i in range(nx):
            for n in range(ny):
                if mask[n,i] == True:
                    tot_sed[m,k] = tot_sed[m,k] + cube[m,n,i]
```

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

```
fmt = "%12.4e  %12.4e  %12.4e  %12.4e  %12.4e \n"
```

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

```
fmt = "%12.4e  %12.4e  %12.4e  %12.4e  %12.4e \n"
ff = open("sed.dat",'w')
```

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

```
fmt = "%12.4e  %12.4e  %12.4e  %12.4e  %12.4e \n"
ff = open("sed.dat",'w')
ff.write("reg1  reg2  reg3  reg4 tot \n")
```

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

```
fmt = "%12.4e  %12.4e  %12.4e  %12.4e  %12.4e \n"
ff = open("sed.dat",'w')
ff.write("reg1  reg2  reg3  reg4 tot \n")
for v in zip(sed1, sed2, sed3, sed4, sedt):
```

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

```
fmt = "%12.4e  %12.4e  %12.4e  %12.4e  %12.4e \n"
ff = open("sed.dat",'w')
ff.write("reg1  reg2  reg3  reg4 tot \n")
for v in zip(sed1, sed2, sed3, sed4, sedt):
    ff.write(fmt%v)
```

Writing the data out

Here […] calculate the SED in rings, NOT in the regions (and subtract the primary emission!!)

```
fmt = "%12.4e  %12.4e  %12.4e  %12.4e  %12.4e \n"
ff = open("sed.dat",'w')
ff.write("reg1  reg2  reg3  reg4 tot \n")
for v in zip(sed1, sed2, sed3, sed4, sedt):
    ff.write(fmt%v)
ff.close()
```

Reading text files in python

Reading text files in python

```python
import astropy.io.ascii as ascii
```

Reading text files in python

```python
import astropy.io.ascii as ascii
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
tdata = ascii.read(sedfile)
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
tdata = ascii.read(sedfile)
lamb = tdata['col1']
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
tdata = ascii.read(sedfile)
lamb = tdata['col1']
tflx = tdata['col2']
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
tdata = ascii.read(sedfile)
lamb = tdata['col1']
tflx = tdata['col2']
agn = tdata['col3']
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
tdata = ascii.read(sedfile)
lamb = tdata['col1']
tflx = tdata['col2']
agn = tdata['col3']
```

Reading text files in python

```
import astropy.io.ascii as ascii

sedfile = “_sed.dat model file here”
tdata = ascii.read(sedfile)
lamb = tdata['col1']
tflx = tdata['col2']
agn = tdata['col3']
```

Note: when comparing fluxes from the “`_sed.dat`” file, and the ones calculated from the image, check the units!!!

Units of the data

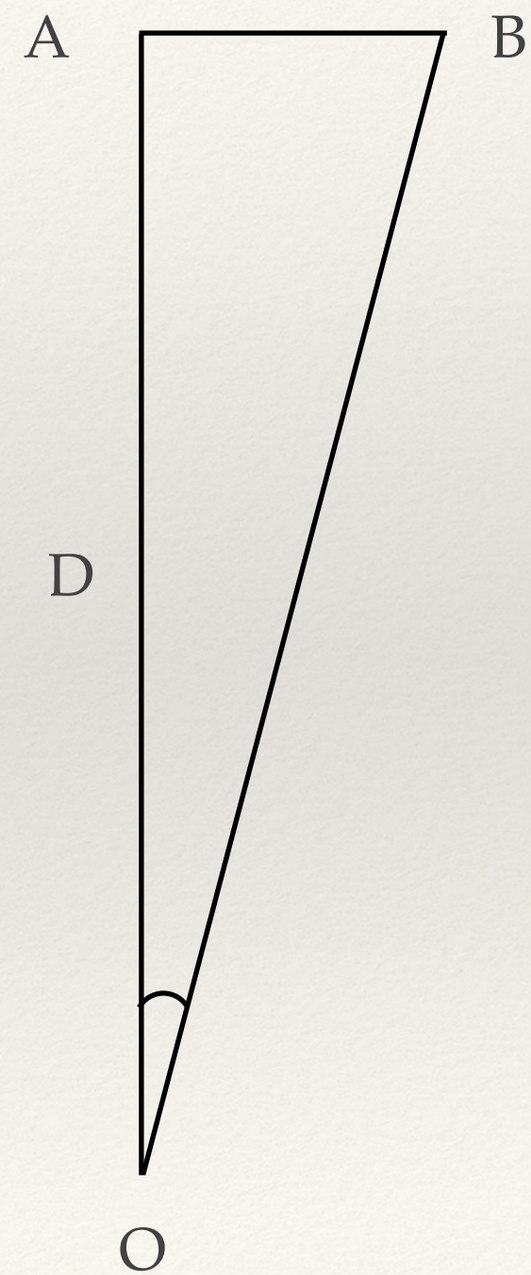
t5_p0_q0_oa30_R10_Mcl0.97_i0_total.fits

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                  -32 / number of bits per data pixel
NAXIS   =                    3 / number of data axes
NAXIS1  =                  600 / length of data axis 1
NAXIS2  =                  600 / length of data axis 2
NAXIS3  =                  119 / length of data axis 3
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
BSCALE  =                   1.
BZERO   =                   0.
DATE    = '2018-08-07T11:06:00' / Date and time of creation (UTC)
ORIGIN  = 'SKIRT simulation'    / Astronomical Observatory, Ghent University
BUNIT   = 'W/m2/arcsec2'        / Physical unit of the array values
CRPIX1  =                 300.5 / X-axis coordinate system reference pixel
CRVAL1  =                    0. / Coordinate system value at X-axis reference pix
CDELT1  =   0.0166666666666667 / Coordinate increment along X-axis
CTYPE1  = 'pc      '            / Physical units of the X-axis increment
CRPIX2  =                 300.5 / Y-axis coordinate system reference pixel
CRVAL2  =                    0. / Coordinate system value at Y-axis reference pix
CDELT2  =   0.0166666666666667 / Coordinate increment along Y-axis
CTYPE2  = 'pc      '            / Physical units of the Y-axis increment
END
```

```
# column 1: lambda (micron)
# column 2: total flux; lambda*F_lambda (W/m2)
# column 3: direct stellar flux; lambda*F_lambda (W/m2)
# column 4: scattered stellar flux; lambda*F_lambda (W/m2)
# column 5: total dust emission flux; lambda*F_lambda (W/m2)
# column 6: dust emission scattered flux; lambda*F_lambda (W/m2)
# column 7: transparent flux; lambda*F_lambda (W/m2)
```

How do we convert?

- W m^{-2} $arcsec^{-2}$ → W m^{-2};
- what is the pixelsize in units of arcsec?
- note that the models have been calculated for a distance D of 10 Mpc.



Plotting SEDs

```
import matplotlib.pyplot as plt
from matplotlib import rc, rcParams
rc('text', usetex=True)
```

Plotting SEDs

```
yma = max(tflx)*1.2
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
plt.ylabel(r'$\lambda$F$_\lambda$   [W m$^{-2}$]',fontsize=20)
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
plt.ylabel(r'$\lambda$F$_\lambda$   [W m$^{-2}$]',fontsize=20)
plt.tick_params(labelsize=20)
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
plt.ylabel(r'$\lambda$F$_\lambda$   [W m$^{-2}$]',fontsize=20)
plt.tick_params(labelsize=20)
plt.xlim(xmi,xma)
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
plt.ylabel(r'$\lambda$F$_\lambda$   [W m$^{-2}$]',fontsize=20)
plt.tick_params(labelsize=20)
plt.xlim(xmi,xma)
plt.ylim(ymi,yma)
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
plt.ylabel(r'$\lambda$F$_\lambda$   [W m$^{-2}$]',fontsize=20)
plt.tick_params(labelsize=20)
plt.xlim(xmi,xma)
plt.ylim(ymi,yma)
fig = plt.gcf()
```

Plotting SEDs

```
yma = max(tflx)*1.2
ymi = yma*0.001
xma = max(lamb)*1.01
xmi = min(lamb)*0.97
plt.figure(figsize=(15,10))
plt.loglog(lamb,tflx,linestyle='-',color='black')
plt.loglog(lamb,tot,linestyle='--',color='black',linewidth=4)
plt.loglog(lamb,reg1,linestyle='-',color='green')
plt.xlabel(r'$\lambda$    [$\mu$m]',fontsize=20)
plt.ylabel(r'$\lambda$F$_\lambda$   [W m$^{-2}$]',fontsize=20)
plt.tick_params(labelsize=20)
plt.xlim(xmi,xma)
plt.ylim(ymi,yma)
fig = plt.gcf()
plt.show()
```

Final comparison

- Take the "`####_sed.dat`" file, and compare:
    1. the total SED;
    2. the dust SED;
- compare the SEDs of the regions at the various radii/distances;
- let's compare various different models (different parameters):
- Everyone make their plots, showing the SED and the parameters;
- Discussion.

Summary

- Historical overview on Active Galactic Nuclei;
- physical and geometrical properties;
- the AGN Unified Scheme;
- the dusty torus;
- galaxy properties derivation through SED fitting;
- radiative transfer;
- working on torus emission models.