2019

Finding Obscure Black Hole Growth via Spectral Energy Distribution Modeling

Randall K. Chan
Colby College

Finding Obscure Black Hole Growth via Spectral Energy Distribution Modeling

Randall K. Chan

Supervisor: Professor Dale Kocevski
Department of Physics and Astronomy
Colby College
Honors Thesis 2019

Table of Contents

ABSTRACT

An active galactic nucleus (AGN) occurs when the supermassive black hole at the center of the galaxy starts to grow. We still currently do not know what triggers AGN. Theories suggest that galaxy mergers could trigger AGNs, but past research has not been able to find a correlation between x-ray detected AGNs and disturbed galaxies. The present research looks specifically at AGN not detected in the x-ray, or obscured AGN. Using a newly updated IDL code, FAST, we were able to identify potential obscured AGN through spectral energy distribution (SED) modeling. We found a total of 526 obscured AGN in all CANDELS fields, all with the following properties: mass greater than 10^{10} M_{solar}, redshifts between 0.5 and 1.5, and a magnitude brighter than 24.5. Using visual classification, galaxies were labeled as either disturbed or undisturbed. Then using a simple binomial distribution, we found a slightly significant difference (σ = 2.18) between obscured AGN and a control group. In conclusion, we found that obscured AGN were slightly more disturbed than their non-AGN counterparts.

ACKNOWLEDGEMENTS

I would like to first acknowledge Dale Kocevski for his assistance and guidance throughout this project. I would also like to thank the Colby College Physics and Astronomy Department for giving me this opportunity. Finally I would like to acknowledge my parents and my friends for their continued supported during this process.

1. INTRODUCTION

Ever since black holes were discovered in 1971, there has been extensive research looking at their nature and behavior. Black holes are objects of infinite density due to having zero radius. They are often referred to as singularities. Their gravitational pull is so great that not even light can escape. While the collapse of high mass stars creates stellar mass black holes, it is believed that at the center of most galaxies, there lies a supermassive black hole (SMBH). The masses of these SMBHs tend to be millions or even billions times greater than the mass of our sun. There has been substantial research looking at the interaction between a SMBH and its host galaxy. Some of the most important questions are: What causes a SMBH to start accreting material and grow in mass? In what situation does a SMBH start accreting?

1.1 Background into Active Galactic Nuclei

A phenomenon connected to SMBHs that occurs at the center of select galaxies is the presence of an active galactic nucleus (AGN). This is characterized by gas accretion into a supermassive black hole, causing outflows and feedback (Hickox and Alexander 2018). Current models describe an AGN as an accretion disk surrounding the black hole and a gaseous torus around the disk. The gas and matter from this disk accretes onto the black hole and causes the black hole to grow in mass (Alexander and Hickox 2011). To expel the extra energy, the AGN will produce jets typically on the rotation axis of the accretion disk.

There are different classifications of AGNs, but these different types of AGNs are merely due to our viewing angle of the central SMBH (see figure 1) (Zackrisson 2005). Each view angle produces a very different light spectra. A Blazar is a galaxy whose disk is perpendicular to our line of sight, and they are the most luminous. If it is not a Blazar, it is classified as a Seyfert AGNs, whose type (I or II) is determined by the steepness of the viewing angle. Each classification type emits different spectral energy distributions (SED), meaning that at each wavelength, the AGN emit different amounts of light.

AGNs are studied because it is observed that some characteristics of SMBH are tied to the properties of its host galaxy and visa versa. A fundamental correlation is the one between the mass of the black hole and the mass of a galaxy's central bulge (Magorrian et al. 1998; Gebhardt et al. 2000). It has been proposed that this could be due to galaxy mergers, star-formation winds, or possibly AGN driven outflows. There also is a correlation found between AGN activity and star formation in its host galaxy (Alexander and Hickox 2011). It is thought to be due to the fact that they both need cold gas. Thus when a galaxy has a surplus of cold gas, it can support star formation and allow for a stable AGN.

1.2 Detection Methods

As matter spirals into a SMBH, it will heat up and emit immense amounts of

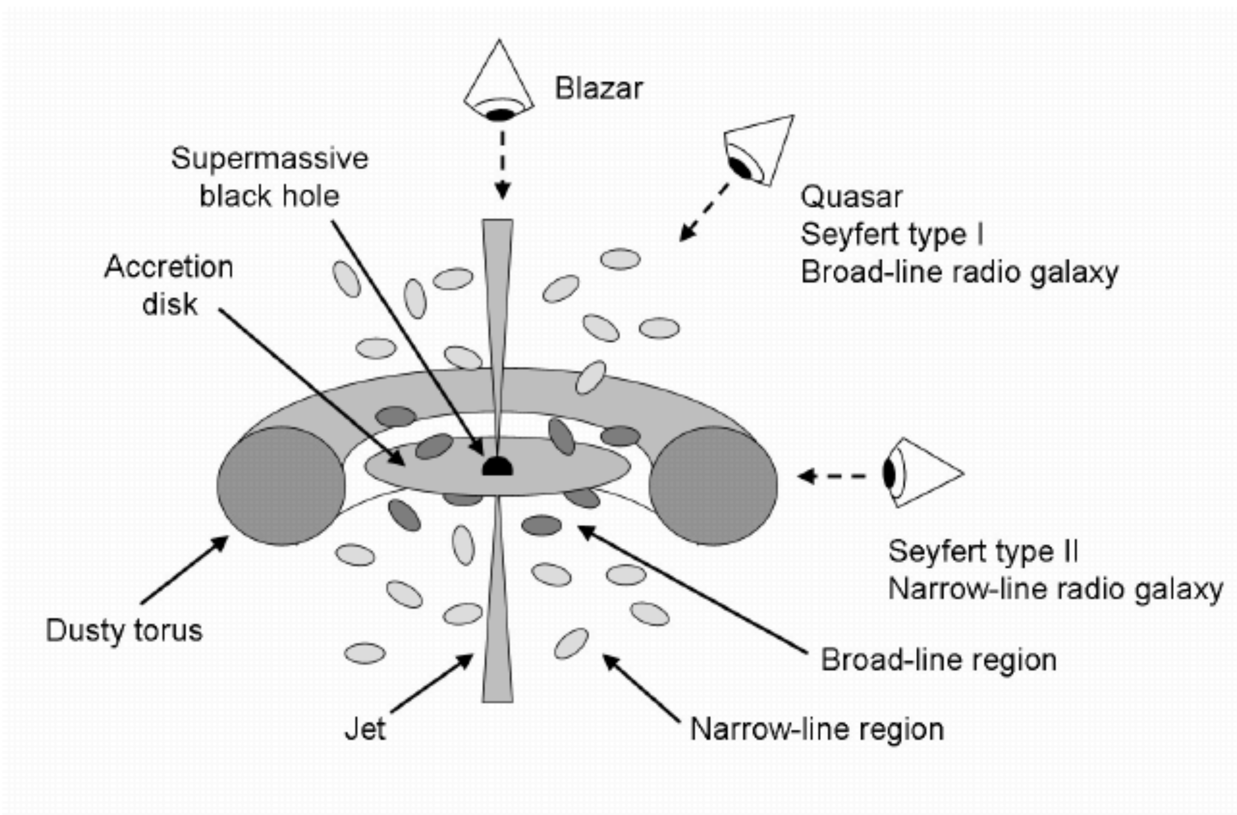


Figure 1. This is a visual diagram of the commonly accepted model of an AGN. The SMBH lies at the center followed by the accretion disk. Further out is the dusty torus. The classification of the type of AGN is also given by the viewing angle. Looking straight into the center of the AGN along the rotation axis classifies as a Blazar. Seyferts are classified by how steep the viewing angle is. Figure is taken from Zackrisson (2005).

light optical, x-ray, and infrared wavelengths. However, each wavelength has its own strengths and complications. Not all AGN will look the same, so a multi-wavelength approach must be taken.

1.2.1 Optical Wavelengths

Optical light typically comes from the heating up of the accretion disk (Hickox and Alexander 2018; Padovani et al. 2017). The wide range of temperature in the accretion disk leads to a wide range of wavelength emission. While this might be the easiest to see, it is difficult to differentiate between light from the AGN and from starlight since both emit in the optical wavelength (Padovani et. al. 2017).

1.2.2 X-Ray Wavelengths

X-rays are the most reliable way to detect AGN. There is little in a galaxy that emits x-rays, thus a high x-ray flux coming from a galaxy is a strong indication that there is an AGN present (Padovani et al. 2017). X-rays are produced by a direct line of sight into a non-obscured central engine of an AGN. This will typically produce photons in the x-ray, UV, and optical wavelengths.

While this may be the most reliable way of detecting AGNs, not all AGNs are visible in the x-ray wavelengths. There are many factors that lead to the obscuration of AGNs in the x-ray, including gas form the surrounding torus blocking one's line of sight

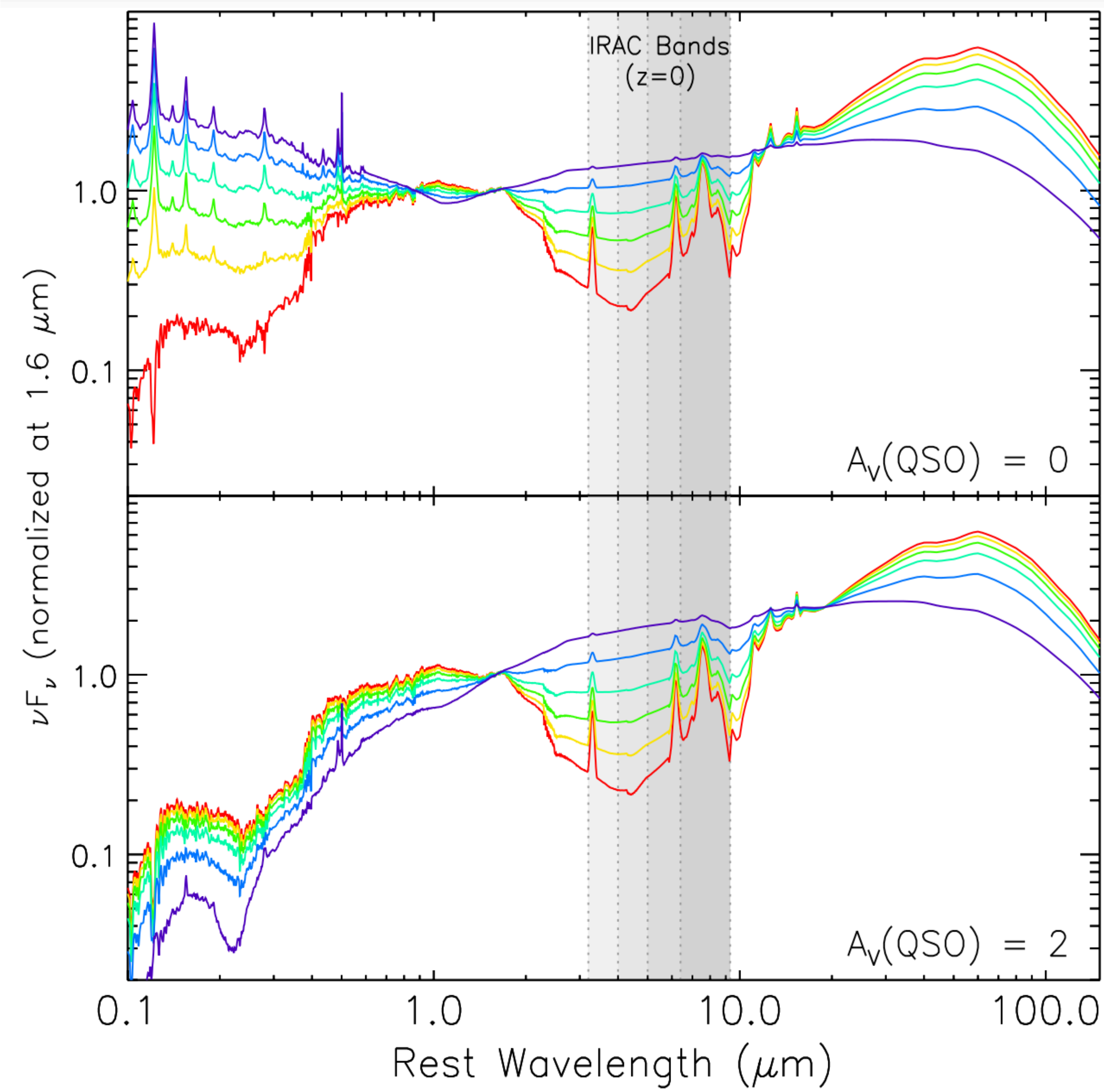


Figure 2. These SEDs are models of two galaxies. The top galaxy has an AGN and the lower galaxy does not. The red line is representative of light coming from the galaxy and the blue line is representative of potential AGN contribution according to the model fit in Donley et al. 2012. A huge discrepancy can be seen in the UV, as there is much more UV contribution in the galaxy with an AGN. Figure adapted from Donley et al. (2012).

to the central engine. Therefore, only using x-ray-detected AGN for research leaves out a large sample of AGN.

1.2.3 Infrared

Infrared emission comes from the dusty torus emission disk light that is absorbed by dust particles. These particles reemit the light in the mid-infrared (Padovani et al. 2017). Infrared is especially important in discerning between the types of AGN. In Seyfert AGNs, the dust obscures the line of sight to the inner regions of the AGN. Thus, only the reemitted light from the dust is detected. For infrared, there is only a small region (mid-infrared) that can be used to identify AGNs. Star formation activity also gives off in the infrared, specifically the far infrared. This indicates using the typical rest frame for infrared (near infrared) consists of both star formation and AGN.

1.3 SED Modeling

A spectral energy distribution (SED) is the flux density (flux per unit wavelength) emitted by an object as a function of wavelength. These prove useful due to the previously stated problems of classifying AGN by only wavelength. AGN will have a different SED than galaxies without an AGN (Donley et al. 2012). Looking at a galaxy's SED, there are certain characteristics that mark an AGN. For example, a rising blue continuum is characteristic of a Blazar. All star-forming galaxies will have emission lines, but AGN produce some emission lines more than others. Namely, AGN excite lines with high ionization potentials, which require highly energetic photons to get produced. These lines require really energetic photons to get produced. In the optical, the two most prominent lines are double ionized oxygen (OIII) and singly ionized nitrogen (NII). Young stellar populations, via recent star formation, preferentially excite low-ionization lines such as Hydrogen and singly ionized oxygen (OII; Kocevski 2018).

Unfortunately, these SEDs are not always easily distinguishable. Previous research has tried to figure out where AGN differ from normal galaxies, especially when the AGN is obscured (Donley et al. 2012). AGN contribution to galaxies' SED can typically be seen in the UV and in the MIR spectra due to emission from the hot accretion disk and radiation reprocessed by dust in the surrounding torus, respectively (see figure 2).

1.4 AGN-Merger Connection

Previous research has looked at whether galaxy mergers may play a role in the turning on of an AGN. Further research has found a correlation between the size of an SMBH and the size of a galaxy's stellar bulge, which may be established by galaxy mergers (Hopkins et al. 2008). The hypothesis that galaxy mergers may cause the creation of an accretion disk stems from the violent nature of galaxy mergers. When galaxies merge, the gases from both galaxies interact. It is theorized that the gas is funneled to the new center of the galaxies. This creates the dust torus that is typically thought to reside around an SMBH. Thus, the SMBH can start to feed on this torus and an accretion disk is created.

Past research tried to look at a potential correlation between mergers and AGN, but there have been mixed results. Some papers find a correlation between AGN and mergers (Ellison et al. 2011; Koss et al. 2018), and some find no relation. (Cisternas et al. 2011; Kocevski et al. 2012). Most of the research has looked at only x-rays detected AGN because they are the easiest to identify. The problem with this method is that potentially about 60% of AGN are obscured and undetectable by x-ray (Rovilos et al. 2014).

1.5 Obscured AGN

Obscured AGN are believed to be AGN with a very narrowing viewing angle. Thus, they are unable to be detected by x-rays, UV and optical. Typically, thick layers of dust and gas form the torus surround these AGN (Hickox and Alexander 2018). Dust blocks UV rays from escaping while gas blocks the x-rays. Again, the optical light and some IR light can be hard to distinguish from the photons from the host galaxy. These AGNs are harder to detect for two reasons: diminished emission of the AGN and host galaxy dilution. These types of AGN are unfortunately harder to detect, but are still a significant portion of all AGNs.

1.6 Present Study Hypothesis

The present study addresses the problem of obscured AGN. Using a new program, FAST, which claims to be able to differentiate between galaxy and AGN light contribution, this study disregards x-ray identified AGN and focuses on obscured AGN. Then visual classification will be used to test for a correlation between disturbed galaxies and AGN contribution to galaxy SED.

2. DATA DESCRIPTION

The data used in this experiment was taken from the Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (CANDELS) (Grogin et al. 2011). This survey is comprised of two parts (Deep Survey and Wide Survey) together spanning 800 $arcsec^2$. The Deep Survey consisted of two areas: GOODS-N (GDN) and GOODS-S (GDS), whereas the Wide Survey consisted of three areas: Extended Groth Strip (EGS); COSMOS (COS); and Ultra Deep Survey (UDS). Together these five areas contain imaging of more than 250,000 galaxies with redshifts from 8 to 0.5. These fields were surveyed mainly using the Hubble Space Telescope with some filters using Spitzer Space

Telescope, the Very Large Telescope, and the Victor Blanco 4m telescope. The first two mentioned telescopes are in orbit and used to collect mainly infrared rays, and the following two are ground-based and used to collect mainly x-ray and UV rays. An overview of the Hubble filters used for each field is displayed in Table 1 (Grogin et al. 2011). Each field has a combination of filters to create the widest array of light possible. Typically, filters will overlap so that the peaks of each filter combine for a full spectrum of light. An example of the filters and their wavelength range for GDS can be seen in Figure 3.

Table 1

Field	Coordinates	Tier	WFC3/IR Tiling	HST Orbits/Tile	IR Filters[a]	UV/Optical Filters[b]
GDN	189.228621, + 62.238572	Deep	~3 × 5	˜13	Y J H	UV, U I (W V z)
GDN		Wide	2 @ ~2 × 4	~3	Y J H	I z (W)
GDS	53.122751, −27.805089	Deep	~3 × 5	˜13	Y J H	I (W V z)
GDS		Wide	~2 × 4	~3	Y J H	I z (W)
COS	150.116321, + 2.2009731	Wide	4 × 11	~2	J H	V I (W)
EGS	214.825000, + 52.825000	Wide	3 × 15	~2	J H	V I (W)
UDS	34.406250, −5.2000000	Wide	4 × 11	~2	J H	V I (W)

Notes.

a. WFC3/IR filters Y ≡ F105W, J ≡ F125W, and H ≡ F160W.

b. WFC3/UVIS filters UV ≡ F275W, W ≡ F350LP; ACS filters V ≡ F606W, I ≡ F814W, z ≡ F850LP. Parenthesized filters indicate incomplete and/or relatively shallow coverage of the indicated field.

3. METHODOLOGY

3.1 FAST Code

To determine whether a galaxy has an active galactic nucleus, we used the IDL based code Fitting and Assessment of Synthetic Templates (FAST) (Kriek et al. 2009). This code fit a library of stellar population templates, or AGN template, to our raw photometry data. From the output, we generated an SED of the target galaxy (see Figure 4 for an example SED). The most recent update of the FAST code allows for simultaneously fitting of the data for two different components: one for the galaxy (based on the stellar population synthesis model) and the other for non-stellar nuclear

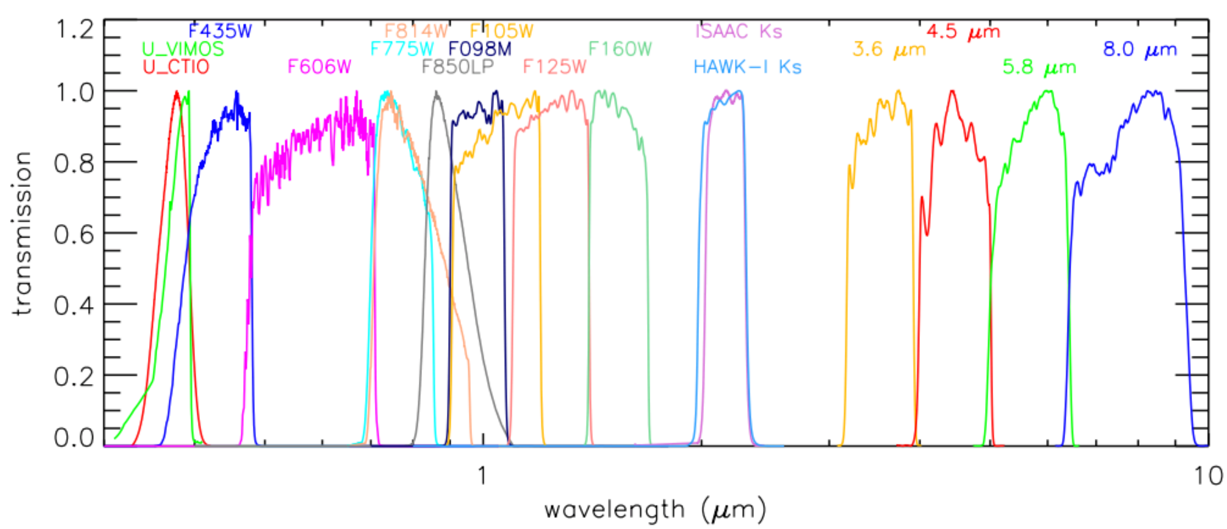


Figure 3. This diagram adapted from Guo et al. (2013) shows the filters and their corresponding wavelength range used for GDS. These filters overlap so that the peaks of each filter combine for a full spectrum of transmission.

light from the AGN. Fitting for both AGN and starlight, the code can identify obscured AGN by separating the photons coming from the AGN from the photons coming from the galaxy's stars.

To determine the accuracy of the code's fitting, we ran test trials using data from the CANDELS EGS field. This test consisted of running FAST on well-known galaxies in the EGS field and matching the characteristics given by FAST with previously measured galaxy properties. We confirmed that FAST accurately gave both galaxy and AGN characteristics, including redshift, luminosity, and mass. The code was able to correctly identify previously known AGN detected via their x-ray emission.

3.2 Cuts and finding AGN

After determining the accuracy of FAST, we began looking for obscured AGN. Starting with EGS, a few characteristics were used to determine the best possible cut to find potentially obscured AGN. Typically, AGN galaxies will be more massive, so only galaxies with a mass over 10^{10} M_{solar} were used. We limit our analysis to galaxies within the redshift range of 0.5 and 1.5 because galaxies at higher redshifts will have their near-infrared emission from obscured AGN redshifted to wavelengths beyond the sensitive range of Hubble. Finally, we only looked at galaxies that had a magnitude greater than 24.5 in the H band. This ensured that enough light came from the source to create an accurate SED. We also excluded x-ray detected galaxies, as we wanted to exclusively look at obscured AGN.

Following these cuts, we input the remaining galaxies in all five CANDELS fields into FAST. The FAST code returned statistics for the fraction of light from an AGN (f_{AGN}) at three specific wavelengths: 1 micron, 5000 Å, and 2800 Å. Using these characteristics, we began to identify potentially obscured AGNs. The 2800 f_{AGN} was disregarded because we did not have enough data in the shorter wavelengths for an accurate calculation. From the 1M and the 5000 f_{AGN}, we took only the galaxies with a 20% AGN light contribution in either wavelength. With these cuts and across all fields of CANDELS, we found 526 potential obscured AGN (EGS: 68; COS: 105; GDS: 92; GDN: 105; UDS: 156).

3.3 Control Group

To compare the morphologies of our potential AGN properly, a control sample was constructed consisting of galaxies with masses similar to those of the AGN hosts. The same cuts were used, but with galaxies with less than 10% of AGN light contribution for all wavelengths. After this sample was found, it was passed through a code, which matched characteristics of each potentially obscured AGN with a galaxy from the control sample. For each AGN host, we randomly selected one unique, nonactive galaxy from whose mass is similar to the AGN host mass. This allows each AGN galaxy to have a counterpart galaxy without an AGN in the control group.

Constructing a mass-matched control sample for this analysis is vital, without taking mass into consideration; the lower mass population, which is predominantly

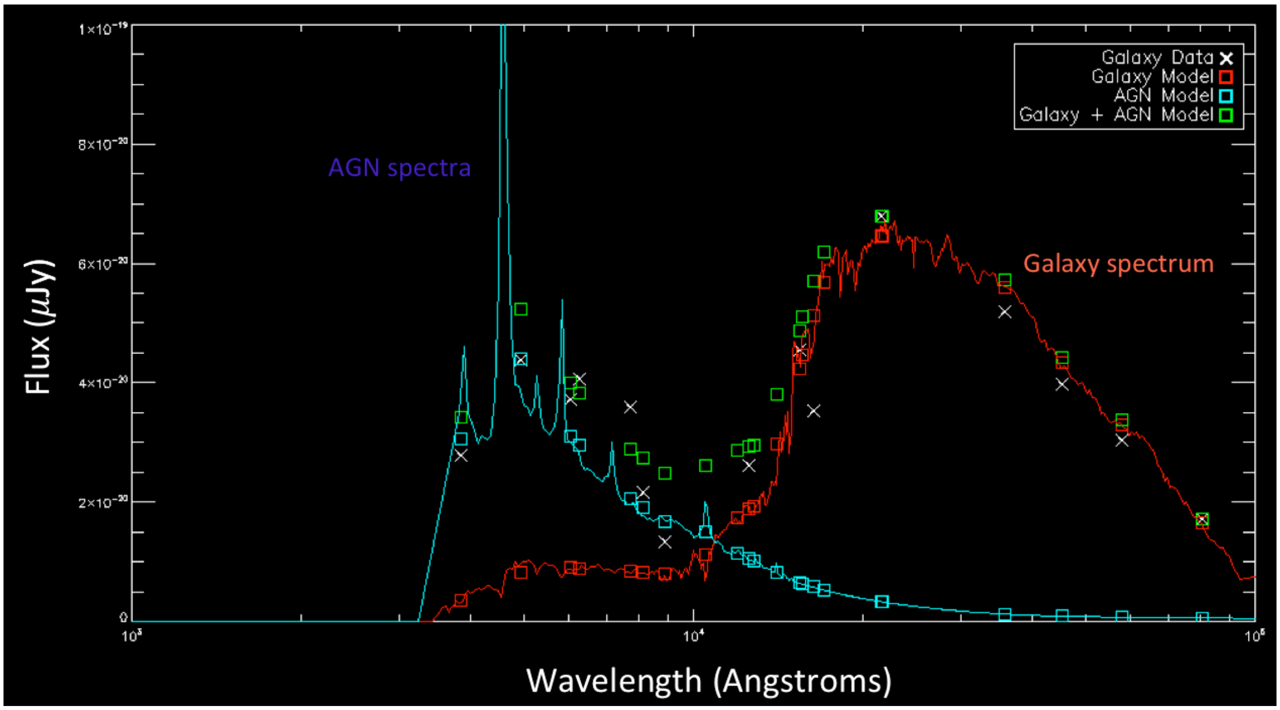


Figure 4. This is an SED of a galaxy with a possible obscured AGN. Flux is plotted on the y-axis and wavelength on the x- axis. The photometry data we currently have is shown with "X". The green boxes indicate the total amount of light FAST predicted at each wavelength. The red boxes indicate the amount of flux coming from the galaxy at specific wavelengths. The red line indicates the high-resolution fits, which is the more detailed flux at every wavelength. The blue boxes indicate the amount of flux coming from the AGN at specific wavelengths, and the blue line indicates the more detailed flux at every wavelength. This specific galaxy comes from EGS and is a strong candidate to be an AGN due the significant contribution of AGN light at the shorter wavelengths.

composed of spiral and irregular galaxies, would dominate any control sample selected potentially biasing any morphological comparison.

3.4 Hubble Thumbnails

Using imaging from the Hubble Space Telescope we were able to obtain thumbnail images of each galaxy at 1.6 microns (H-band). To accomplish this, we used a custom IDL code that made cutouts of galaxy given a specific RA, DEC, and band. Examples of the Hubble thumbnails can be seen in Appendix II.

3.5 Visual Classification

To determine whether a galaxy is merging with a neighboring galaxy, we

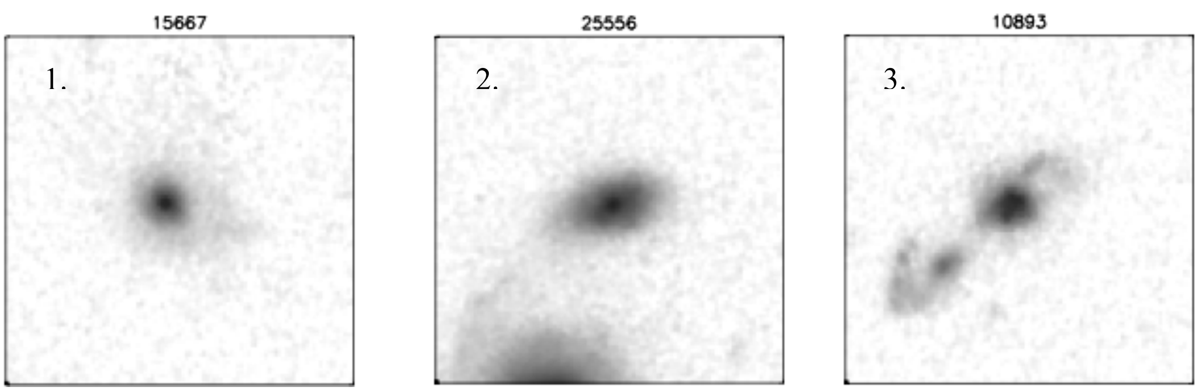


Figure 5. These are examples of the Hubble thumbnails used to classify each galaxy as either disturbed or undisturbed. Thumbnail 1 showcases a galaxy that would be labeled undisturbed. This galaxy is symmetric and there are no nearby galaxies. Thumbnails 2 and 3 showcase galaxies that would be labeled as disturbed. There are clear signs of interactions such as the tails connecting to other galaxies.

employed visual classification. In a double-blind classification, the control and AGN galaxy thumbnails were given to the classifier and catalogued as either disturbed or undisturbed. The classifier was instructed to look for symmetry in undisturbed galaxies and tails or interactions in disturbed galaxies (see Figure 5 for examples of galaxy classification).

4. RESULTS

4.1 Fractions

To test for a difference between the AGN contaminated galaxies and the normal galaxies, we calculated the percent of disturbed galaxies in each condition. Combining all fields, we found that 66.34% of the obscured AGN were disturbed, whereas 59.60% of the galaxies in the control group were disturbed. The percentages of disturbed galaxies in each field are seen in Table 2.

4.2 Error Calculation

To determine the significance of these results, we used the following equation to calculate sigma:

$$\sigma = \frac{\Delta \text{ percentage}}{(\text{error}_1^2 + \text{error}_2^2)^{\frac{1}{2}}}$$

Table 2

	EGS	COS	GDN	GDS	UDS
Obscured AGN	73.52%	81.90%	53.33%	72.82%	57.69%
Control	48.43%	53.33%	62.50%	63.41%	66.99%

The error bars on each fraction reflect the 68.3% binomial confidence limits given the number of sources in each category, which was calculated using the method of Cameron (2010).

For the combined CANDELS field, we found a 2.18 sigma difference between the AGN group and the control group. After excluding UDS, we found a 3.49 sigma difference.

For the individual field we found: EGS: 2.91 sigma, COS: 4.43 sigma, GDN: -1.33 sigma, GDS: 1.32 sigma, and UDS: -1.48 sigma.

5. DISCUSSION

Using a new code, FAST, we were able to locate potentially obscured AGN by producing an SED. Locating the galaxies with a high fraction of AGN light contribution, we compared those galaxies with a control group. We found a loose positive correlation between obscured AGN and disturbed galaxies.

5.1 Removal of UDS

The loose correlation we found ($\sigma = 2.18$) seemed to be low solely due to the UDS field. We decided to exclude UDS from the overall calculation and found a statistically significant sigma of 3.49. Our reasoning behind excluding UDS was that this field was that this field has the least amount of photometry data. Because it has the worst multiband imaging, it also has the shallowest x-ray data. This could be problematic because there may have been strong X-ray galaxies that were not excluded in our experimental condition. X-ray detected AGN are typically undisturbed, so this could pollute the undisturbed category and drive up the percentage of undisturbed galaxies. Future research should test our theory behind excluding UDS. One potential way of doing this is by putting X-ray detected sources back into other fields and recalculating disturbed and undisturbed percentages to see if there is an increase in the undisturbed percentage.

5.2 Implications for AGN Theory

Our finding supports the unproven but commonly accepted theory that AGN turn on due to galaxy mergers (Sanders et al. 1988; Barnes & Hernquist 1991). Previous research testing this theory failed to include obscured AGN, as there was not an accurate way identify them. However, obscured AGN could make up more than half of all AGN (Rovilos et. al. 2014). Thus, our focus on only obscure AGN provides data to accurately representing all AGN. Additionally, with FAST, there is now an alternative way to identify AGN. Through these findings, this study provides breakthroughs in two areas of astronomical quandaries: identifying obscure AGN and what makes AGNs turn on.

Additionally, the on-going theory is that obscured AGN is solely due to the viewing angle of the galaxy. This implies that there should be nothing different between unobscured AGN and obscured AGN. However, the fact that previous research has been unable to find a relationship between unobscured AGN and disturbed galaxies (Kocevski et. al. 2015), but we found a direct relationship between obscured AGN and disturbed galaxies. This could mean that obscured AGN are in a special transition phase during a merger. A phase where there is more dust covering the AGN from the violent nature of galaxy mergers.

5.3 Limitations and Future Direction

Our study had both limitations and areas of needed improvement. Firstly, we only employed one classifier's data for the galaxies. So, future studies could not only gather more classifiers, but also include a more in-depth classification code. Rather than just disturbed or undisturbed, future studies could delineate between the different types of disruption, such as whether a galaxy is actually in a merger or if it disturbed for another reason. Additionally, we only looked at low redshift and high mass galaxies due to the ease of the classification and the more conventional SEDs. Future analysis should widen the criteria used for galaxy cuts. Widening the criteria would also allow for more sources and give the study more variance. Our study presents the basic finding that obscured AGN are linked to disturbed galaxies. Future studies will hopefully replicate our findings and provide further insight to this theory.

6. CONCLUSION

Using a newly updated IDL code, FAST, we were able to identify potential obscured AGN and study their morphologies. We found a total of 526 obscured AGN in all CANDELS fields, all with the following properties: mass over 10^{10} M_{solar}, redshifts between 0.5 and 1.5, and a magnitude brighter than 24.5. After classifying the galaxies as either disturbed or undisturbed, we used a simple binomial distribution and found a slightly significant difference ($\sigma = 2.18$) between obscured AGN and a control group. In conclusion, we found that obscured AGN were slightly more disturbed than their non-AGN counterparts.

7. REFERENCES

Alexander, D. M., Hickox, R. C. 2012, NewAR, 56, 93A

Barnes, J. E., & Hernquist, L. E. 1991, ApJL, 370, L65

Cameron, E. 2010, arXiv:1012.0566

Cisternas, M., Jahnke, K., Inskip, K. J., et al. 2011, ApJ, 726, 57C

Cuo, Y., Ferguson, H. C., Giavalisco, M., et al. 2013, ApJS, 207, 24G

Ellison, S. L., Patton, D. R., Mendel, J. T. et al. 2011, MNRAS, 418, 2043E

Gebhardt, K., Bender, R., Bower, G., et al. 2000, ApJL, 539, L13

Grogin, N. A., Kocevski, D.D., Faber, S. M., et al. 2011, ApJS, 197, 35G

Hickox, R. C., Alexander, D. M. 2018, ARA&A, 56, 625H

Hopkins, P. F., Hernquist, L., Cox, T. J., et al. 2008, ApJS, 175, 356H

Hsu, L., Salvato, M., Nandra, K., et al. 2014, ApJ, 796, 60H

Magorrian, J., Tremaine, S., Richstone, D., et al. 1998, AJ, 115, 2285

Padovani, P., Alexander, D. M., Assef. R. F., et al. 2017, A&Arv, 25, 2P

Kocevski, D. D., Brightman, M., Nandra, K. et al. 2015, ApJ, 814, 104K

Kocevski, D. D., Faber, S. M., Mozena, M. et al. 2012, ApJ, 744, 148K

Koss, M. J., Blecha, L., Bernhard, P., et al. 2018, Natur, 563, 214K

Sanders, D. B., Soifer, B. T., Elias, J. H., et al. 1988, ApJ, 325, 74

Zackrisson, E. 2005, PhDT, 1Z

APPENDIX I: IDL CODE

```
pro agn_cuts, original, cut1, min_1, max_1, cut2, min_2,dataSheet, outFile, p

; Name:
;   agn_cuts
;
; Purpose:
;   Using the given cuts to output the information of the targeted agns
;
; Inputs:
;   orginal = the file with all information necessary to make the cuts
;   cut1, min_1, max_1 = the intial variable cut and its limits
;   cut2, min_2, max_2 = the secondary variable cut and its limits
;   dataSheet = the file with all the information required for FAST for the a
;   paramFile = the file for the parameters for FAST
;
; Optional Keyword Inputs:
;   none
;
; Outputs:
;   positions of all targeted agns
;
; Example:
; agn_cuts, '/Users/rkchan/Research/CANDELS/CANDELS.EGS.1018.sav', 'zbest', 0
;

restore, original

if (isa(egs) eq 1) then struct = egs
if (isa(cos) eq 1) then struct = cos
if (isa(gdn) eq 1) then struct = gdn
if (isa(gds) eq 1) then struct = gds
if (isa(uds) eq 1) then struct = uds

Tags = Tag_names(struct)
g = where(Tags eq STRUPCASE(cut1))
h = where(Tags eq STRUPCASE(cut2))

if (isa(egs) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
```

```
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(cos) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(gdn) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
```

```
  else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(gds) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(uds) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
```

```
        endelse
        endif
        endelse
        endelse
        endelse
        endelse

print, 'Number of target galaxies:', n_elements(tagn)

if (write ne 'yes') then print, 'Skipping writing .cat file' else begin
if (write eq 'yes') then begin

;readcol, dataSheet, id, z_spec, Flux_u_cfht, FluxErr_u_cfht, Flux_g_cfht, Fl
;index = crossmatch_ids(struct[tagn].id, id)

lread, dataSheet, lines

; for i = 0,n_elements(tagn)-1 do print, struct[tagn[i]].id, id[tagn[i]], lin

print, n_elements(tagn)

  print, 'writing data file'
  openw, 1, outFile
  printf, 1, lines[0]
;  for i = 0,n_elements(tagn)-1 do print, lines[tagn[i]+1]
  for i = 0,n_elements(tagn)-1 do printf, 1, lines[tagn[i]+1]
  close, 1
endif
endelse

if (test eq !NULL) then print, 'Skipping FAST' else begin
if (test eq 'yes') then begin
    print, 'running fast'
    fast, param = paramFile
    print, 'fast complete'
    endif
    endelse

print, 'agn_cuts completed'

END

; restore, '/Users/rkchan/Research/CANDELS/CANDELS.EGS.1018.sav'
; tagn = where(egs.zbest lt 3 and egs.zbest gt 0.3 and egs.lx gt 43.8)
; lread, '/Users/rkchan/Research/CANDELS/CANDELS.EGS.F160W.v2.id_zbest_phot.c
```

```
; openw, 1, '/Users/rkchan/Research/CANDELS/EGS_test/target_agn.cat'
; printf, 1, lines[0]
; for i = 0,n_elements(tagn)-1 do printf, 1, lines[tagn[i]]
```

```
pro agn_plot, inFolder, name, galID, field

; Name:
;   agn_plot
;
; Purpose:
;   Read input photometry from a FAST output and plot the SED
;
; Inputs:
;   inFolder = the folder where the input and output for FAST is
;   name = the name used to label the respective FAST input/output
;   cWaveFile = location of the targeted central wavelength file
;   galID = the id of the galaxy wanted
;
; Optional Keyword Inputs:
;   none
;
; Outputs:
;   graph of SED
;
; Example:
;   agn_plot, '/Users/rkchan/Research/FAST_tests/EGS_test', 'target_agn', 202
;
; Notes:
;   Works specifically for the

if (field eq 'egs') then begin
    readcol, inFolder + '/' + name +'.cat', id, z_spec, Flux_u_cfht, FluxErr_
    igal = where(id eq galID)
    fnu = [Flux_u_cfht[igal], Flux_g_cfht[igal], Flux_r_cfht[igal], Flux_i_cf
    endif

if (field eq 'cos') then begin
    readcol2, inFolder + '/' + name +'.cat', id, z_spec, CFHT_U_FLUX, CFHT_U_
    igal = where(id eq galID)
    fnu = [CFHT_U_FLUX[igal], CFHT_G_FLUX[igal], CFHT_R_FLUX[igal], CFHT_I_FL
    endif


if (field eq 'gdn') then begin
    readcol, inFolder + '/' + name +'.cat', id, z_spec, KPNO_U_FLUX, KPNO_U_F
    igal = where(id eq galID)
    fnu = [KPNO_U_FLUX[igal], LBC_U_FLUX[igal],ACS_F435W_FLUX[igal], ACS_F606
    endif

```

```

if (field eq 'gds') then begin
    readcol, inFolder + '/' + name +'.cat', id, z_spec, CTIO_U_FLUX, CTIO_U_F
    igal = where(id eq galID)
    fnu = [CTIO_U_FLUX[igal], VIMOS_U_FLUX[igal], ACS_F435W_FLUX[igal], ACS_F
    endif

if (field eq 'uds') then begin
    readcol, inFolder + '/' + name +'.cat', id, z_spec, Flux_U_cfht, Fluxerr_
    igal = where(id eq galID)
    fnu = [Flux_U_cfht[igal], Flux_B_subaru[igal], Flux_V_subaru[igal], Flux_
    endif

; Read central wavelengths
cWaveFile = inFolder + '/BEST_FITS/'+ name +'_'+ trim(id[igal])+'.input_res.f
readcol, cWaveFile, lam_fit, f_fit, format='D,D'

lambda = lam_fit


fnu_jan = fnu * 1e-6
flam = fnu_jan * (3e-5)/(lambda^2.0)
;  plot, lambda, flam/max(flam)>0, psym=7, yrange=[0,1.2]




; Now read best fit

; Low Res
best_gal_fit = inFolder + '/BEST_FITS/'+ name +'_'+ trim(id[igal])+'.input_re
readcol, best_gal_fit, lfit, ffit_gal, format='D,D'
best_agn_fit = inFolder + '/BEST_FITS/'+ name + '_' + trim(id[igal])+'.AGN.in
readcol, best_agn_fit, lfit, ffit_agn, format='D,D'

; High Res
best_gal_fit2 = inFolder + '/BEST_FITS/'+ name + '_' + trim(id[igal])+'.fit'
readcol, best_gal_fit2, lfit2a, ffit_gal2, format='D,D'
best_agn_fit2 = inFolder + '/BEST_FITS/'+ name + '_' + trim(id[igal])+'.AGN.f
readcol, best_agn_fit2, lfit2b, ffit_agn2, format='D,D'

; Total

total_fit = ffit_gal + ffit_agn

```

```
; fudge = 2.5

; fudge factor that makes max flam = max total_fit
fudge = (max(flam)/max(total_fit)) * 1e19

; fudge factor that makes average of flam = average of total_fit (sometimes w
; fudge = (mean(flam)/mean(total_fit)) * 1e19

print, 'the correction factor is: ', fudge

plot, lambda, flam / fudge, psym=7, yrange=[0,max(flam/fudge)*1.2], symsize=1

; plot, lambda, flam, psym=7, yrange=[0,max(flam)*1.2], symsize=2, /xlog

mkct2
oplot, lfit, ffit_gal * 1e-19 , psym=6, symsize=1.5, color=2
oplot, lfit, ffit_agn * 1e-19 , psym=6, symsize=1.5, color=7

oplot, lfit, total_fit * 1e-19 , psym=6, symsize=1.5, color=3

;oplot, lfit2a, ffit_gal2 * 1e-19, psym=3, symsize=2, color=2
;oplot, lfit2b, ffit_agn2 * 1e-19, psym=3, symsize=2, color=7

oplot, lfit2a, ffit_gal2 * 1e-19, color=2
oplot, lfit2b, ffit_agn2 * 1e-19, color=7

; forprint, flam, flam/fudge, total_fit, ffit_gal, ffit_agn

;.compile legend
legend2, ['Galaxy Data', 'Galaxy Model', 'AGN Model', 'Galaxy + AGN Model'],

end
```

```
pro binomial_fraclim2,k,n,c,lo_lim,up_lim

if n_params() lt 3 then begin
   print,' syntax: binomial_lim,k,n,c,lo_lim,up_lim'
   print,'         where'
   print,'         o "k" is the observed (integer) number of galaxies with a
   print,'         o "n" is the observed (integer) number of galaxies in the
   print,'         o "c" is the desired confidence interval (0.683 = 1sigma,
   print,'         o "lo_lim" is the (approximate) Binomial lower limit ON FR
   print,'         o "up_lim" is the (approximate) Binomial upper limit ON FR
   print,''
   print,'         --> This version gives the upper and lower limits on the f
   print,''
   print,' (reference: Cameron et al. 2010)'
   print,''
   return
endif


z = FINDGEN(10000)*0.0001
Beta = IBETA(k+1,n-k+1,z)
il = VALUE_LOCATE(Beta,(1-c)/2)
ul = VALUE_LOCATE(Beta,1-(1-c)/2)
p_lower = z[il]
p_upper = z[ul]


lo_lim = p_lower
up_lim = p_upper

print, lo_lim, up_lim

return

end

```

```
pro compare_plot, original, cut1, min_1, max_1, cut2, min_2, max_2, var1, out

; Name:
;   compare_plot
;
; Purpose:
;   Read output  from a FAST output and plots it against the original code
;
; Inputs:
;   inFolder = the folder where the input and output for FAST is
;   name = the name used to label the respective FAST input/output
;
; Optional Keyword Inputs:
;   none
;
; Outputs:
;   graph of SED
;
; Example:
;   compare_plot, '/Users/rkchan/Research/CANDELS/CANDELS.EGS.1018.sav', 'zbe
;
; Notes:
;

restore, original

if (isa(egs) eq 1) then struct = egs
if (isa(cos) eq 1) then struct = cos
if (isa(gdn) eq 1) then struct = gdn
if (isa(gds) eq 1) then struct = gds
if (isa(uds) eq 1) then struct = uds

Tags = Tag_names(struct)
g = where(Tags eq STRUPCASE(cut1))
h = where(Tags eq STRUPCASE(cut2))

tgal = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(h) gt mi

mass_source = egs[tgal].mass
z_source = egs[tgal].zbest
sfr_source = egs[tgal].sfr

print, 'Number of target galaxies:', n_elements(tgal)

readcol, output, id, z_b, ltau, metal, lage, Av, lmass, lsfr, lssfr, la2t, L2
```

```
mass_diff = lmass - mass_source
sfr_diff = lsfr - sfr_source
z_diff = z_b - z_source

galNumber = intarr(n_elements(mass_diff))

for i = 0, n_elements(mass_diff) do galNumber[i-1] = i

bound = max(mass_diff)
if (bound lt -min(mass_diff)) then bound = abs(min(mass_diff))

plot, galNumber, mass_diff, yrange=[-bound*1.2,bound*1.2], psym=4
mkct2
oplot, galNumber, z_diff, psym=2, symsize=2

forprint, galNumber, sfr_source, 10^(lsfr), sfr_diff

end
```

```
pro make_thumb, original, output1, output2, box_in, image, band, inFolder, su


if n_params() lt 5 then begin
   print,''
   print,'syntax: gal_info, ra, dec, id, box_(in arcsec), image, band, [rootn
   print,''
   print,'         Script to create fits cutouts using any mosaic you choose.
   print,'         Optionally, script with create a hardcopy mosaic image of
   print,''
   print,'         Primary arguements for making fits cutouts:'
   print,'           -> original = the file with all information necessary to
   print,'           -> output = the output to FAST you want to graph
   print,'           -> box = size (radius) of desired thumbnails in arcsec'
   print,'           -> image = name of image mosaic to use'
   print,'           -> band = band name (i.e. H, V, etc - used in thumbnail
   print,'           -> inFolder = the folder where the input and output for
   print,'           -> subname =the name used to label the respective FAST i
   print,'           -> outdir = output directory (overrides default ./thumbs
   print,'           -> rootname = naming option for thumbs, see below.'
   print,'           -> prefix = naming option for thumbs, see below.'
   print,'           -> scl = optional pixel scale of image.  Default is 0.06
   print,''
   print,'         Optional arguements for hardcopy thumbnail plot:'
   print,'           -> plotname = name of output ps file (string)'
   print,'           -> path = path to directory containing fits thumbnails (
   print,'           -> title = array of object IDs.  If not specified, ID ta
   print,'           -> ngal = number of galaxies to display per page'
   print,'           -> maxfactor = sets max over the image_max for stretch.
   print,'           -> stretch = beta factor for asinh scaling.  Decrease to
   print,''
   print,'         Cutout naming options include rootnames, prefixes and suff
   print,'           -> Naming convention: [rootname].[prefix][id].[suffix].[
   print,''
   print,'         By default, fits cutouts are placed in the directory ./thu
   print,'         This can be changed via the outdir optional parameter.'
   print,''
   print,'         Plotting Defaults:'
   print,'           -> maxfactor = 2.0'
   print,'           -> stretch = 0.025'
   print,'           -> astro = 0 (no WCS info will be shown)'
   print,'           -> path = ./'
   print,'           -> ngal = 4'
   print,''
   print,'         Additional Notes:'
```

```
   print,'           -> The box size is a radius (i.e. thumbnail size = box*2
   print,'           -> The outdir name should end in a slash (i.e. ./output/
   print,''
   print,"         Example:  make_thumb, '/Users/rkchan/Research/CANDELS/CAND
   print,''
   return
endif

restore, original

if (isa(egs) eq 1) then struct = egs
if (isa(cos) eq 1) then struct = cos
if (isa(gdn) eq 1) then struct = gdn
if (isa(gds) eq 1) then struct = gds
if (isa(uds) eq 1) then struct = uds

readcol, output1, id_fast,zbest,ltau,metal,lage,Av,lmass,lsfr,lssfr,la2t,lsfr

readcol, output2, id_fast2,zbest2,ltau2,metal2,lage2,Av2,lmass2,lsfr2,lssfr2,

id = [id_fast, id_fast2]

g = crossmatch_ids(id, struct.id)

ra = struct[g].ra
dec = struct[g].dec

; if title eq !NULL then title = struct[g].id
; title = struct[g].id
title = sindgen(n_elements(id_fast)+n_elements(id_fast2))
for i=0, n_elements(title)-1 do title[i] = strn(struct[g[i]].id)

; Set output directory
outdir_name = './thumbs_fits/'
if keyword_set(outdir) then outdir_name = outdir
outdir = outdir_name

; Make outdir if neccessary
spawn, 'mkdir -vp '+outdir
print, ''
print, 'Placing thumbnails in directory: '+outdir

; Read Naming Options
root = ''
if keyword_set(rootname) then root=rootname
```

```
pfx = ''
if keyword_set(prefix) then pfx=prefix
sfx = ''
if keyword_set(suffix) then sfx=suffix
if keyword_set(segmap) then segmap_name = segmap

; Set Scale (default
if keyword_set(scl) then scl=scl else scl=0.06 ; Default is for WFC3


; Make array to hold fits names
outfits = sindgen(n_elements(ra))


; Determine if plot requested

if keyword_set(plotname) then begin

    ; Set number of gals in mosaic
    if keyword_set(ngal) then begin
        ngalin = ngal
    endif else ngalin = 4

    ; Set p.multi
    !p.multi = [0,3,ngalin]

    ; Set output file
    outfile = strn(plotname)
    psopen, outfile, /portrait, /color

    ; Set path
    fullpath = outdir

    ; Set stretch and minmax
    if keyword_set(stretch) then begin
        betain = stretch
    endif else betain = 0.025
    if keyword_set(maxfactor) then begin
        xfactor = maxfactor
    endif else xfactor = 2.0
    if keyword_set(astro) then begin
        astroin = astro
    endif else astroin = 0

endif
```

```
;; Make thumbnails
print, ""
print, "Making thumbnails..."
im = readfits(image, h1)  &$

if (n_elements(box_in) eq 1) then begin
    box = replicate(box_in/scl, n_elements(ra))
endif else box = box_in/scl

for i=0, n_elements(ra)-1 do begin &$
  print, "Working on "+strn(id[i])  &$
  nxmx = sxpar(h1,'NAXIS1')-1  &  nymx = sxpar(h1,'NAXIS2')-1    &$
  if (sfx ne '') then outfits[i] = outdir+root+'.'+pfx+strn(trim(id[i]))+'.'+
  if (sfx eq '') then outfits[i] = outdir+root+'.'+pfx+strn(trim(id[i]))+'.'+
  adxy,h1,ra[i],dec[i],x0,y0  &$

  ; Proceed only if sources falls within mosaic
  if (x0 lt nxmx and x0 gt 0 and y0 lt nymx and y0 gt 0) then begin
    hextract,im,h1,im2,h1b,(x0-box[i])>0,(x0+box[i])<nxmx,(y0-box[i])>0,(y0+b
    writefits, outfits[i], im2, h1b  &$
  endif else begin
    ; Make dummy thumbnail if image falls outside mosaic
    print, 'Source '+strn(i)+' falls outside mosaic!'
    hextract,im,h1,im2,h1b,0,(0+2.0*box[i])<nxmx,0,(0+2.0*box[i])<nymx   &$
    writefits, outfits[i], im2-im2, h1b  &$
  endelse

endfor




;; Make Hardcopy
if keyword_set(plotname) then begin

    print, ""
    print, "Making Hardcopy Mosaic..."

    for i=0, n_elements(ra)-1 do begin &$

      ; Read image
      im = readfits(outfits[i], h1)  &$
```

```

      ; Display B/W thumbs
      im2 = im
      im2[0,0] = max(im)*xfactor
      astrim_asinh, (im2>(0.0)<max(im2)), h1, beta=betain,grid=0,charsize=1.0
    endfor

    psclose
endif


print, ""
print, "Script Complete!"
print, ""

end
```

```
pro plot_fit_wimage, original, output, box_in, image, band, inFolder, subname


if n_params() lt 5 then begin
   print,''
   print,'syntax: gal_info, ra, dec, id, box_(in arcsec), image, band, [rootn
   print,''
   print,'         Script to create fits cutouts using any mosaic you choose.
   print,'         Optionally, script with create a hardcopy mosaic image of
   print,''
   print,'         Primary arguements for making fits cutouts:'
   print,'           -> original = the file with all information necessary to
   print,'           -> output = the output to FAST you want to graph
   print,'           -> box = size (radius) of desired thumbnails in arcsec'
   print,'           -> image = name of image mosaic to use'
   print,'           -> band = band name (i.e. H, V, etc - used in thumbnail
   print,'           -> inFolder = the folder where the input and output for
   print,'           -> subname =the name used to label the respective FAST i
   print,'           -> outdir = output directory (overrides default ./thumbs
   print,'           -> rootname = naming option for thumbs, see below.'
   print,'           -> prefix = naming option for thumbs, see below.'
   print,'           -> scl = optional pixel scale of image.  Default is 0.06
   print,''
   print,'         Optional arguements for hardcopy thumbnail plot:'
   print,'           -> plotname = name of output ps file (string)'
   print,'           -> path = path to directory containing fits thumbnails (
   print,'           -> title = array of object IDs.  If not specified, ID ta
   print,'           -> ngal = number of galaxies to display per page'
   print,'           -> maxfactor = sets max over the image_max for stretch.
   print,'           -> stretch = beta factor for asinh scaling.  Decrease to
   print,''
   print,'         Cutout naming options include rootnames, prefixes and suff
   print,'           -> Naming convention: [rootname].[prefix][id].[suffix].[
   print,''
   print,'         By default, fits cutouts are placed in the directory ./thu
   print,'         This can be changed via the outdir optional parameter.'
   print,''
   print,'         Plotting Defaults:'
   print,'           -> maxfactor = 2.0'
   print,'           -> stretch = 0.025'
   print,'           -> astro = 0 (no WCS info will be shown)'
   print,'           -> path = ./'
   print,'           -> ngal = 4'
   print,''
   print,'         Additional Notes:'
```

```
    print,'            -> The box size is a radius (i.e. thumbnail size = box*2
    print,'            -> The outdir name should end in a slash (i.e. ./output/
    print,''
    print,"          Example:  plot_fit_wimage, '/Users/rkchan/Research/CANDELS
    print,''
    return
endif

restore, original

if (isa(egs) eq 1) then begin
    struct = egs
    field = 'egs'
    endif
if (isa(cos) eq 1) then begin
    struct = cos
    field = 'cos'
    endif
if (isa(gdn) eq 1) then begin
    struct = gdn
    field = 'gdn'
    endif
if (isa(gds) eq 1) then begin
    struct = gds
    field = 'gds'
    endif
if (isa(uds) eq 1) then begin
    struct = uds
    field = 'uds'
    endif

readcol, output, id_fast,zbest,ltau,metal,lage,Av,lmass,lsfr,lssfr,la2t,lsfr1

id = id_fast
ra = struct[id-1].ra
dec = struct[id-1].dec

; if title eq !NULL then title = struct[id-1].id
; title = struct[id-1].id
title = sindgen(n_elements(id_fast))
for i=0, n_elements(title)-1 do title[i] = strn(struct[id[i]-1].id) + ', 2K:



; Set output directory
```

```
outdir_name = './thumbs_fits/'
if keyword_set(outdir) then outdir_name = outdir
outdir = outdir_name

; Make outdir if neccessary
spawn, 'mkdir -vp '+outdir
print, ''
print, 'Placing thumbnails in directory: '+outdir

; Read Naming Options
root = ''
if keyword_set(rootname) then root=rootname
pfx = ''
if keyword_set(prefix) then pfx=prefix
sfx = ''
if keyword_set(suffix) then sfx=suffix
if keyword_set(segmap) then segmap_name = segmap

; Set Scale (default
if keyword_set(scl) then scl=scl else scl=0.06 ; Default is for WFC3


; Make array to hold fits names
outfits = sindgen(n_elements(ra))


; Determine if plot requested

if keyword_set(plotname) then begin

    ; Set number of gals in mosaic
    if keyword_set(ngal) then begin
        ngalin = ngal
    endif else ngalin = 4

    ; Set p.multi
    ; !p.multi = [0,2,ngalin]
    !p.multi = [0,2,4]

    ; Set output file
    outfile = strn(plotname)
    psopen, outfile, /portrait, /color

    ; Set path
    fullpath = outdir
```

```

    ; Set stretch and minmax
    if keyword_set(stretch) then begin
        betain = stretch
    endif else betain = 0.025
    if keyword_set(maxfactor) then begin
        xfactor = maxfactor
    endif else xfactor = 2.0
    if keyword_set(astro) then begin
        astroin = astro
    endif else astroin = 0

endif



;; Make thumbnails
print, ""
print, "Making thumbnails..."
im = readfits(image, h1)  &$

if (n_elements(box_in) eq 1) then begin
    box = replicate(box_in/scl, n_elements(ra))
endif else box = box_in/scl

for i=0, n_elements(ra)-1 do begin &$
  print, "Working on "+strn(id[i])  &$
  nxmx = sxpar(h1,'NAXIS1')-1  &  nymx = sxpar(h1,'NAXIS2')-1    &$
  if (sfx ne '') then outfits[i] = outdir+root+'.'+pfx+strn(trim(id[i]))+'.'+
  if (sfx eq '') then outfits[i] = outdir+root+'.'+pfx+strn(trim(id[i]))+'.'+
  adxy,h1,ra[i],dec[i],x0,y0  &$

  ; Proceed only if sources falls within mosaic
  if (x0 lt nxmx and x0 gt 0 and y0 lt nymx and y0 gt 0) then begin
    hextract,im,h1,im2,h1b,(x0-box[i])>0,(x0+box[i])<nxmx,(y0-box[i])>0,(y0+b
    writefits, outfits[i], im2, h1b  &$
  endif else begin
    ; Make dummy thumbnail if image falls outside mosaic
    print, 'Source '+strn(i)+' falls outside mosaic!'
    hextract,im,h1,im2,h1b,0,(0+2.0*box[i])<nxmx,0,(0+2.0*box[i])<nymx   &$
    writefits, outfits[i], im2-im2, h1b  &$
  endelse

endfor

```

```



;; Make Hardcopy
if keyword_set(plotname) then begin

    print, ""
    print, "Making Hardcopy Mosaic..."

    j = 0
    for i=0, n_elements(ra)-1 do begin &$

      j = j+1

      ; Read image
      im = readfits(outfits[i], h1)  &$

      ; Display B/W thumbs
      im2 = im
      im2[0,0] = max(im)*xfactor



      ; Manually set position of plot window (With Axis Labels)
      if (j eq 1) then !p.position = [0.0, 0.7375, 0.3, 0.9300]
      if (j eq 2) then !p.position = [0.0, 0.5050, 0.3, 0.6975]
      if (j eq 3) then !p.position = [0.0, 0.2725, 0.3, 0.4650]
      if (j eq 4) then !p.position = [0.0, 0.0400, 0.3, 0.2325]

      ; ALT: Manually set position of plot window (No Axis Labels)
;     if (j eq 1) then !p.position = [0.0, 0.7275, 0.3, 0.9300]
;     if (j eq 2) then !p.position = [0.0, 0.4950, 0.3, 0.6975]
;     if (j eq 3) then !p.position = [0.0, 0.2625, 0.3, 0.4650]
;     if (j eq 4) then !p.position = [0.0, 0.0300, 0.3, 0.2325]

      astrim_asinh, (im2>(0.0)<max(im2)), h1, beta=betain,grid=0,charsize=1.0


      ; Manually set position of plot window (With Axis Labels)
      if (j eq 1) then !p.position = [0.39, 0.7375, 1.0, 0.9300]
      if (j eq 2) then !p.position = [0.39, 0.5050, 1.0, 0.6975]
      if (j eq 3) then !p.position = [0.39, 0.2725, 1.0, 0.4650]
      if (j eq 4) then !p.position = [0.39, 0.0400, 1.0, 0.2325]

      ; Manually set position of plot window (No Axis Labels)
```

```
;      if (j eq 1) then !p.position = [0.36, 0.7275, 1.0, 0.9300]
;      if (j eq 2) then !p.position = [0.36, 0.4950, 1.0, 0.6975]
;      if (j eq 3) then !p.position = [0.36, 0.2625, 1.0, 0.4650]
;      if (j eq 4) then !p.position = [0.36, 0.0300, 1.0, 0.2325]


;       gal_plot, inFolder, subname, i

        agn_plot, inFolder, subname, id[i], field

       if (j eq 4) then j = 0

    endfor

    psclose
endif


print, ""
print, "Script Complete!"
print, ""

end
```

```
pro xray_compare, output, original, cutoff ,cut1, min_1, max_1, cut2, min_2,

; Name:
;   xray_compare
;
; Purpose:
;   Read output  from a FAST output
;
; Inputs:
;   output - output file from fast with agn comparison
;
; Optional Keyword Inputs:
;   none
;
; Outputs:
;   none
;
; Example:
;   xray_compare,'/Users/rkchan/Research/AGN_FAST/EGS_Second_Cut/egs_cut2.fou
;
; Notes:
;
restore, original

if (isa(egs) eq 1) then struct = egs
if (isa(cos) eq 1) then struct = cos
if (isa(gdn) eq 1) then struct = gdn
if (isa(gds) eq 1) then struct = gds
if (isa(uds) eq 1) then struct = uds

Tags = Tag_names(struct)
g = where(Tags eq STRUPCASE(cut1))
h = where(Tags eq STRUPCASE(cut2))

if (isa(egs) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
```

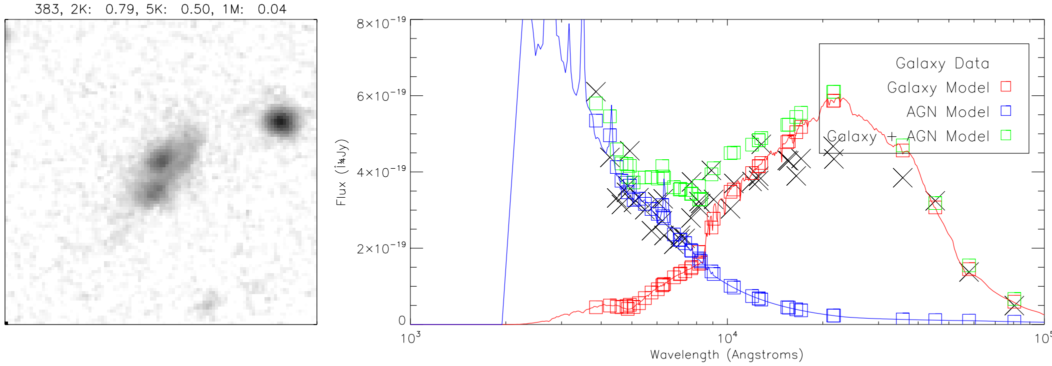
```
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(cos) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(gdn) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
```

```
        endelse
        endelse
        endelse
    endif else begin
if (isa(gds) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
    endif else begin
if (isa(uds) eq 1) then begin
    if (max_2 ne !NULL and var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'first'
        endif else begin
    if (var eq 'yes') then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'second'
        endif else begin
    if (max_2 ne !NULL) then begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'third'
        endif $
    else begin
        tagn = where(struct.(g) lt max_1 and struct.(g) gt min_1 and struct.(
        print, 'fourth'
        endelse
        endelse
        endelse
        endif
        endelse
```

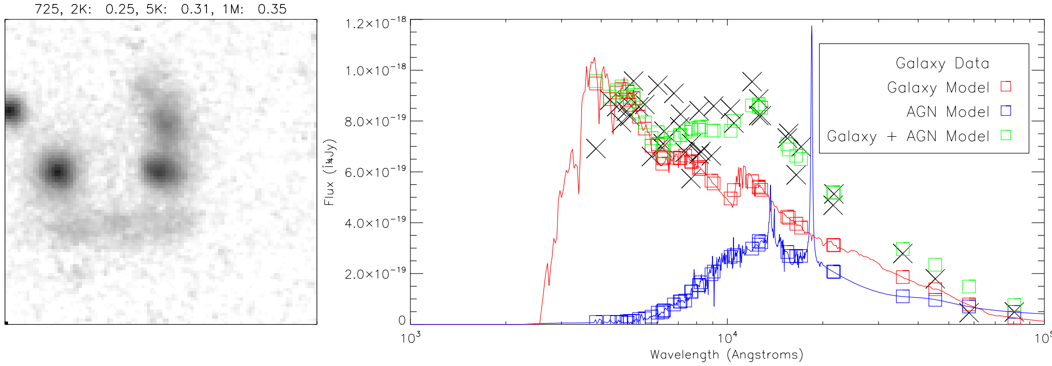
```
        endelse
        endelse
        endelse

print, 'Number of target galaxies:', n_elements(tagn)

xray_source = struct[tagn].flag_xray

readcol, output, id,zbest,ltau,metal,lage,Av,lmass,lsfr,lssfr,la2t,lsfr100,lm

; high_ratio = where(fagn1m gt 0.5 or fagn5000 gt 0.5 or fagn2800 gt 0.5)
high_ratio = where(fagn1m gt cutoff or fagn5000 gt cutoff)

potentials = xray_source[high_ratio]

special = high_ratio[where(potentials eq 0)]

print, 'Number of target galaxies with high ratio and zero xray' , n_elements

lread, output, lines

if (writefile eq 'yes') then begin
  print, 'writing data file'
  openw, 1, outFile
  printf, 1, lines[0:16]
;  for i = 0,n_elements(tagn)-1 do print, lines[special]
  for i = 0,n_elements(special)-1 do printf, 1, lines[special[i]+17]
  close, 1
endif

print, 'Finished Xray_compare'

end
```

APPENDIX II: THUMBNAILS AND SED FIT EXAMPLES

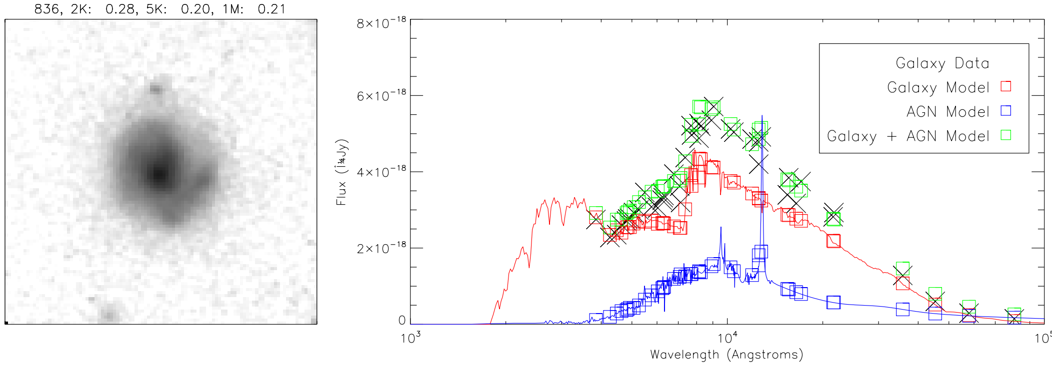
383, 2K: 0.79, 5K: 0.50, 1M: 0.04



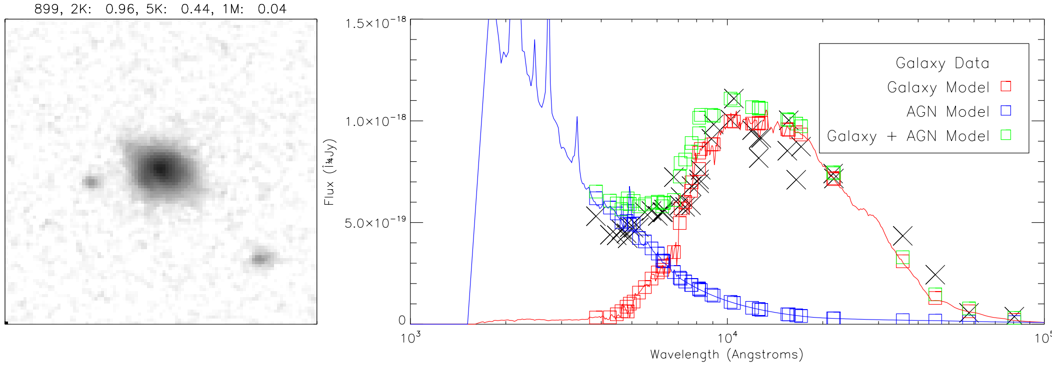
725, 2K: 0.25, 5K: 0.31, 1M: 0.35



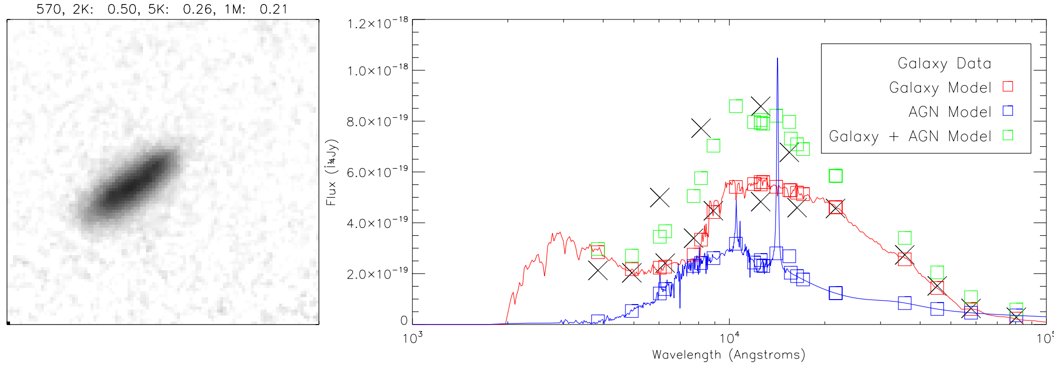
836, 2K: 0.28, 5K: 0.20, 1M: 0.21



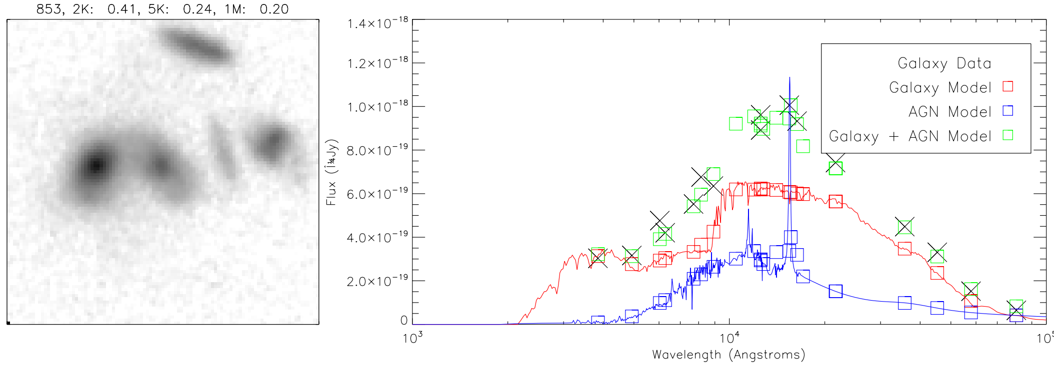
899, 2K: 0.96, 5K: 0.44, 1M: 0.04



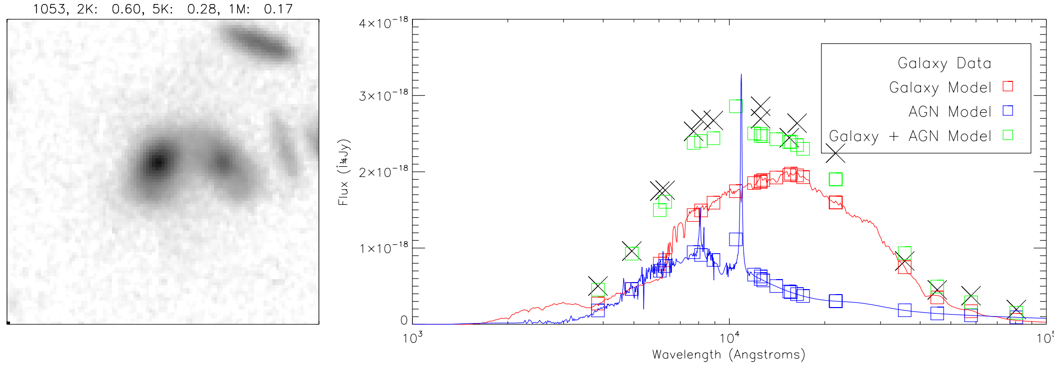
570, 2K: 0.50, 5K: 0.26, 1M: 0.21



853, 2K: 0.41, 5K: 0.24, 1M: 0.20



1053, 2K: 0.60, 5K: 0.28, 1M: 0.17



1489, 2K: 0.33, 5K: 0.19, 1M: 0.24