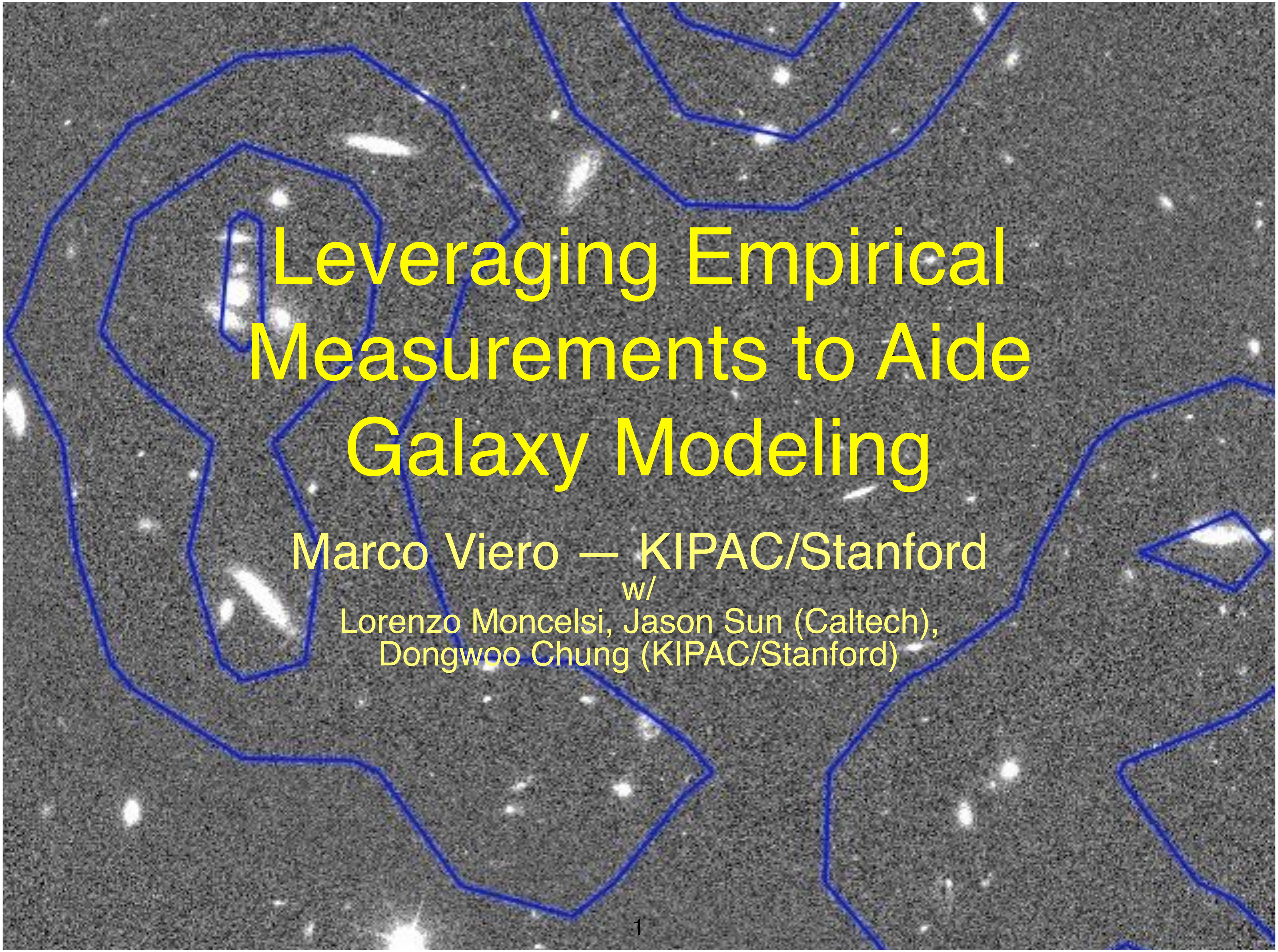


Leveraging Empirical Measurements to Aide Galaxy Modeling

Marco Viero — KIPAC/Stanford
w/
Lorenzo Moncelsi, Jason Sun (Caltech),
Dongwoo Chung (KIPAC/Stanford)

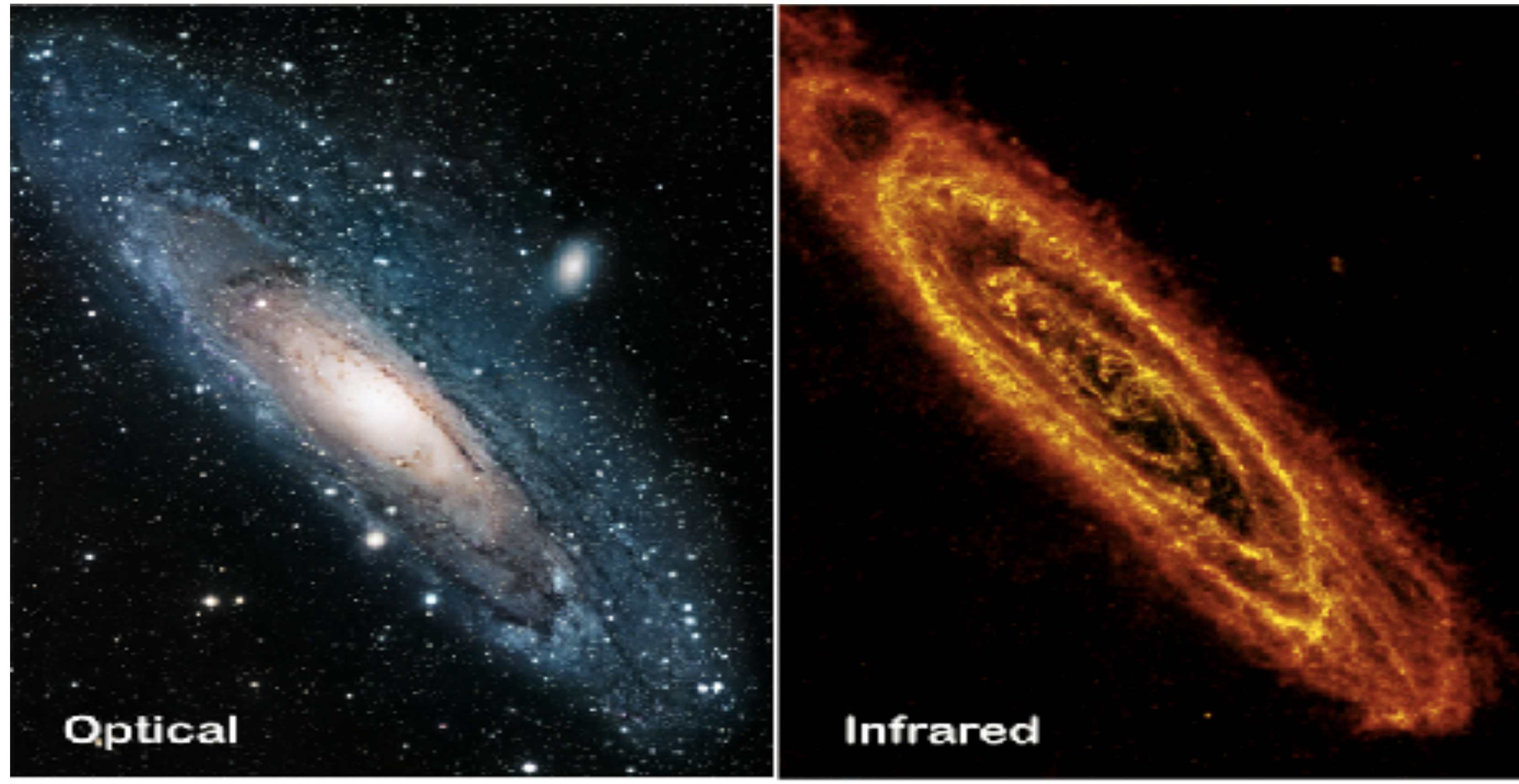
Outline

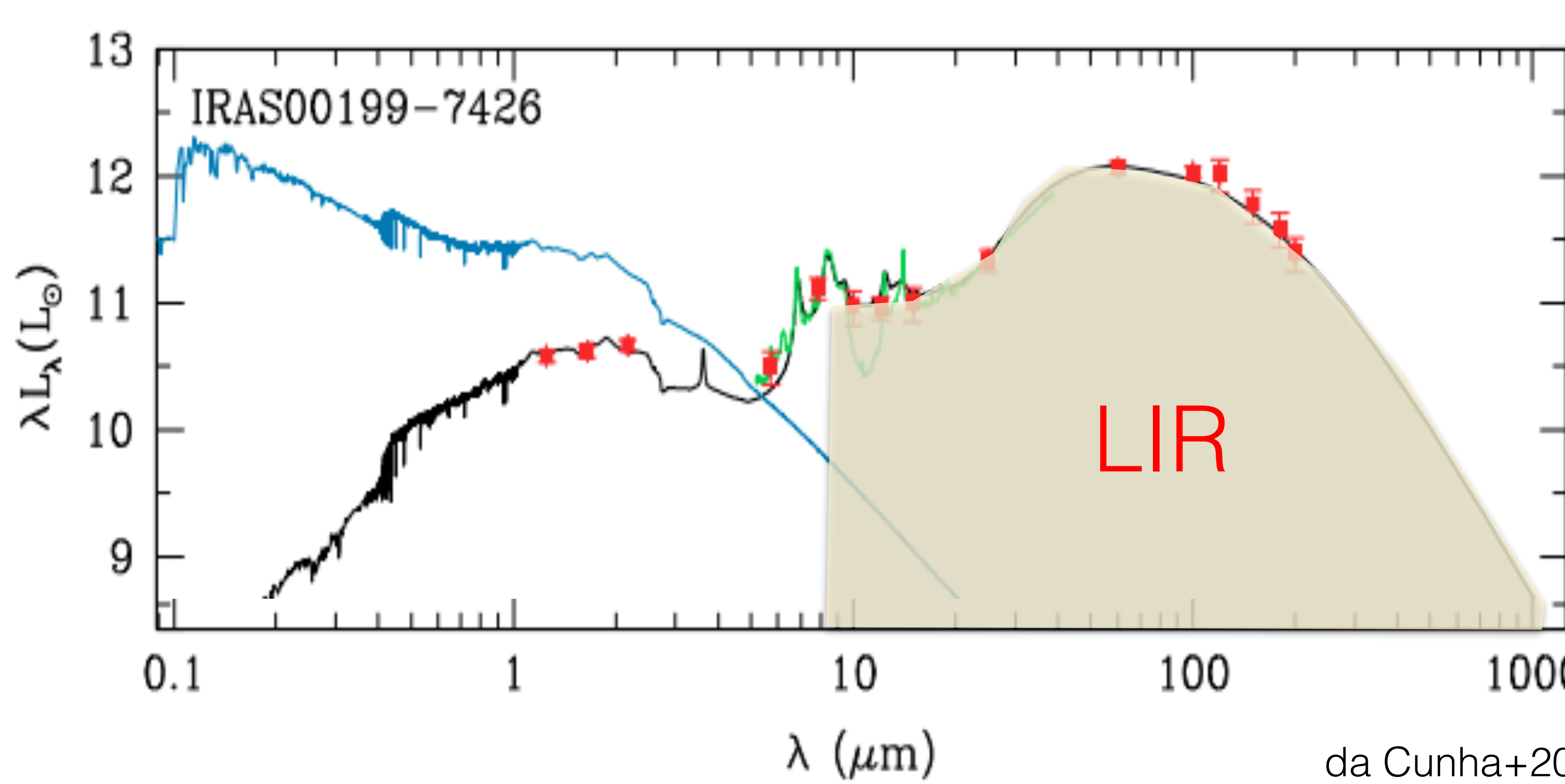
- Models outlined here all need some form of star-formation rate or bolometric infrared luminosity (SFR or LIR).

- Why not pull straight from measurements?
  - Short answer: *it's hard*.
  - Less-short answer: it's not that hard, and look, I've done half of the work for you.

Challenge

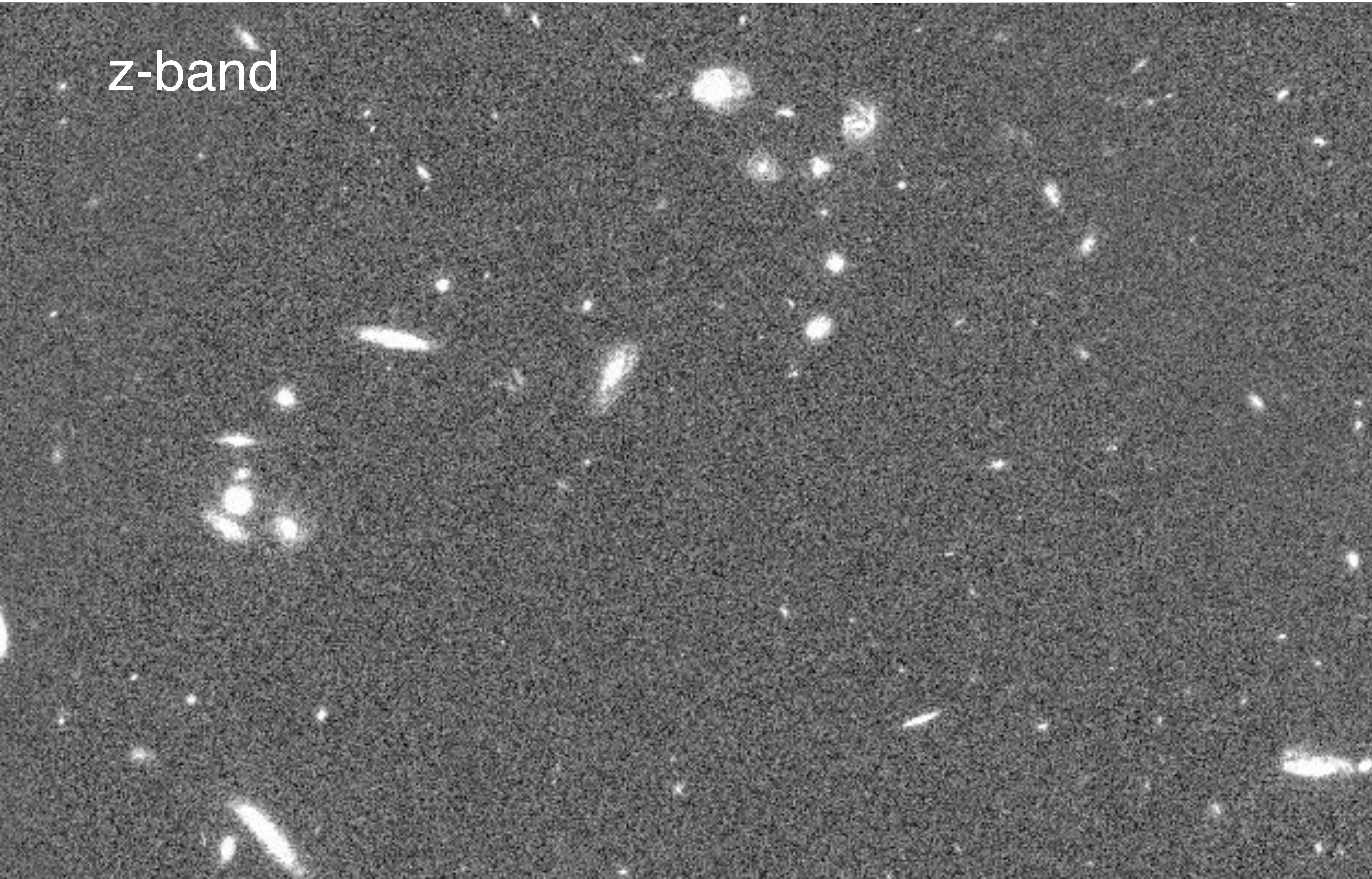
- Infrared/Submillimeter emission reprocessed starlight by dust
- IR/Submm traces star formation
- Half the emission is tied up in dust
  - ➡We want to quantify the optical-LIR connection



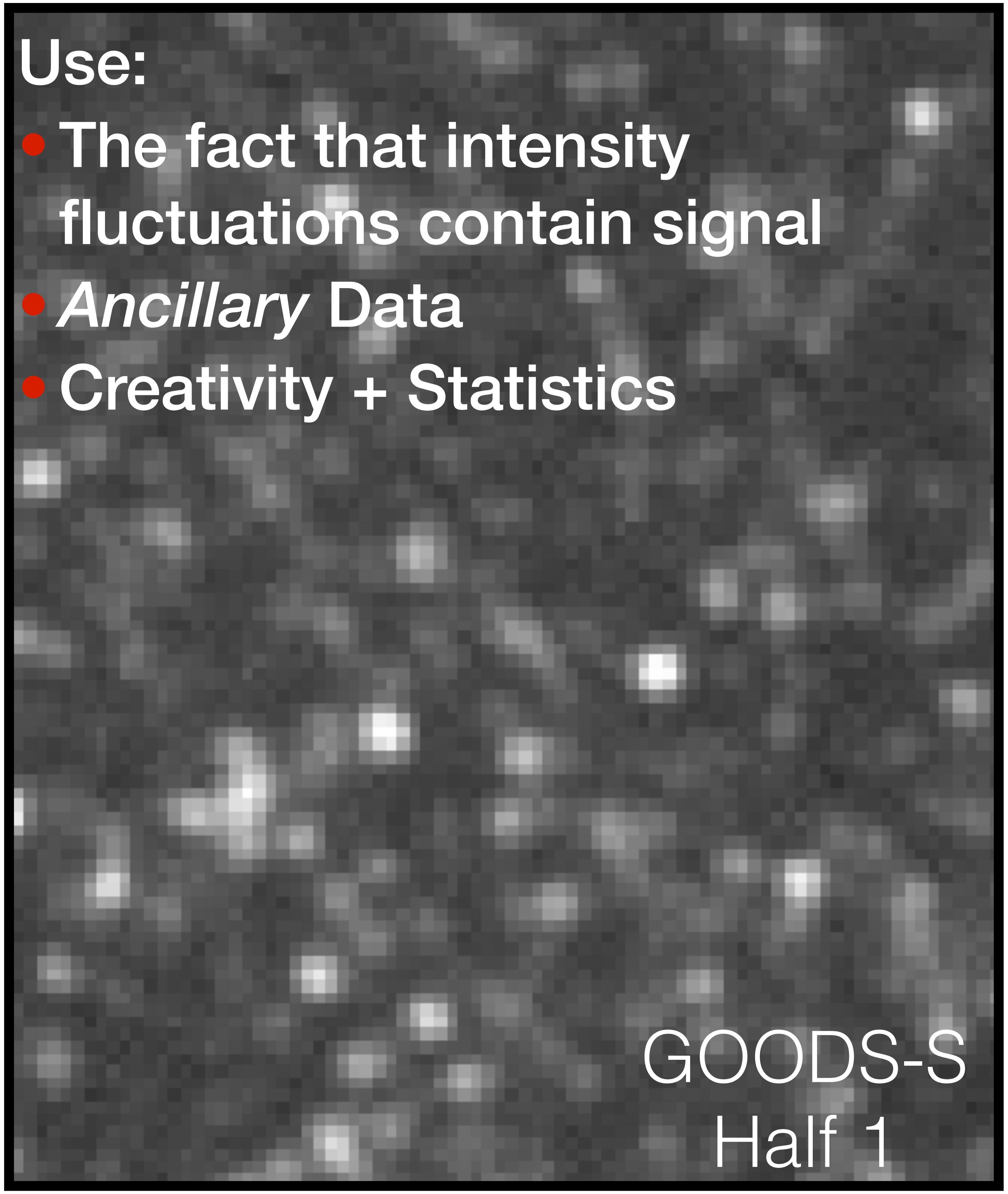


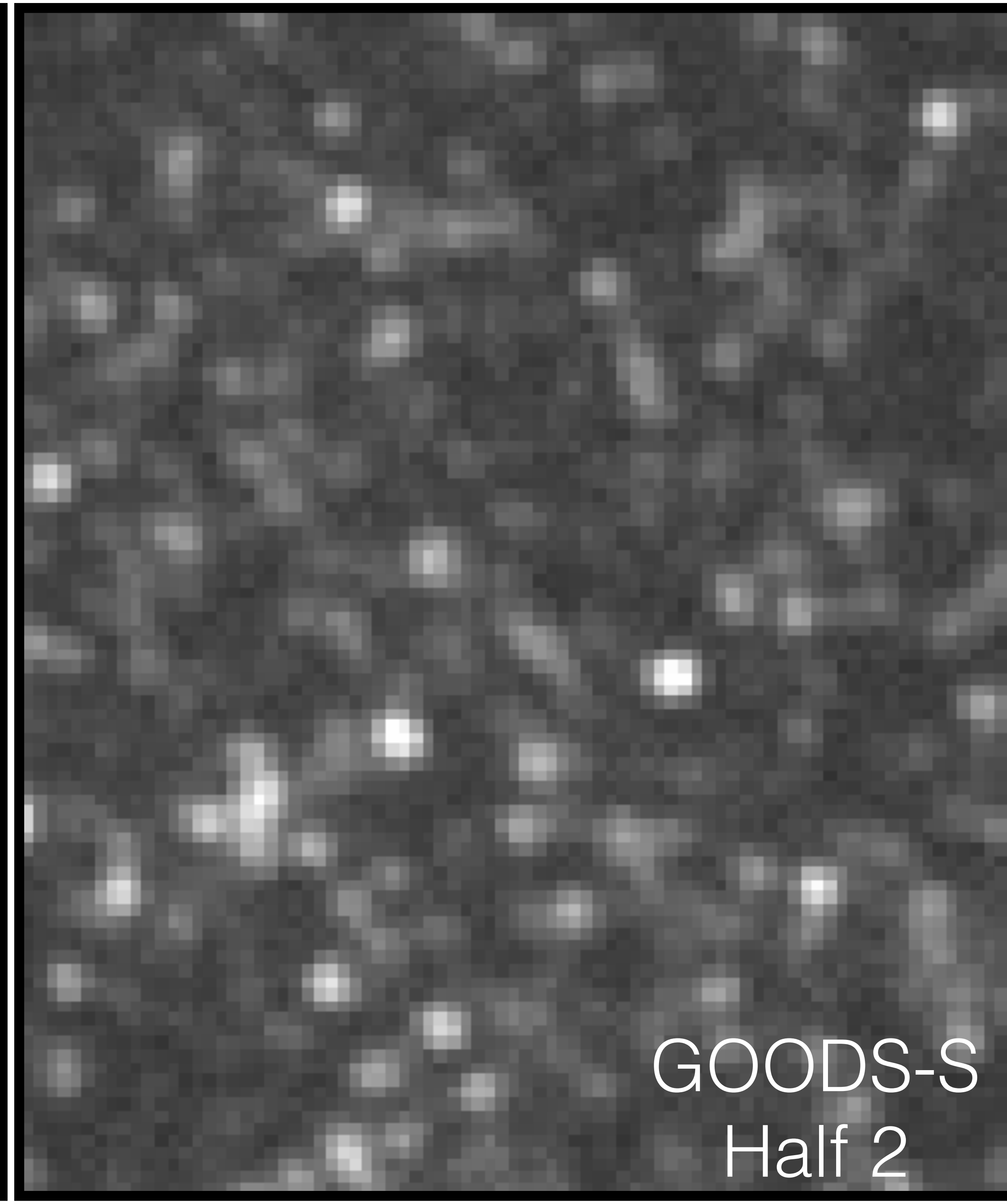
da Cunha+2010

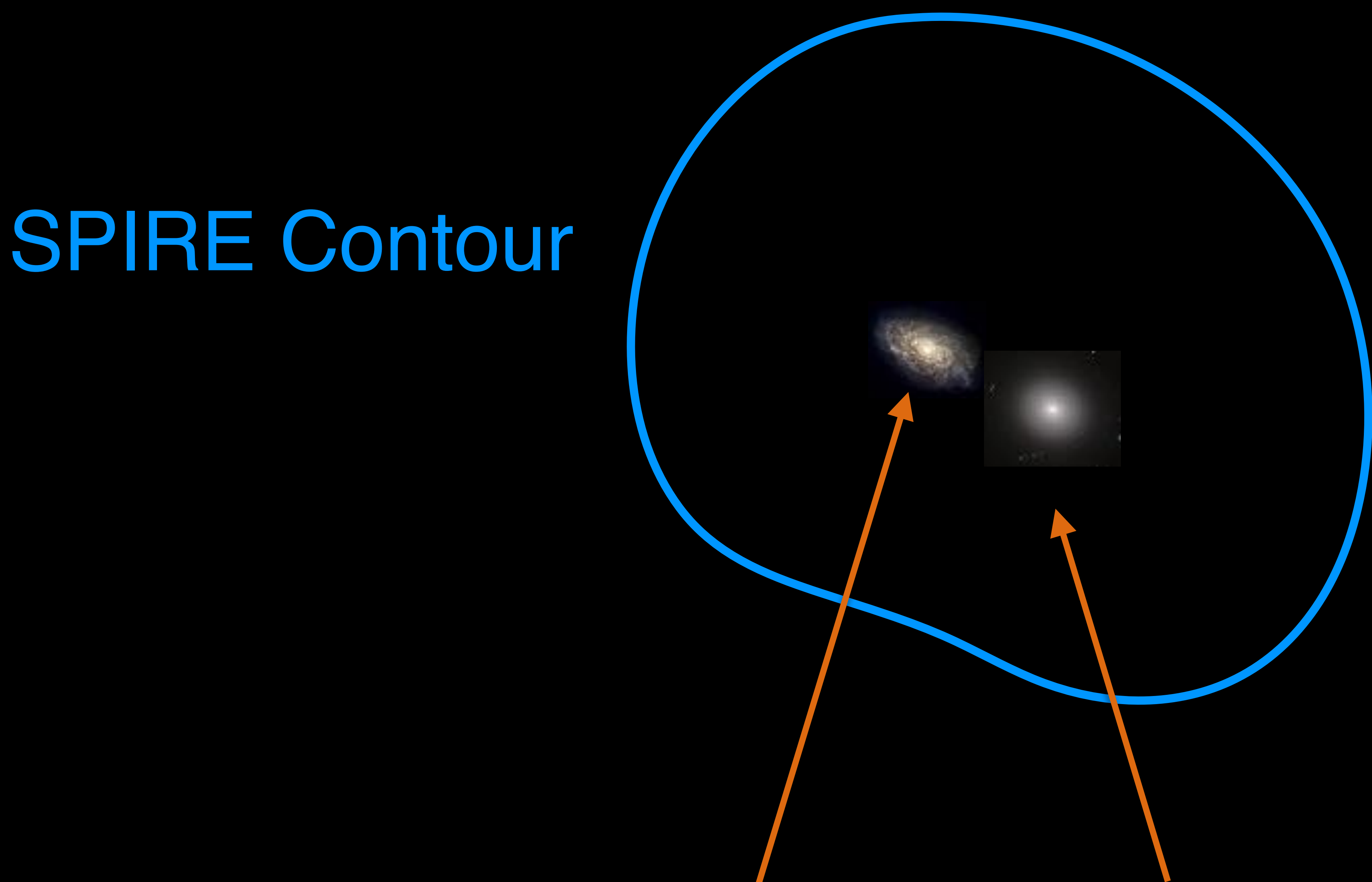
Challenge — Source Confusion



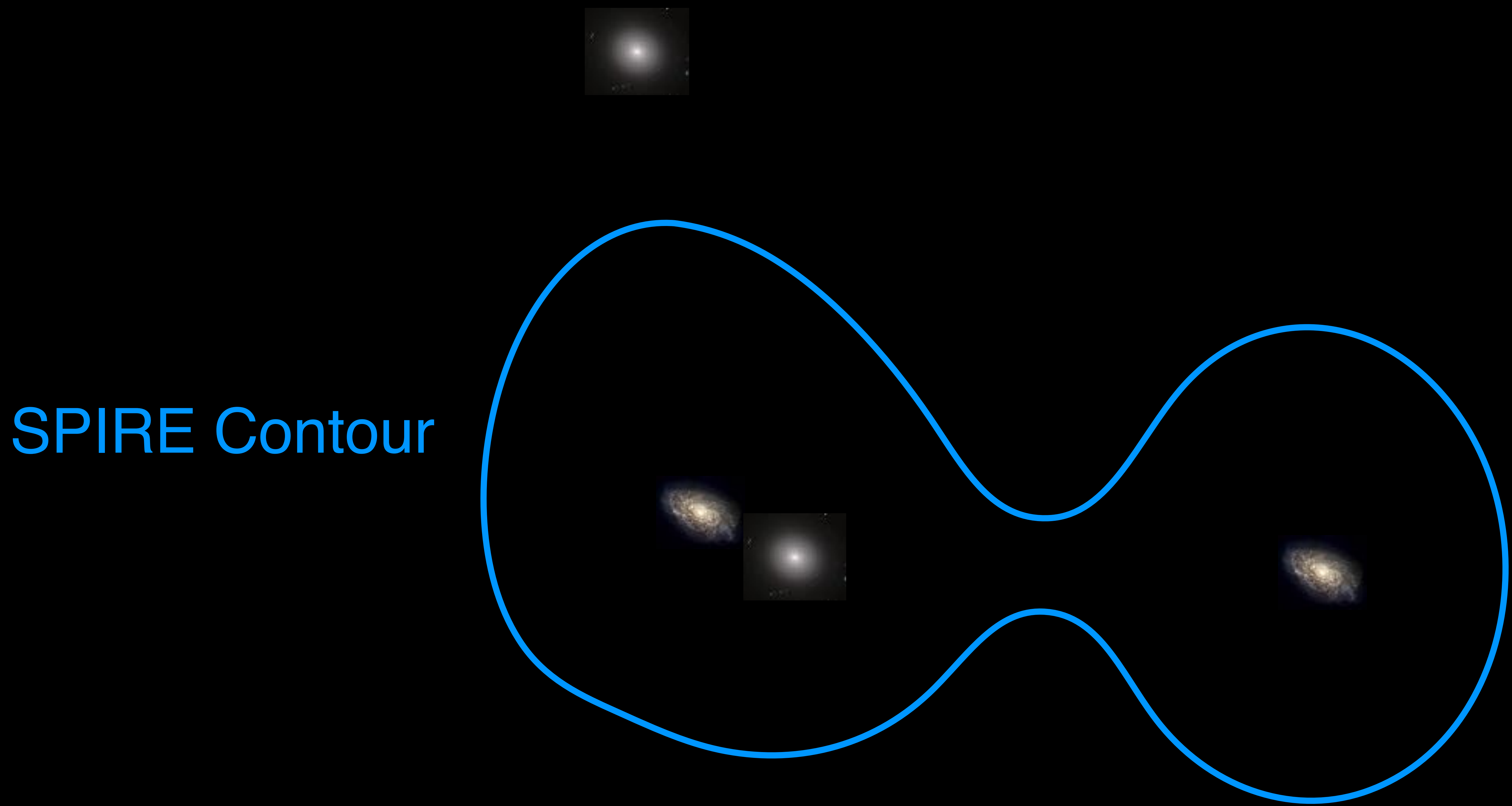
Solution


Use:
• The fact that intensity fluctuations contain signal
• *Ancillary* Data
• Creativity + Statistics
GOODS-S
Half 1


GOODS-S
Half 2



- What if you know these galaxies are here?



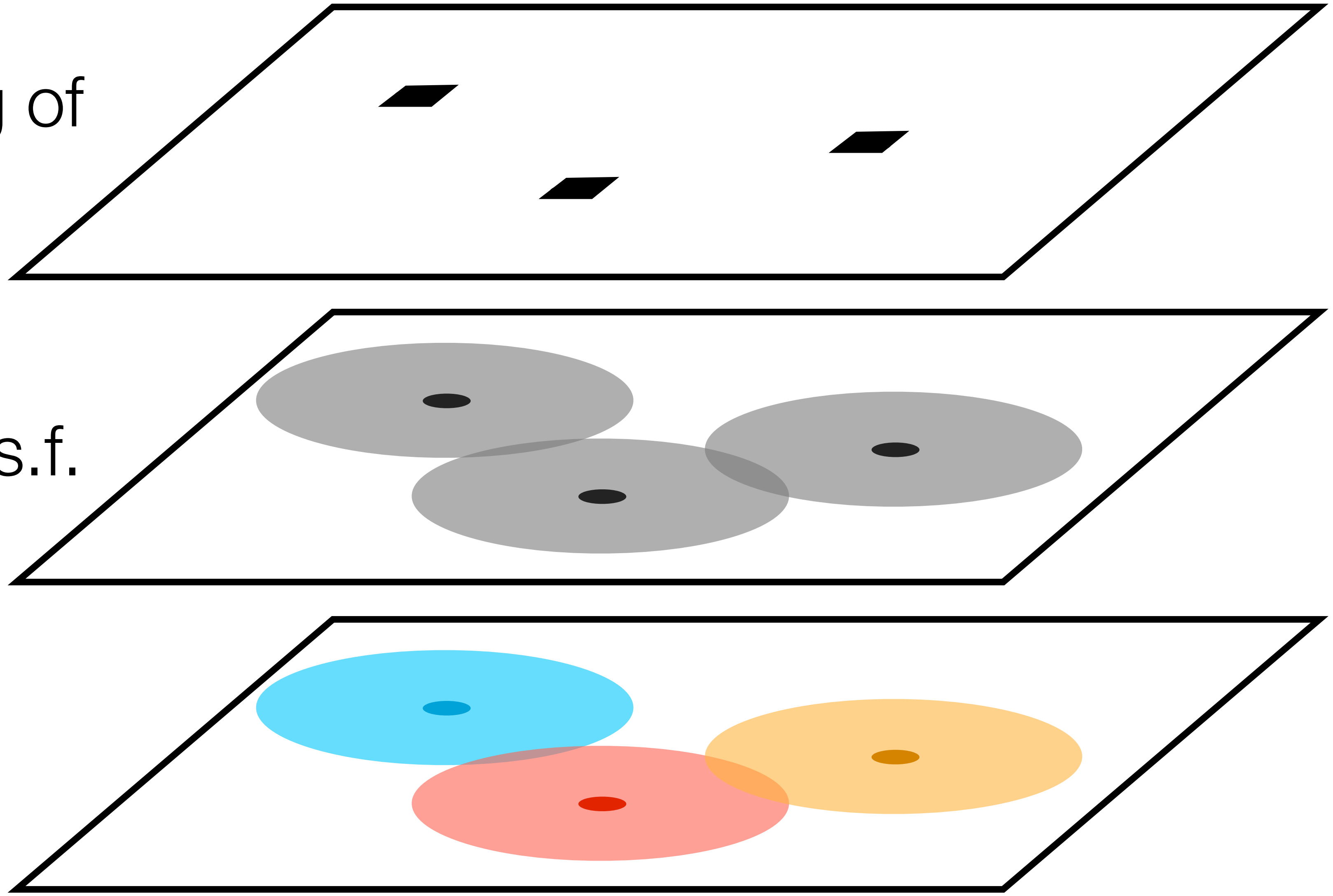
- Key is to identify and ***group*** galaxies with similar *physical* properties, and then rely on ***statistics*** *to fit* ***fluctuations***

SIMSTACK: Simultaneous Stacking Algorithm

make hits map from catalog of similar objects

convolve with instrument p.s.f.

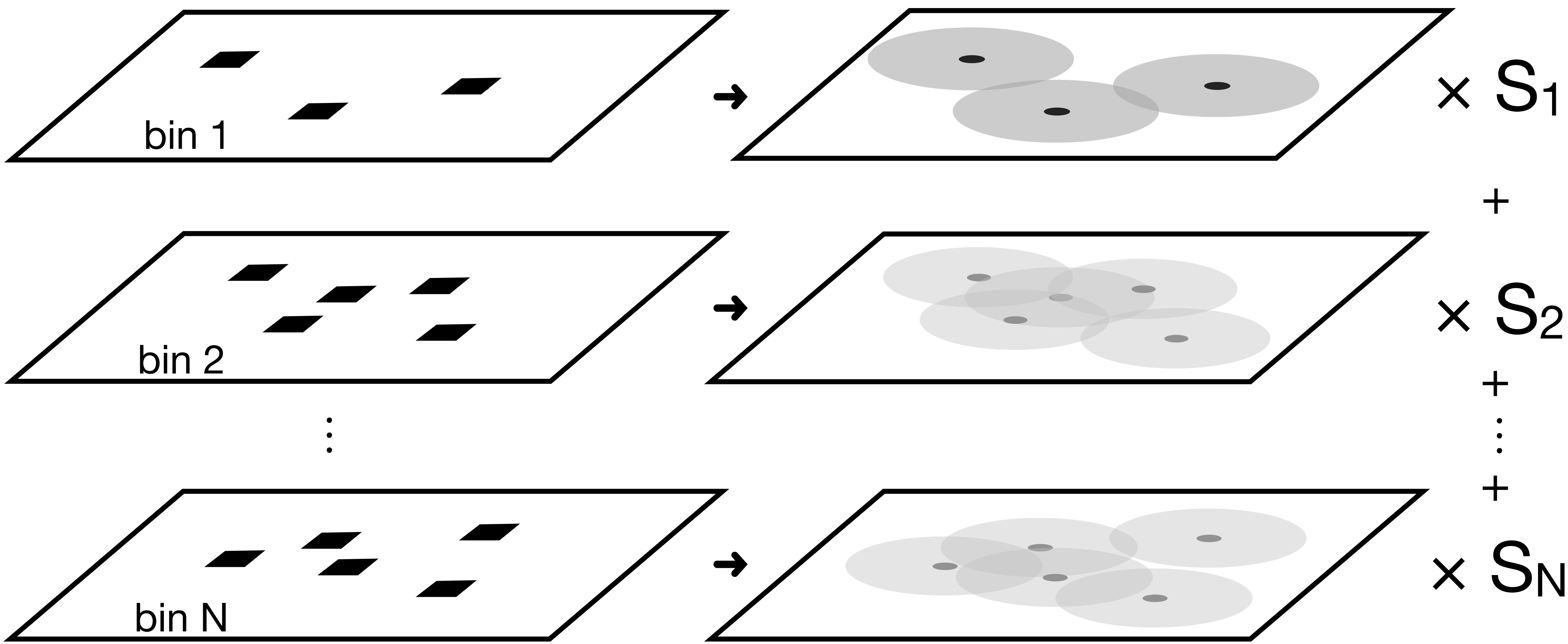
regress to find *mean* flux density, S

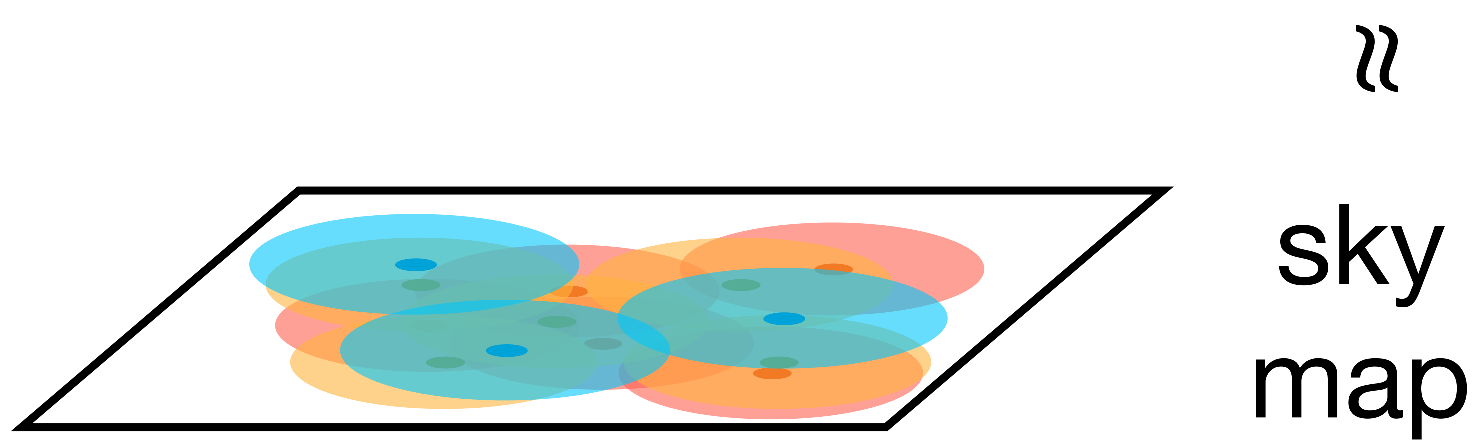


Formalism developed w/ Lorenzo Moncelsi (Caltech);
also see Kurczynski & Gawiser (2010), Roseboom et al. (2010)

SIMSTACK code publicly available (see arXiv:1304.0446):
IDL (old) — https://web.stanford.edu/~viero/downloads.html
Python — https://github.com/marcoviero/simstack

SIMSTACK: Simultaneous Stacking Algorithm





Formalism developed w/ Lorenzo Moncelsi (Caltech)

SIMSTACK code publicly available (see arXiv:1304.0446):
Python — https://github.com/marcoviero/simstack

SIMSTACK is simple to use (Python and IDL)

- Define type of stack, and where everything is, **in config file**.
    - ➡ ./run_simstack_cmd_line.py config_file_name.cfg

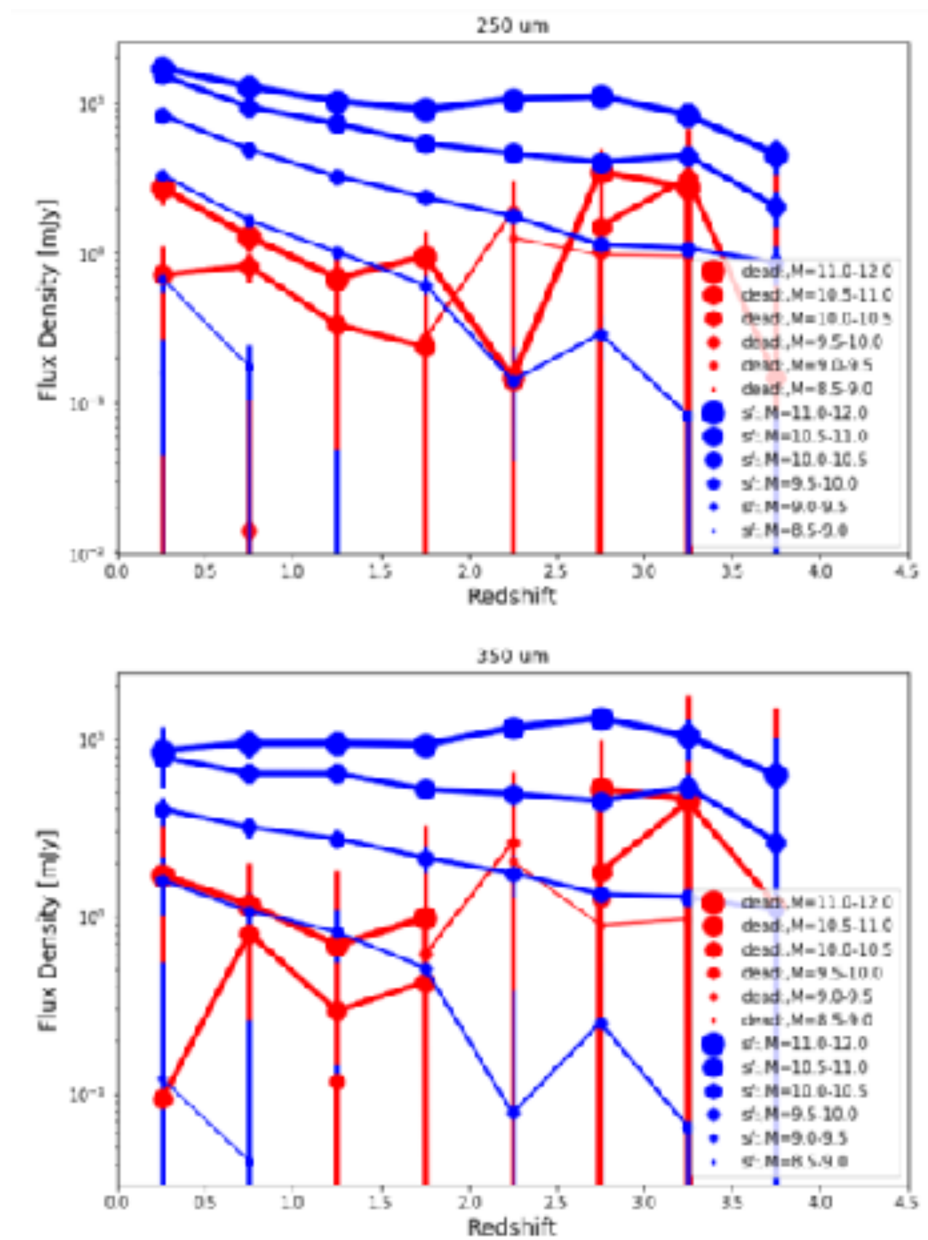
```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Example parameter file for simstack code
;
; Contact: Marco Viero (marco.viero@stanford.edu)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[general]
;populations chooses how the catalog is split into groups with like-properties
;classifying_scheme chooses how the catalog is split into groups with like-properties
;Options are: sf-qt ; general ; uvj ;
classification_scheme = general
bootstrap       = False 0 2 ; True/False, initial number, number of iterations
;Catalog specific names for redshift, stellar mass, RA, and DEC
zkey = PHOTOZ
mkey = MASS_MED
ra_key = ALPHA_J2000
dec_key = DELTA_J2000

[populations]
;Name_of_sub-population = index, [conditions]
;Here conditions are: feature, greater than, less than, equal to
;False when one of those does not apply
sf    = 1 CLASS False False 1
dead = 0 CLASS False False 0

[cosmology] ; Cosmology - Planck15
omega_m = 0.3075
omega_l = 0.6910
omega_k = 0.
h       = 0.6774

[io] ; Input/output
;output_folder will contain the directories:
; - simstack_fluxes
; - bootstrapped_fluxes
;If they don't exist the code will create them!
output_folder            = PICKLESPATH simstack/stacked_flux_densities/
flux_densities_filename = simstack_flux_densities
shortname                = uVista__Laigle_v1.1__sf-qt__z_bins_in_slices_test

[catalogs]
catalog_path = CATSPATH uVista/
catalog_file = COSMOS2015_Laigle+_Simplified_v1.1.csv

[binning]
optimal_binning = False ; Not yet working
bin_in_lookback= False ; Not yet working from command line, and requires NPpredict be installed
all_z_at_once   = False
;If binning in lookback time, redshift_nodes should be in Gyr from present day.
redshift_nodes  = 0.01 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
mass_nodes      = 8.5 9.0 9.5 10.0 10.5 11.0 12.0
```

```
[maps_to_stack]
; True/False represents whether to stack them
mips_24    =   24.0 False
pacs_green =  100.0 False
pacs_red   =  160.0 False
spire_PSW  =  250.0 True
spire_PMW  =  350.0 False
spire_PLW  =  500.0 False
scuba_450  =  450.0 False
scuba_850  =  850.0 False

[map_path]
mips_24 = MAPSPATH /data/cutouts/
pacs_green = MAPSPATH /data/cutouts/
pacs_red = MAPSPATH /data/cutouts/
spire_PSW = MAPSPATH /data/cutouts/
spire_PMW = MAPSPATH /data/cutouts/
spire_PLW = MAPSPATH /data/cutouts/
scuba_450 = MAPSPATH /data/cutouts/
scuba_850 = MAPSPATH /data/cutouts/

[map_file]
; Maps need to be in Jy/beam.  If they are not, use second element in [beams] below to convert them.
mips_24    = mips_24_GO3_sci_10.cutout.fits
pacs_green = pep_COSMOS_green_Map.DR1.sci.cutout.fits
pacs_red   = pep_COSMOS_red_Map.DR1.sci.cutout.fits
spire_PSW  = cosmos-uvista-hipe12_itermap_10_iterations_6.0_arcsec_pixels_PSW.signal.cutout.fits
spire_PMW  = cosmos-uvista-hipe12_itermap_10_iterations_6.0_arcsec_pixels_PMW.signal.cutout.fits
spire_PLW  = cosmos-uvista-hipe12_itermap_10_iterations_6.0_arcsec_pixels_PLW.signal.cutout.fits
scuba_450  = map450_new_header.cutout.fits
scuba_850  = S2CLS_COSMOS_NMF_DR1_new_header.cutout.signal.fits

[noise_file]
; If fits file contains noisemap in second extension, has same name as signal map
mips_24    = mips_24_GO3_unc_10.cutout.fits
pacs_green = pep_COSMOS_green_Map.DR1.err.cutout.fits
pacs_red   = pep_COSMOS_red_Map.DR1.err.cutout.fits
spire_PSW  = cosmos-uvista-hipe12_itermap_10_iterations_6.0_arcsec_pixels_PSW.noise.cutout.fits
spire_PMW  = cosmos-uvista-hipe12_itermap_10_iterations_6.0_arcsec_pixels_PMW.noise.cutout.fits
spire_PLW  = cosmos-uvista-hipe12_itermap_10_iterations_6.0_arcsec_pixels_PLW.noise.cutout.fits
scuba_450  = map450_new_header_rms.cutout.fits
scuba_850  = S2CLS_COSMOS_NMF_DR1_new_header.cutout.noise.fits

[beams]
;1- PSF file path+names, or effective FWHM
;2- Beam area in sr.  Should be 1.0 if maps are in Jy/beam, otherwise actual effective area if Jy/sr
mips_24    = 6.32 1.550e-09
pacs_green = 6.7 2.0271e-07 ; MJy/sr to Jy/beam
pacs_red   = 11.2 4.5398e-09 ; MJy/sr to Jy/beam
spire_PSW  = 17.6 1.0
spire_PMW  = 23.9 1.0
spire_PLW  = 35.2 1.0
scuba_450  = 7.8 1.0
scuba_850  = 12.1 1.0
```

SIMSTACK

- It will save stacked results in a folder you define.
- You can access the results with an iPython Notebook
  - https://github.com/marcoviero/simstack/blob/master/notebooks/plot_simstack_output.ipynb

```
In [1]: import pdb
        import numpy as np
        import pandas as pd
        import os
        import pylab as plt
        from utils import clean_args
        from utils import clean_nans
        from utils import fast_sed
        from utils import fast_sed_fitter
        from utils import fast_Lir
        from utils import stagger_x
        from utils import subset_averages_from_ids
        from utils import main_sequence_s15
        from bincatalogs import Field_catalogs
        from astropy.cosmology import Planck15 as cosmo
        import astropy.units as u
        try:
            from simstack import PickledStacksReader, measure_cib
        except:
            from simstack.simstack import PickledStacksReader, measure_cib

        %matplotlib inline

In [2]: conv_luv_to_sfr = 2.17e-10
        conv_lir_to_sfr = 1.72e-10
        L_sun = 3.839e26 # W
        c = 299792458.0 # m/s
        a_nu_flux_to_mass = 6.7e19
        h = 6.62607004e-34 #m2 kg / s  #4.13e-15 #eV/s
        k = 1.38064852e-23 #m2 kg s-2 K-1 8.617e-5 #eV/K

In [3]: popcolor=['red','blue','green','orange','black','grey','chocolate','darkviolet','pink','magenta','dodgerblue','lavender

In [4]: path_pickles = os.environ['PICKLESPATH']
        path_maps    = os.environ['MAPSPATH']
        path_catalogs= os.environ['CATSPATH']

In [5]: #Location of the stacked parameter file
        shortname = 'uVista__Laigle_v1.1__sf-qt__z_bins_in_slices_test'
        path_config = path_pickles + '/simstack/stacked_flux_densities/simstack_fluxes/'+shortname+'/'
        file_config = 'example.cfg'
        if os.path.exists(path_config+file_config) == True:
            print path_config+file_config

/data/pickles//simstack/stacked_flux_densities/simstack_fluxes/uVista__Laigle_v1.1__sf-qt__z_bins_in_slices_test/exam
ple.cfg
```



SIMSTACK: Measurement Data

Catalogs

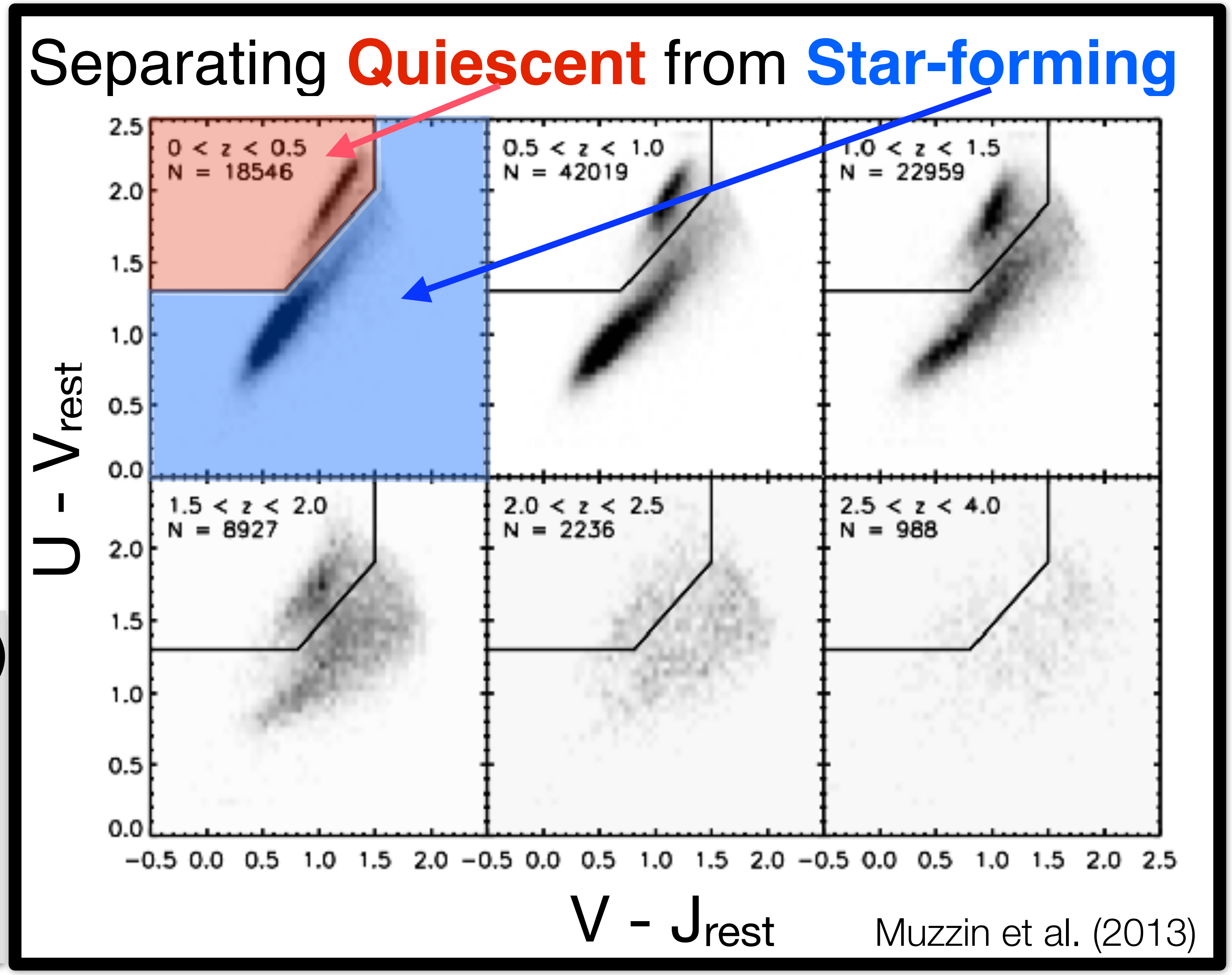
- UKIDSS/UDS [2/3 deg²] / COSMOS [1.6 deg²]

  uBVRizJHK + IRAC ch1234

  K-band cut 23.4 / 24 AB

  80,000 / 120,000 sources

- **Redshifts** - EAZY (Brammer 2008)
- **Masses** - FAST (Kriek 2009)
- **Colors** - UVJ (Williams 2009)

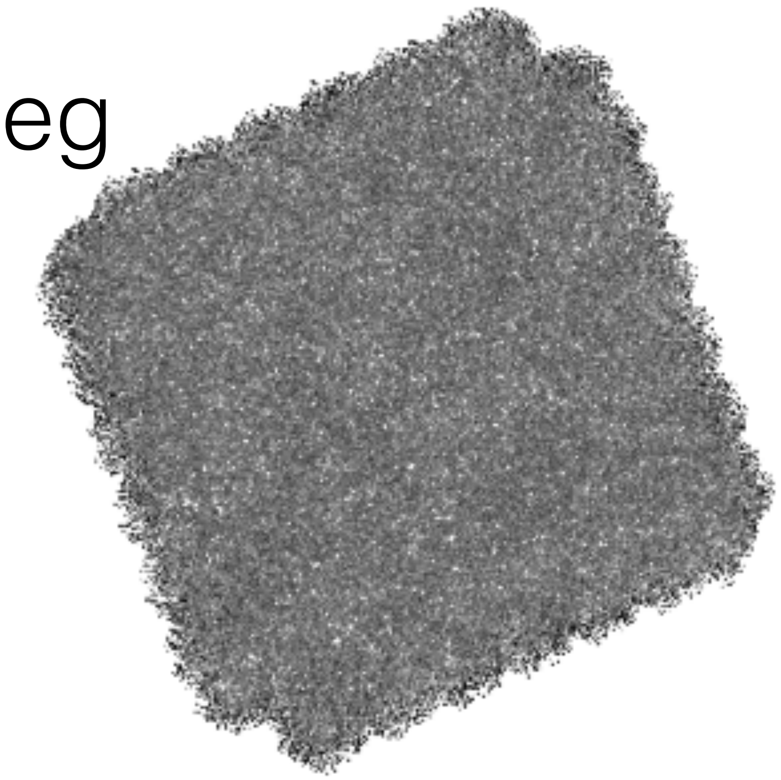


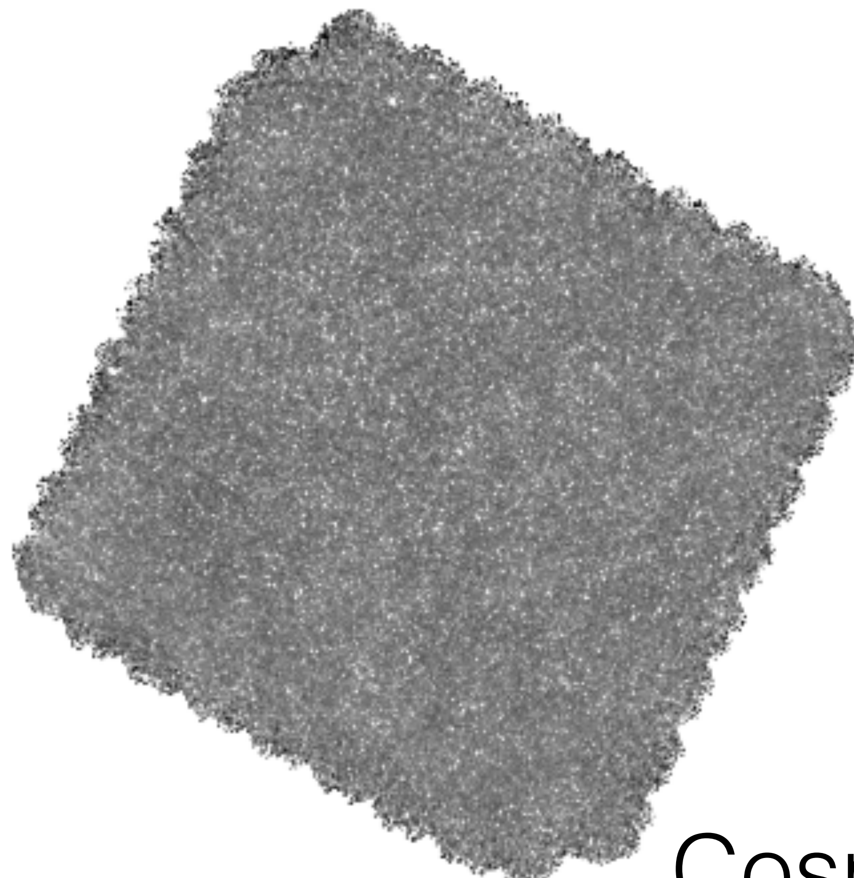
SIMSTACK: Measurement Data

Maps

- *Spitzer*/MIPS
  - 24, 70μm
- *Herschel*/PACS
  - 100, 160μm
- *Herschel*/SPIRE
  - 250, 350, 500μm
- ASTE/AzTEC
  - 1100μm

UDS - 1.4 x 1.4 deg



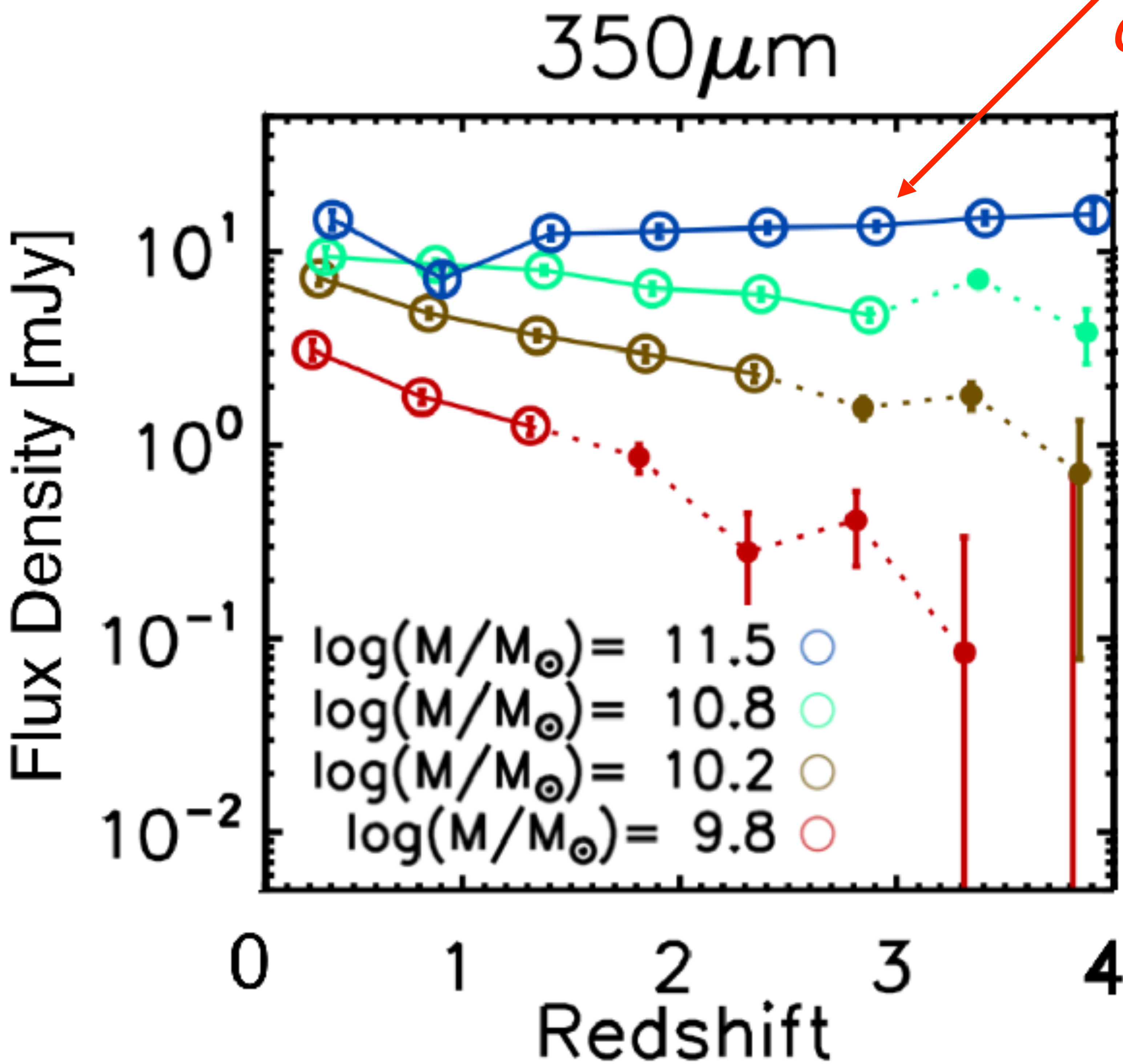


Cosmos - 1.8 x 1.8 deg

SIMSTACK: Flux Densities (M,z)

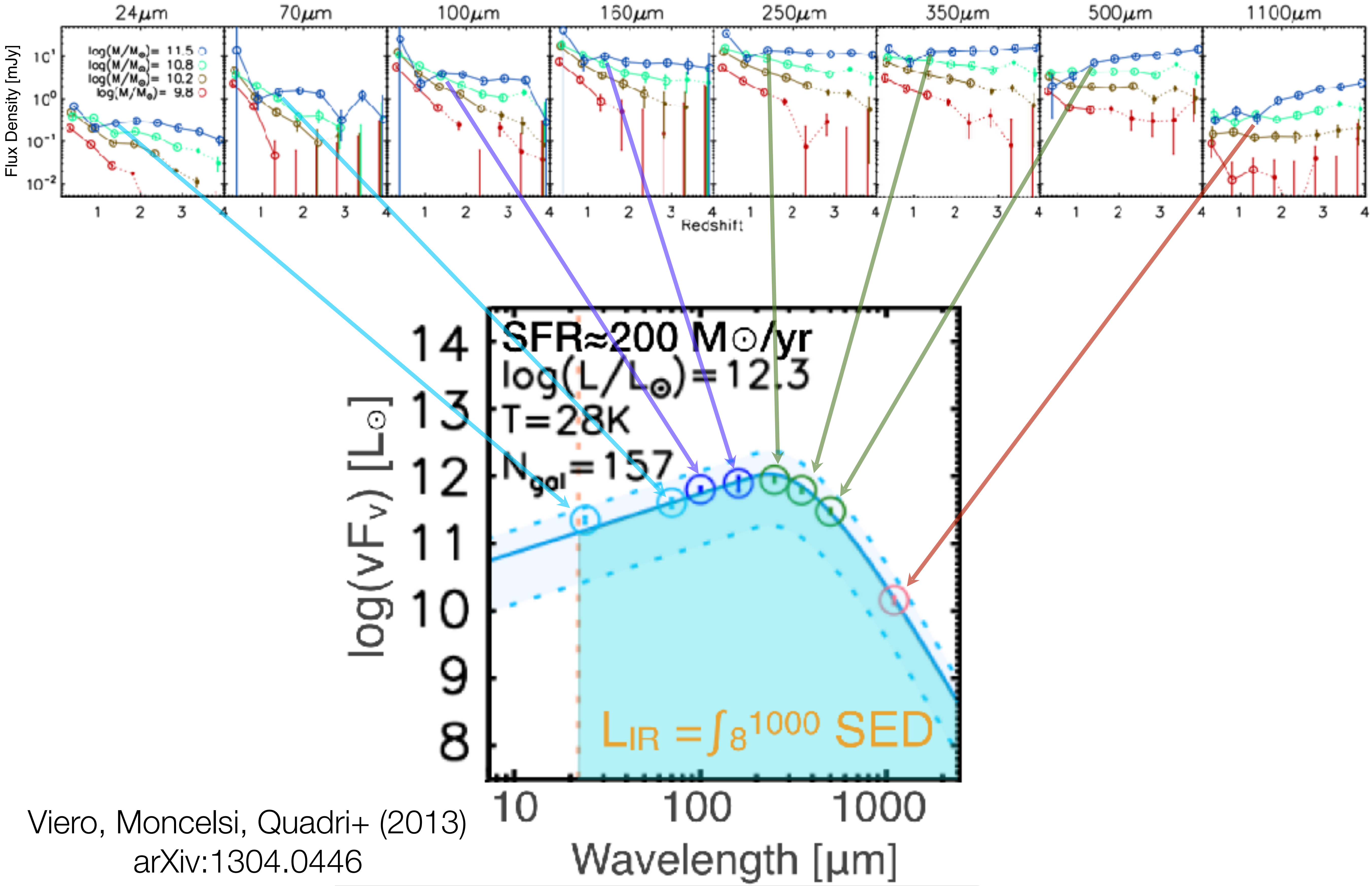
HERMES

Each circle one bin



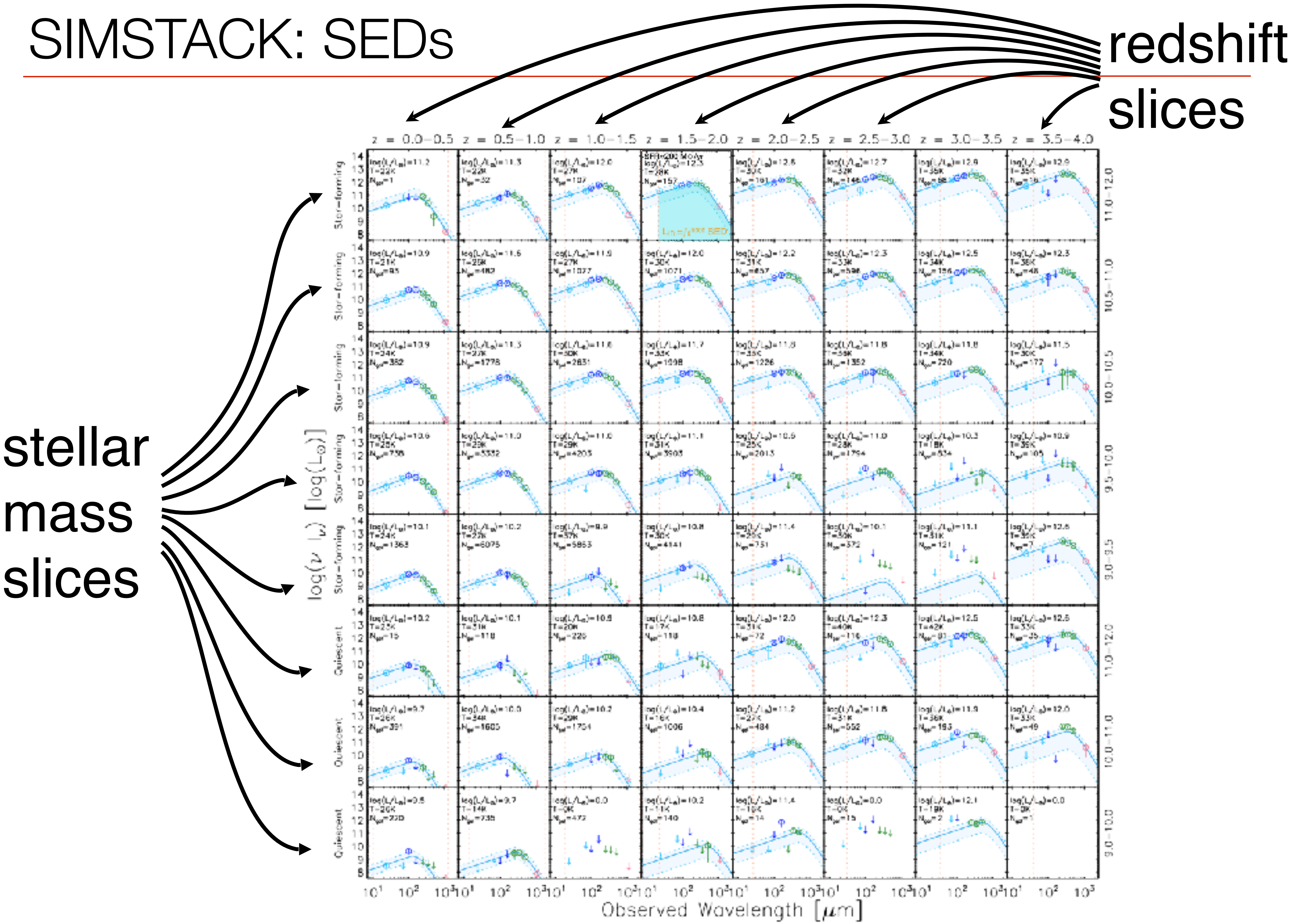
Data here: UDS catalog & Herschel SPIRE maps

SIMSTACK: SEDs



Viero, Moncelsi, Quadri+ (2013)
arXiv:1304.0446

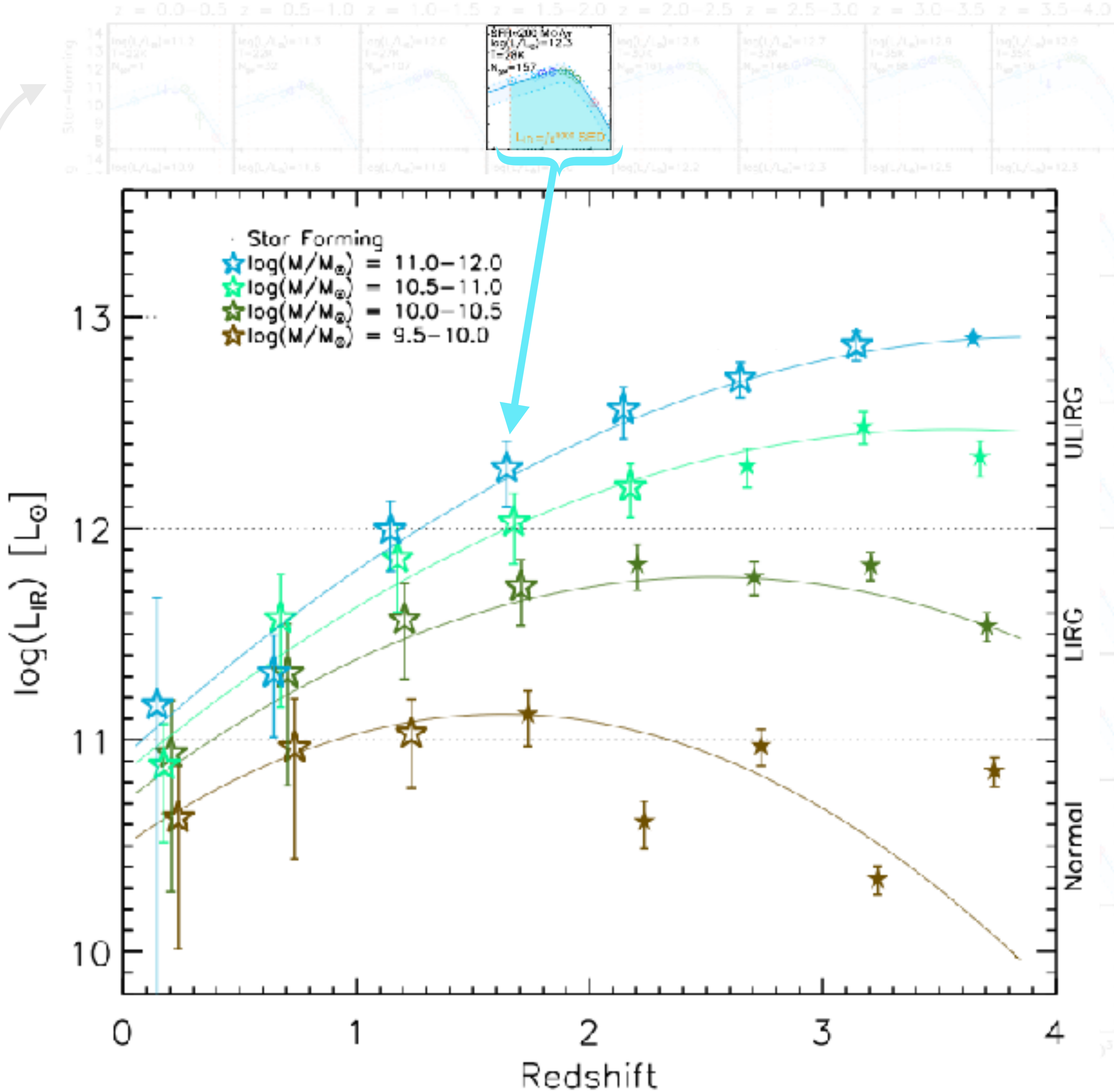
SIMSTACK: SEDs



SIMSTACK: $L_{IR}(M,z)$

redshift slices

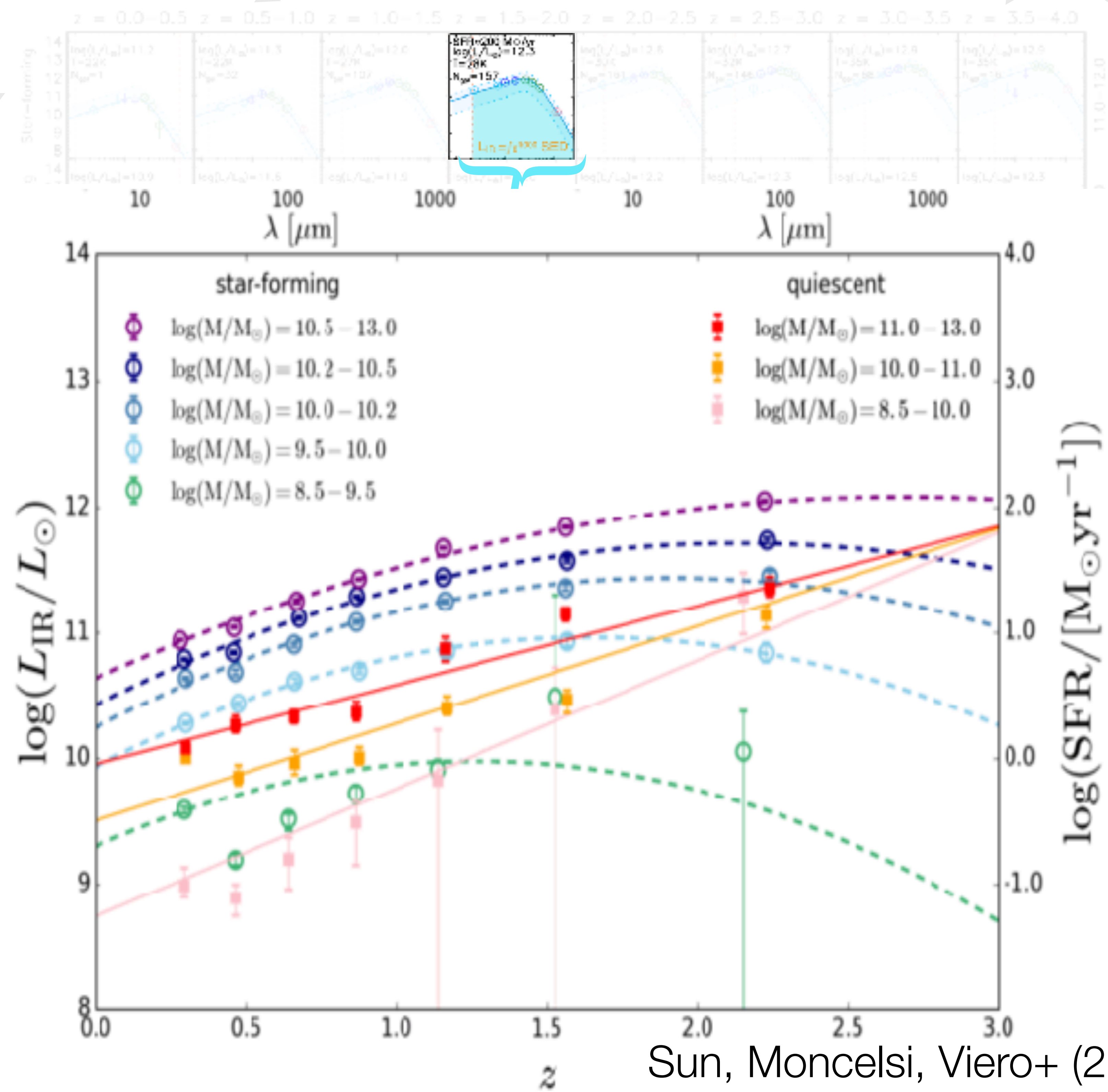
stellar mass slices



SIMSTACK: $L_{IR}(M,z)$

redshift slices

stellar mass slices

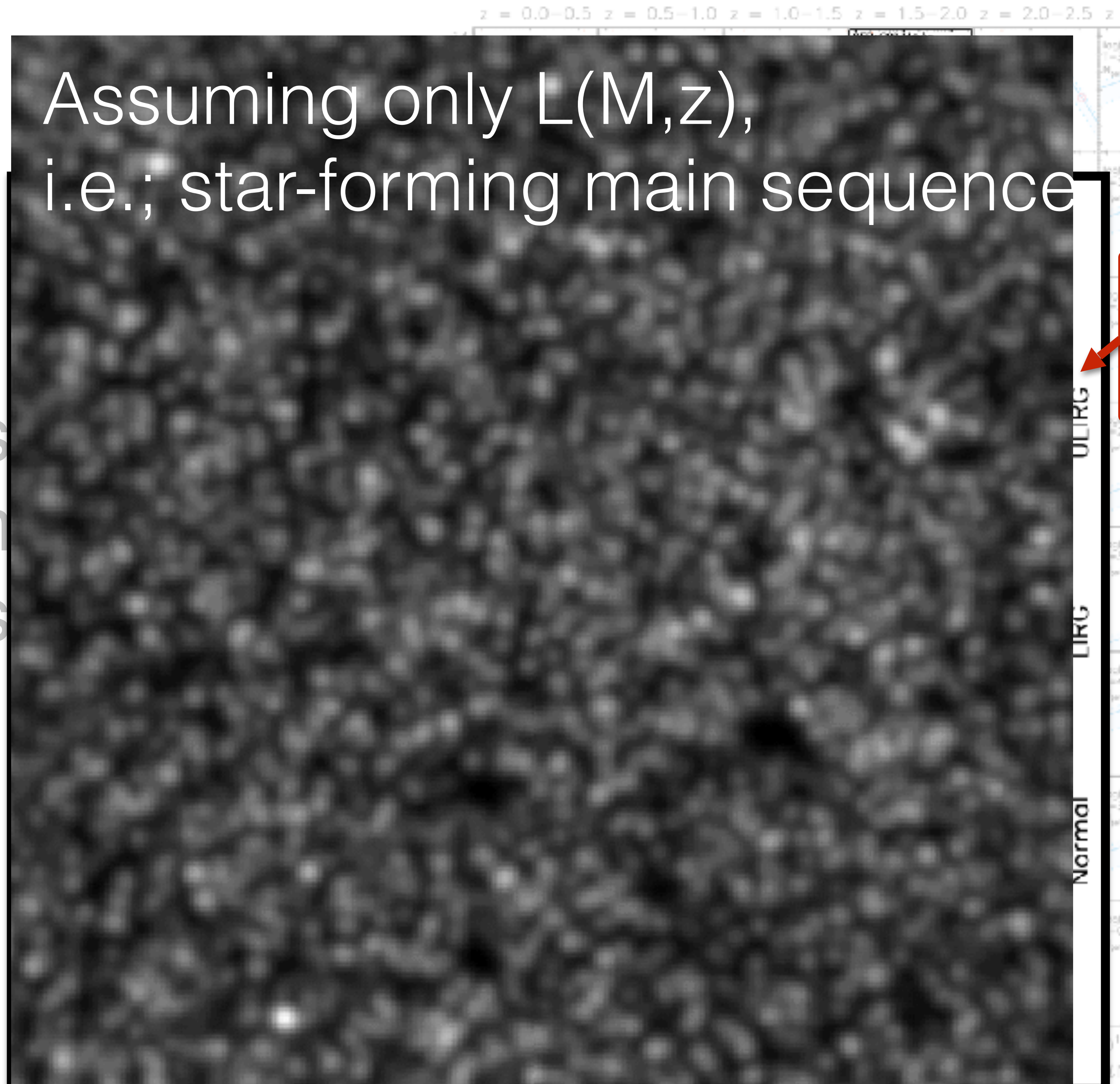


Sun, Moncelsi, Viero+ (2017)
arXiv:1610.10095

SIMSTACK: L_{IR}(M,z,…)

redshift slices

Assuming only L(M,z),
i.e.; star-forming main sequence

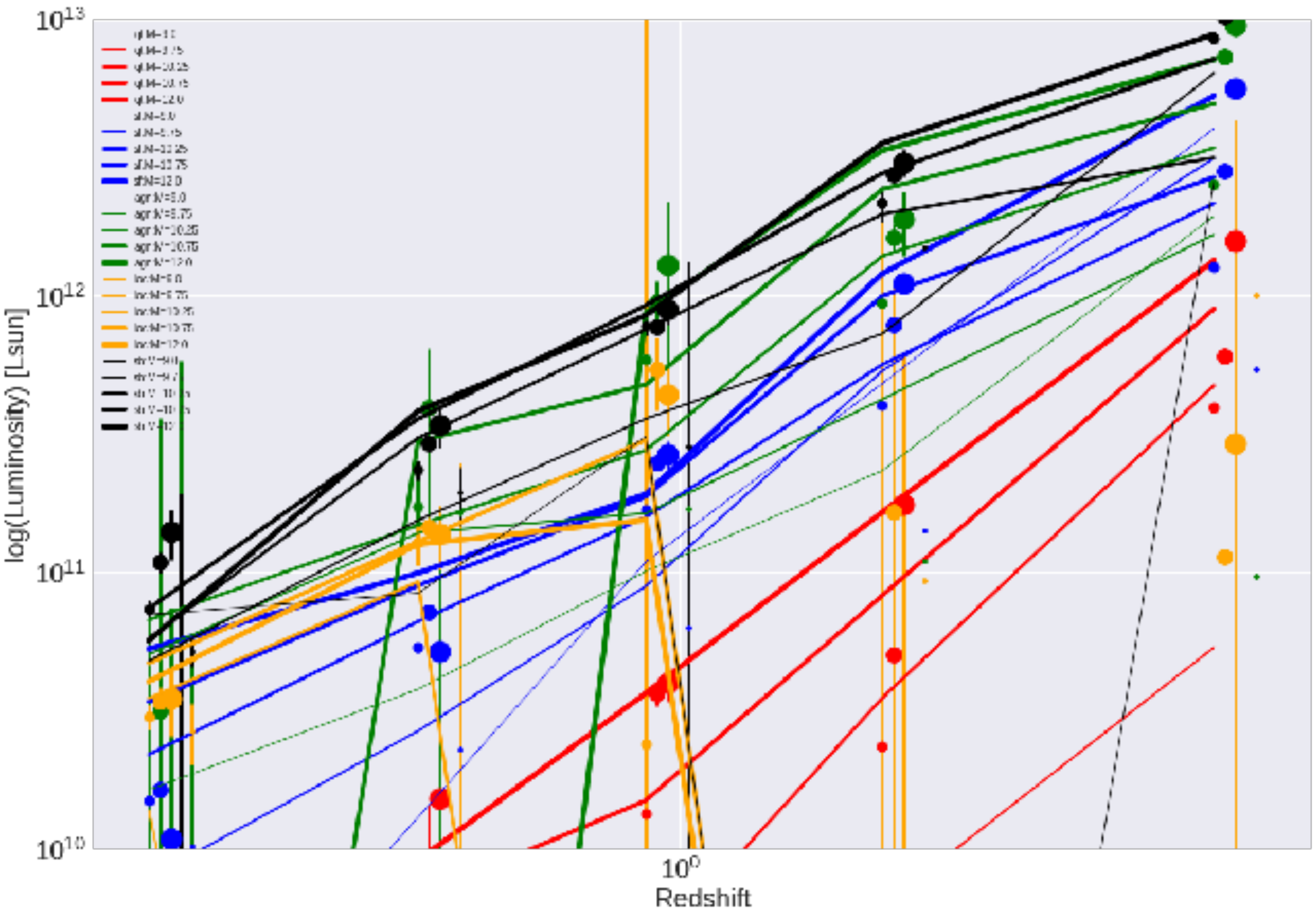


- Deep ancillary data can be fit with SED models, providing:
  - Stellar Mass
  - Redshift
  - Extinction/UV slope
  - AGN fraction
  - Age/Tau…

Each bin therefore has <M>,<z>, <Av>, $<F_{agn}>$, etc., which can be fit with function of form:

- $LIR=P(z)^{\alpha}P(M)^{\beta}P(A_V)^{\gamma}$ …

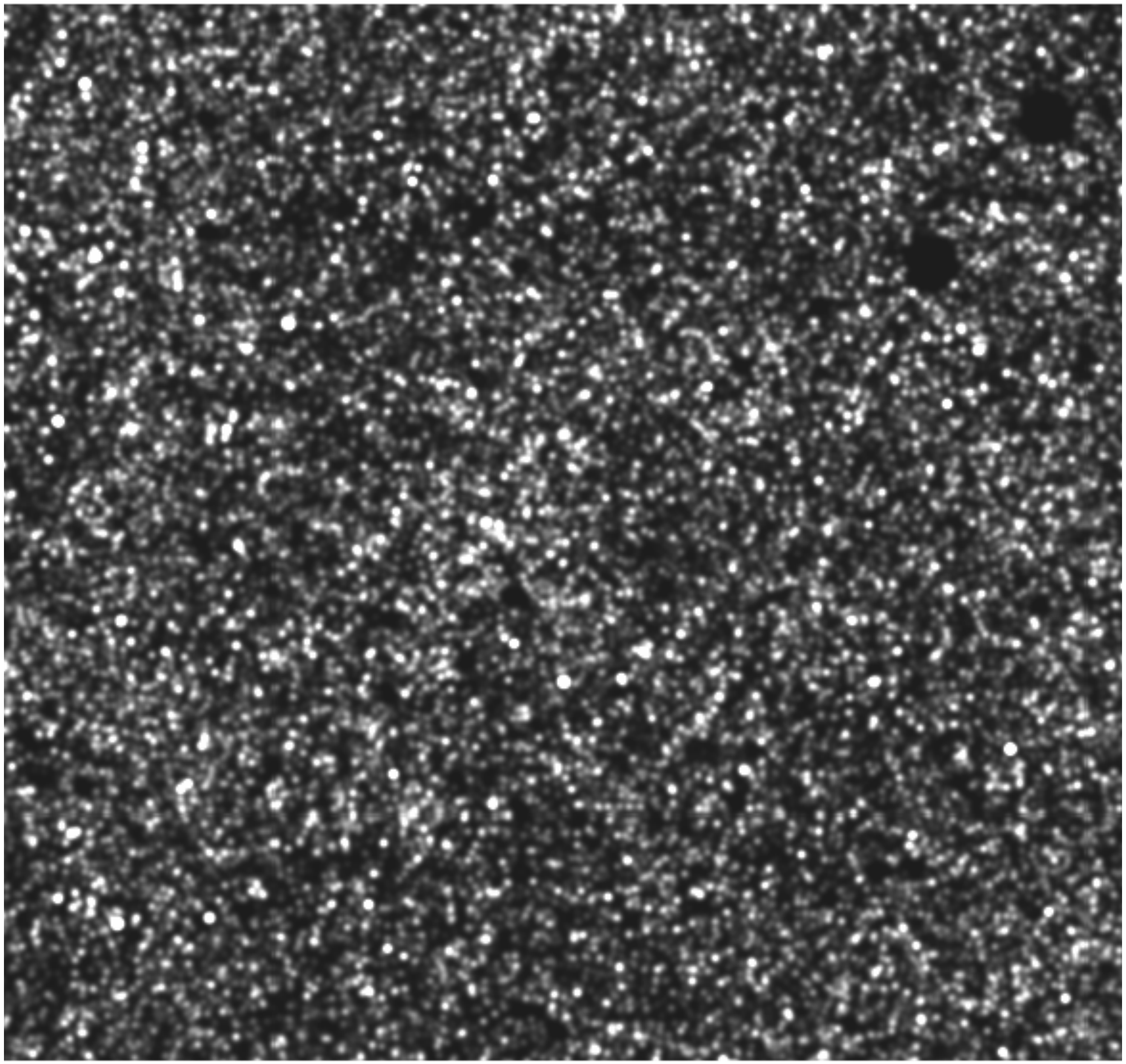
SIMSTACK: L_{IR}(M,z,Av,Fagn)



Fit can be improved by splitting the sample into finer subsamples, isolating e.g,;

- Star-forming/Quiescent
- AGN
- Starbursts

SIMSTACK: L_{IR}(M,z,Av,Fagn)



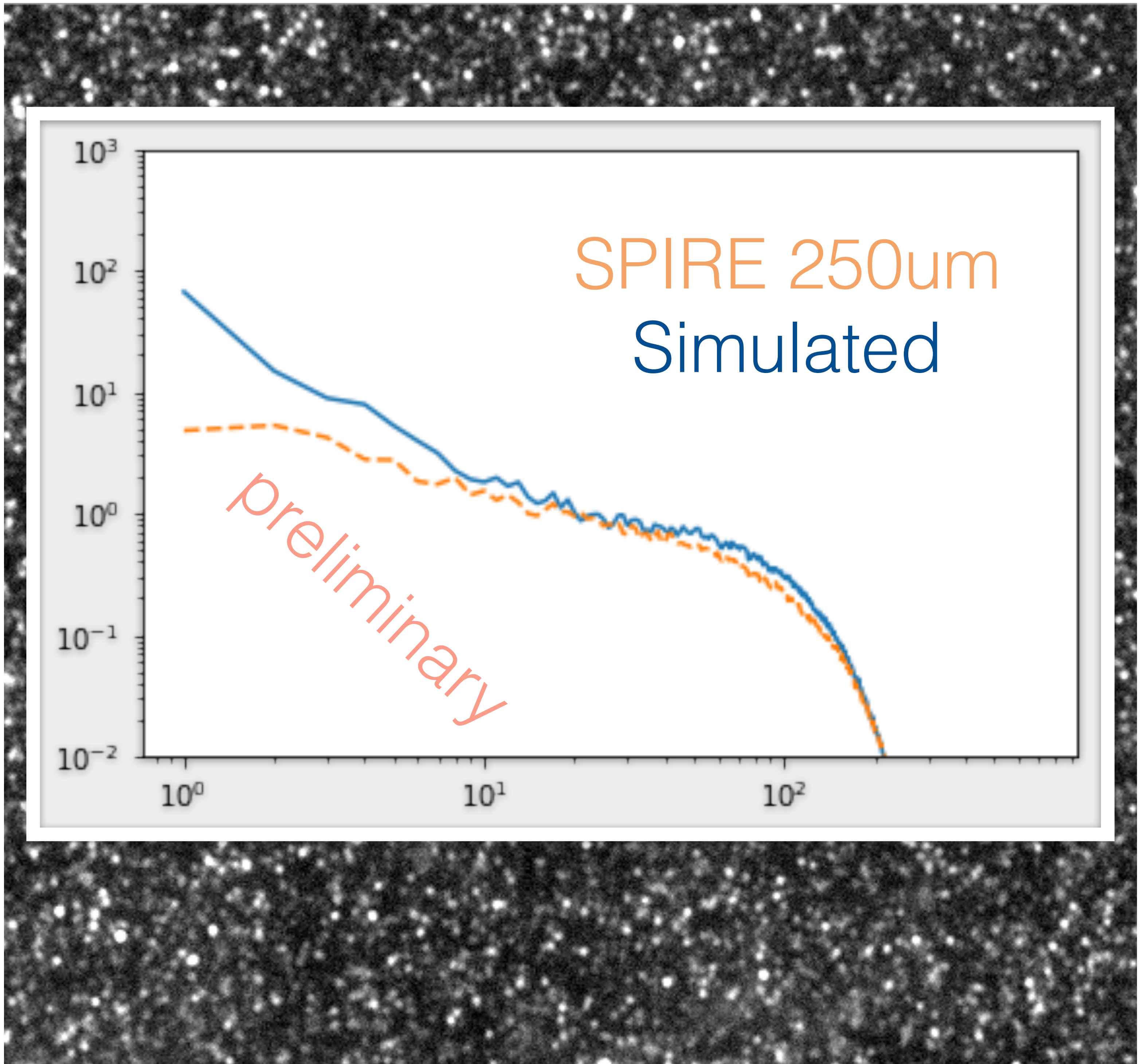
Fit can be improved by splitting the sample into finer subsamples, isolating e.g,;

- Star-forming/Quiescent
- AGN
- Starbursts

We find features most influential are, for 4 subsamples:

- $\log(LIR) = C + \alpha(z)\times\log(1+z) + \beta(z)\times\log(M) + \gamma(z)\times\log(Av) + \delta(z)\times\log(F_{agn})$

SIMSTACK: L_{IR}(M,z,Av,Fagn)



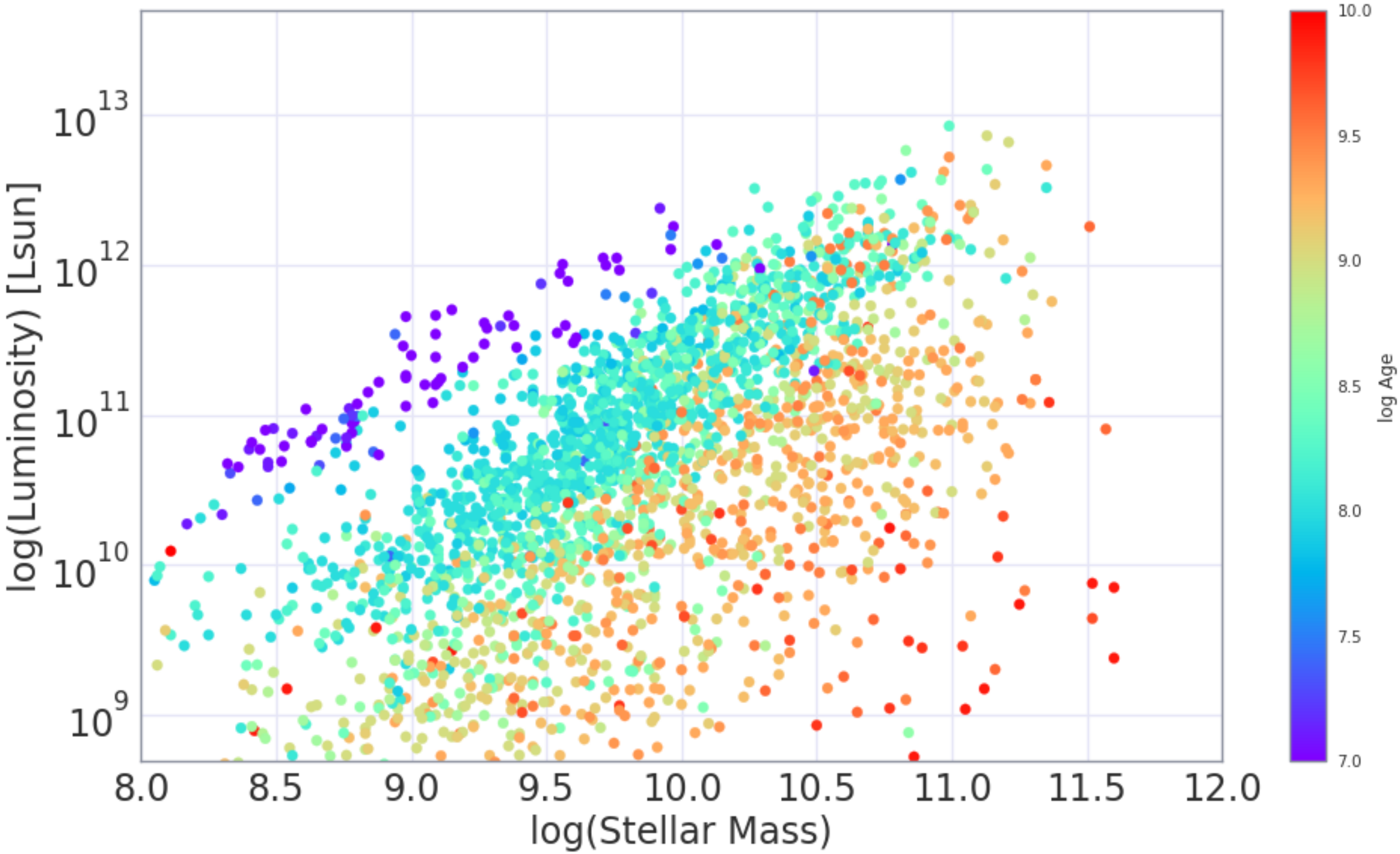
Fit can be improved by splitting the sample into finer subsamples, isolating e.g,;

- Star-forming/Quiescent
- AGN
- Starbursts

We find features most influential are, for 4 subsamples:

- $\log(LIR) = C + \alpha(z)\times\log(1+z) + \beta(z)\times\log(M) + \gamma(z)\times\log(Av) + \delta(z)\times\log(F_{agn})$

SIMSTACK: L_{IR}(M,z,Av,Fagn)



Applications

- Signal
  - ➡ Connect to Halo properties (including assembly bias) to:
    - estimate CO levels,
    - construct covariances,
    - test different estimators (i.e., beyond power spectrum),
    - Details being discussed during this meeting!
  - ➡ Extend to other lines that correlate with thermal dust SED
    - CII, OII, OIII, NII
    - r.f. 850um as tracer of ISM Mass.

- Foregrounds
  - ➡ Predict CO contamination in CII data cubes (e.g, Sun and the TIME collaboration, 2017)

Summary

- CIB continuum intensities are key to empirically connecting optical features of typical galaxies to their FIR/submm components
- Applications for this model include:
  - ➡Forecasting CO power for:
    - Survey design
    - Covariance construction
    - Testing Estimators
    - Measurement Interpretation

- SIMSTACK is easy to use, and available at:
  - ➡https://github.com/marcoviero/simstack