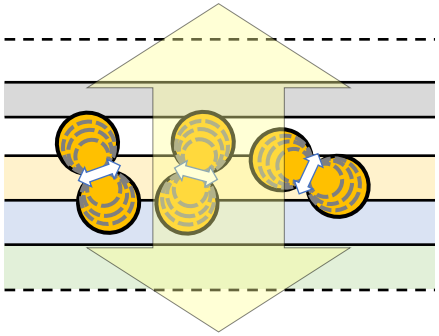
# Dipole Orientation Factors and Intrinsic Spectrum Extraction

**-DOFExtractor.py**

Author: Wei-Kai Lee

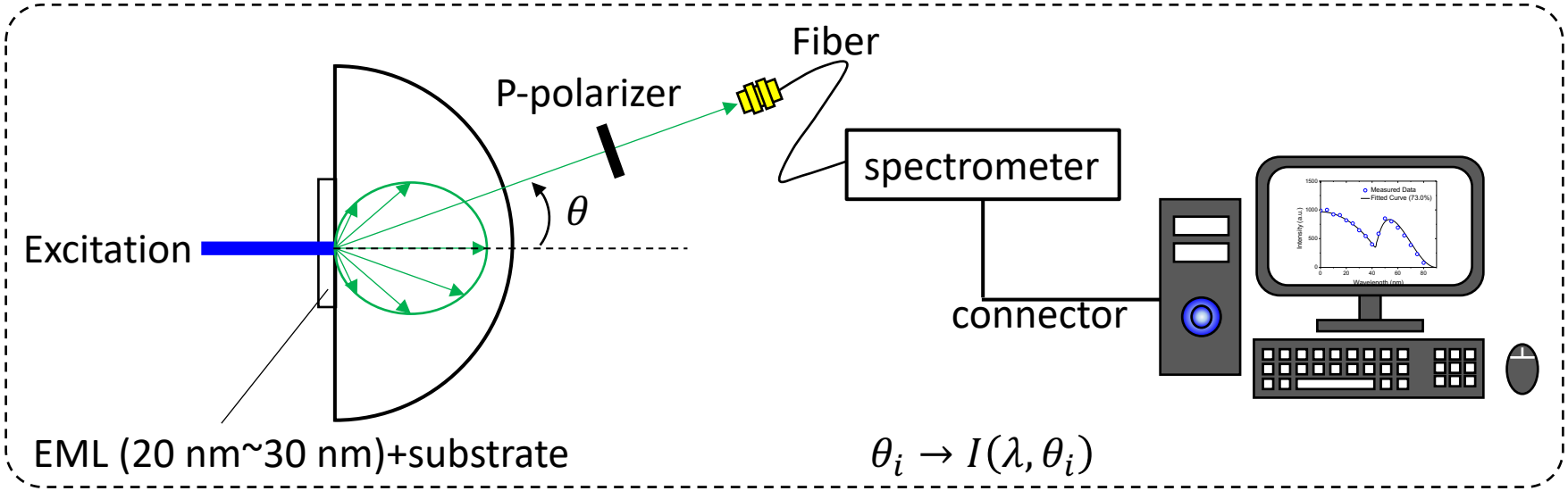


# Objective

- This function help find the dipole orientation factors and the intrinsic spectrum from angle- and p-polarization resolved spectrum experiment.

# Experimental Setup

**Software analyzer**




**Determination of the emitting dipole orientation of an emitting layer:**

To determine the emitting dipole orientation in a molecular emitting film, angle-resolved and polarization resolved PL measurements were performed. The sample consisted of a fused silica substrate with the 20-nm-thick emitting layer (EML). The sample was attached to a fused silica half cylinder prism by index matching liquid. The excitation of the samples was performed with the 325-nm line of the continuous-wave He:Cd laser (excitation source) with a fixed excitation angle of 45°. The emission angle was changed by use of an automatic rotation stage. The spectra were resolved by using a p-polarizing filter and were measured by a fiber optical spectrometer. The angle-dependent p-polarized emission intensity at the peak wavelength of the PL spectrum of the emitting layer was detected. The emitting dipole orientation (the horizontal dipole ratio $\Theta_{//}$) was then determined by least square fitting of the measured angle-dependent p-polarized emission intensity with calculated results.

# Measurement

Variables: $\theta_i$ for $i = 1 \sim N$
P polarization spectra : $I(\lambda_j, \theta_i)$ for $i = 1 \sim n$ and $j = 1 \sim m$

From the electromagnetic wave model, the corresponding spectra can then be simulated. (i.e. $\tilde{I}(\lambda_j, \theta_i; P_o, \eta_{1\sim5})$, where $P_o$ is the intrinsic emitting power and $\eta_{1\sim5}$ are dipole orientation factors)

# Optimization Model-1

**Point by point model**

Loss function/Goodness of fit

$$Loss(\lambda_j, P_o, \eta_{1\sim5}) = \frac{1}{n-f}\sum_{i=1}^{n}\left(\tilde{I}(\lambda_j, \theta_i; P_o, \eta_{1\sim5}) - I(\lambda_j, \theta_i)\right)^2$$

$n$: number of angles

$f$ : number of free variables

Objective:

$$\bar{P}_o, \bar{\eta}_{1\sim5} = \min_{P_o, \eta_{1\sim5}} Loss(\lambda_j, P_o, \eta_{1\sim5})$$

From EM model $\tilde{I}(\lambda_j, \theta_i; P_o, \eta_{1\sim5})$ is linear to $P_o$ and $P_o \times \eta_{1\sim5}$.

As a result, the parameters of point by point model are evaluated by pseudo-inverse.

# Optimization Model-2

**Wavelength dependent model (still under research)**

Loss function/Goodness of fit

$$Loss(\boldsymbol{\alpha}) = \frac{1}{n \times m - f} \sum_{i,j=1}^{n,m} \left( \tilde{I}(\lambda_j, \theta_i; P_o, \eta_{1\sim5}) - I(\lambda_j, \theta_i) \right)^2$$

$n$: number of angles

$\boldsymbol{\alpha}$ : vector of free variables

$f$ : number of free variables ($\boldsymbol{\alpha}$)

We suppose $P_o$ and $\eta_{1\sim5}$ are functions of wavelength.

Objective:

$$\bar{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} Loss(\boldsymbol{\alpha})$$

From EM model $\tilde{I}(\lambda_j, \theta_i; P_o, \eta_{1\sim5})$ is nonlinear to $\boldsymbol{\alpha}$. As a result, $\boldsymbol{\alpha}$ is evaluated by the gradient descent (Adam).

```
varDict = {'learning_rate':0.05, 'beta1':0.9, 'beta2':0.999, 'epsilon':1e-8, 'tol':1e-6, 'iter_TOL':1e4}
Learner = lambda dfda, a_init, f: myLearner.Adam(dfda, a_init, beta1=varDict['beta1'], beta2=varDict['beta2'],
                                                 epsilon=varDict['epsilon'], learning_rate=varDict['learning_rate'], f=f,
                                                 tol=varDict['tol'], iter_TOL=varDict['iter_TOL'], MSGOPENBOOL=True, MSGCount=10)
```

$learning\ rate = 0.05 \quad \beta_1 = 0.9 \quad \beta_2 = 0.999 \quad \epsilon = 10^{-8} \quad Tol. = 10^{-6} \quad Tol.Count = 10^4$

# Optimization Model-2

**Intrinsic emitting power:**

1. $\delta\ function: P_o = \sum_{j=1}^{m} a_j \delta(\lambda - \lambda_j)$

**Dipole orientation factors**

1. $\delta\ function$: $\eta_{1\sim5} = \sum_{j=1}^{m} a_j \delta(\lambda - \lambda_j)$
2. Cauchy function: $\eta_{1\sim5}(\lambda) = A_0 + \frac{A_1}{\lambda} + \frac{A_2}{\lambda^2}, \quad (\lambda: \mu m)$
3. Polynomial function: $\eta_{1\sim5}(\lambda) = A_0 + A_1\lambda + A_2\lambda^2, \quad (\lambda: \mu m)$

# Open DOFExtractor.py

OPEN FILES
× DOFExtractor.py

FOLDERS
- ori_src
  - Command
  - Execution
    - __pycache__
    - Example
    - DataMatrixAnalyzerScript.py
    - DeviceFieldCmd.py
    - DeviceOpticsCmd.py
    - DOFExtractor.py
    - FarFieldCmd.py
    - LazyCmd.py
    - materialMgrCmd.py
    - materialOpticsCmd.py
    - ModeDistributionAnalyzerCm
    - PowerDensityCmd.py
    - PurcellFactorCmd.py
    - rtauCmd.py
    - TextExecute.py
    - TRACmd.py
  - FunctionClass
  - Help
  - Material
  - MyColorScience
  - myOptimizor
  - Optics
  - OpticsCmd
  - ParameterExtractor
  - PlanarStructure
  - ReaderWriter
  - __init__.py
  - setup.py

DOFExtractor.py

```python
#!/usr/bin/env python3
# Copyright (c) 2019 Wei-Kai Lee/ Ting-An Lin. All rights reserved
# coding=utf-8
# -*- coding: utf8 -*-

### python module
import sys, os
import time
### my module
ExecPath = os.path.dirname(os.path.abspath(__file__))
srcPath = os.path.dirname(ExecPath)
if srcPath not in sys.path:
    sys.path.append(srcPath)
from ParameterExtractor.DOF_Spectrum_Extractor.FittorExecution import FittorExecution

######################################## Program Start ########################################

######################################## Setting Start ########################################
# data file information
readfilepath = '..\\ParameterExtractor\\DOF_Spectrum_Extractor\\Example\\S1_Qz_CBPIrppy2acac_20\\5 deg'
readfilename = 'fw1221'
# save file information
savefilepath = '..\\..\\Example\\DOFExtractor\\S1_Qz_CBPIrppy2acac_20'
savefilename = readfilename
# measurement setting information
thetaStart, thetaEnd, thetaSpace  = 0.0, 80.0, 5
wvStart, wvEnd, wvSpace  = 480, 650, 0.5
ModeList   = ['P']
# Device structure
materialList = ['sio2_nk', 'cbp_irppy2acac', 'air' ]
ThicknessList = ['X', 20, 'X']
EMLNo = 1 # calculate from 0
pos = ThicknessList[EMLNo]/2 # emission position
# Simulation Parameters
thetaSimStart, thetaSimEnd, thetaSimSpace  = 0, 90, 0.5
wvCenter = [500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 620, 640]
wvRange  = 2
######################################## Setting End ########################################
```

# Open DOFExtractor.py

Measured data file path and name

```
# data file information
readfilepath = '..\\ParameterExtractor\\DOF_Spectrum_Extractor\\Example\\S1_Qz_CBPIrppy2acac_20\\5 deg'
readfilename = 'fw1221'
```

$\theta$ and $\lambda$

```
# measurement setting information
thetaStart, thetaEnd, thetaSpace  = 0.0, 80.0, 5
wvStart, wvEnd, wvSpace  = 480, 650, 0.5
```

Structure

substrate EML

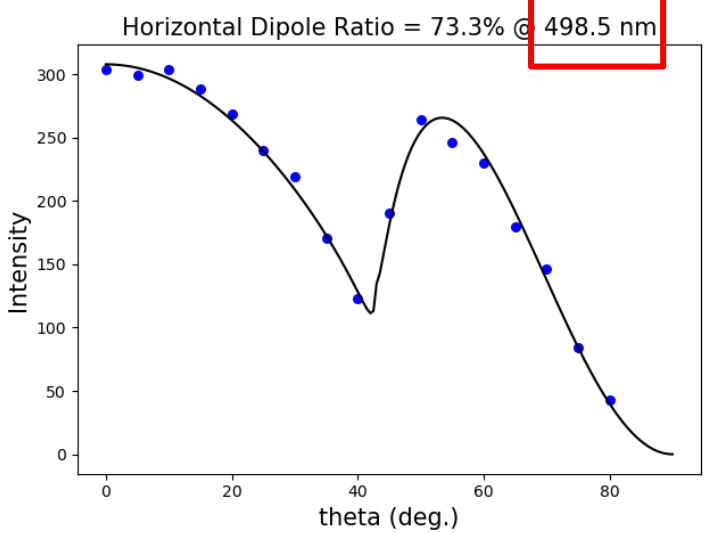
```
# Device structure
materialList = ['sio2_nk', 'cbp_irppy2acac', 'air' ]
ThicknessList = ['X', 20, 'X']
EMLNo = 1 # calculate from 0
```

# Open DOFExtractor.py

Simulation parameters

```
# Simulation Parameters
thetaSimStart, thetaSimEnd, thetaSimSpace  = 0, 90, 0.5
wvCenter = [500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 620, 640]
wvRange  = 2
```

Find the best fitting between $500 \pm 2nm$



# Fitting

```python
day = time.strftime("%Y-%m-%d", time.localtime())

savefilepath = os.path.join( readfilepath, day )

# point by point function fitting
FittingMode = 'PTBYPT' # PTBYPT / WAVELENGTHMODEL
DOFfunction = 'delta' # delta / cauchy / poly
FittorExecution(readfilepath, readfilename, savefilepath, savefilename, thetaStart, thetaEnd, thetaSpace,
                wvStart, wvEnd, wvSpace, ModeList, materialList, ThicknessList, EMLNo, pos, thetaSimStart, thetaSimEnd,
                thetaSimSpace, wvCenter, wvRange, FittingMode=FittingMode, DOFfunction=DOFfunction)

# wavelength model function fitting
FittingMode = 'WAVELENGTHMODEL' # PTBYPT / WAVELENGTHMODEL
DOFfunction = 'delta' # delta / cauchy / poly
FittorExecution(readfilepath, readfilename, savefilepath, savefilename, thetaStart, thetaEnd, thetaSpace,
                wvStart, wvEnd, wvSpace, ModeList, materialList, ThicknessList, EMLNo, pos, thetaSimStart, thetaSimEnd,
                thetaSimSpace, wvCenter, wvRange, FittingMode=FittingMode, DOFfunction=DOFfunction)

FittingMode = 'WAVELENGTHMODEL' # PTBYPT / WAVELENGTHMODEL
DOFfunction = 'cauchy' # delta / cauchy / poly
FittorExecution(readfilepath, readfilename, savefilepath, savefilename, thetaStart, thetaEnd, thetaSpace,
                wvStart, wvEnd, wvSpace, ModeList, materialList, ThicknessList, EMLNo, pos, thetaSimStart, thetaSimEnd,
                thetaSimSpace, wvCenter, wvRange, FittingMode=FittingMode, DOFfunction=DOFfunction)

FittingMode = 'WAVELENGTHMODEL' # PTBYPT / WAVELENGTHMODEL
DOFfunction = 'poly' # delta / cauchy / poly
FittorExecution(readfilepath, readfilename, savefilepath, savefilename, thetaStart, thetaEnd, thetaSpace,
                wvStart, wvEnd, wvSpace, ModeList, materialList, ThicknessList, EMLNo, pos, thetaSimStart, thetaSimEnd,
                thetaSimSpace, wvCenter, wvRange, FittingMode=FittingMode, DOFfunction=DOFfunction)
```

# How to execute

python: windows
python3: mac, linux

Execution file

```
D:\Dropbox\GoodLabSimulator_aniso\LegendDesign\ori_src\Execution>python DOFExtractor.py
************************************  GOODLAB SIMULATOR Info  ************************************
Optical Planar OLED Simulation Tool/Console interface
Anisotropic Version 1.0
Author : Wei-Kai Lee
Publication Date : 2019/03/15

Copyright(c) 2019 Wei-Kai Lee. All right reserved.
Dipole orientation factor extractor.
************************************  GOODLAB SIMULATOR Info  ************************************


Trial will been expired on  Sun May 10 00:00
>>> Please insert username : user-1
Now loading user setup : user-1 (../../SETT

Open log file...

Construct device...
```

Type user name

# Loading Material Manager System

```
Now loading user setup : user-1 (../../SETTING/user-1)

Open log file...

Construct device...

Now reading nk file (..\..\..\sim\material n k/3TPYMB.mat)
Now reading nk file (..\..\..\sim\material n k/3tpymb_shb.mat)
Now reading nk file (..\..\..\sim\material n k/B3PYMPM_isotropic.mat)
Now reading nk file (..\..\..\sim\material n k/B3PYMPM_uniaxial.mat)
Now reading nk file (..\..\..\sim\material n k/B3PYMPM_uniaxial.mat)
Now reading nk file (..\..\..\sim\material n k/air.mat)
Now reading nk file (..\..\..\sim\material n k/Al.mat)
Now reading nk file (..\..\..\sim\material n k/LiF.mat)
Now reading nk file (..\..\..\sim\material n k/CBP.mat)
Now reading nk file (..\..\..\sim\material n k/TAPC.mat)
Now reading nk file (..\..\..\sim\material n k/cito.mat)
Now reading nk file (..\..\..\sim\material n k/glass.mat)
Now reading spectrum file (..\..\..\sim\material PL/cbp_irppy3.spc)
Now reading spectrum file (..\..\..\sim\material PL/cbp_irppy2acac.spc)
Now reading dipole orientation factor file (..\..\..\sim\material eta/cbp_irppy3.eta)
Now reading dipole orientation factor file (..\..\..\sim\material eta/cbp_irppy2acac.eta)
Successfully reading materialMgr.mMgr
```

# Loading Material Manager System

```
Successfully reading materialMgr.mMgr
Now printing the information stored in the material manager...
[A]: er                  [N]3TPYMB(#2)
                         [N]B3PYMPM_isotropic(#1)
                         [N]B3PYMPM_uniaxial(#1)
                         [N]B3PYMPM(#1)
                         [N]air(#1)
                         [N]Al(#1)
                         [N]LiF(#1)
                         [N]CBP(#1)
                         [N]TAPC(#1)
                         [N]cito(#1)
                         [N]glass(#1)
-------------------------------------------------------------------------------
[A]: Fluorescence        [N]cbp_irppy3(#1)
                         [N]cbp_irppy2acac(#1)
-------------------------------------------------------------------------------
[A]: Phosphorescence     /*Empty*/
-------------------------------------------------------------------------------
[A]: DOF                 [N]cbp_irppy3(#1)
                         [N]cbp_irppy2acac(#1)
-------------------------------------------------------------------------------
[A]: wavelengthunitstr   [N]nm(#1)
                         [N]um(#1)
                         [N]m(#1)
-------------------------------------------------------------------------------
[A]: Attribute/ [N]: Name(# of data)



..\..\..\sim\material n k/
Now reading nk file (..\..\..\sim\material n k/sio2_nk.mat)
..\..\..\sim\material n k/
Now reading nk file (..\..\..\sim\material n k/cbp_irppy2acac.mat)

Construct Purcell Factor Object...
```

# Point by Point

Remove pulsed data

```
Read CCD Data...
Remove pulse data...
..\ParameterExtractor\DOF_Spectrum_Extractor\Example\S1_Qz_CBPIrppy2acac_20\5 deg\fw1221_5.txt : Number of removed data : 4
..\ParameterExtractor\DOF_Spectrum_Extractor\Example\S1_Qz_CBPIrppy2acac_20\5 deg\fw1221_8.txt : Number of removed data : 2

Construct data matrix...

Save processed measured data...

Dipole orientation factor (DOF) fitting...

Now preparing for optimization...
|Wavelength-650.0nm|(100%)>>>>>>>>>>>>>>>>>>>>>>>|Elapsed:0day(s)/0hr(s)/0min(s)/0sec(s)

==================================================
Loss average = 3.78673e-04
R2 average   = 0.9918292954413103
==================================================
```

Loss and $R^2$

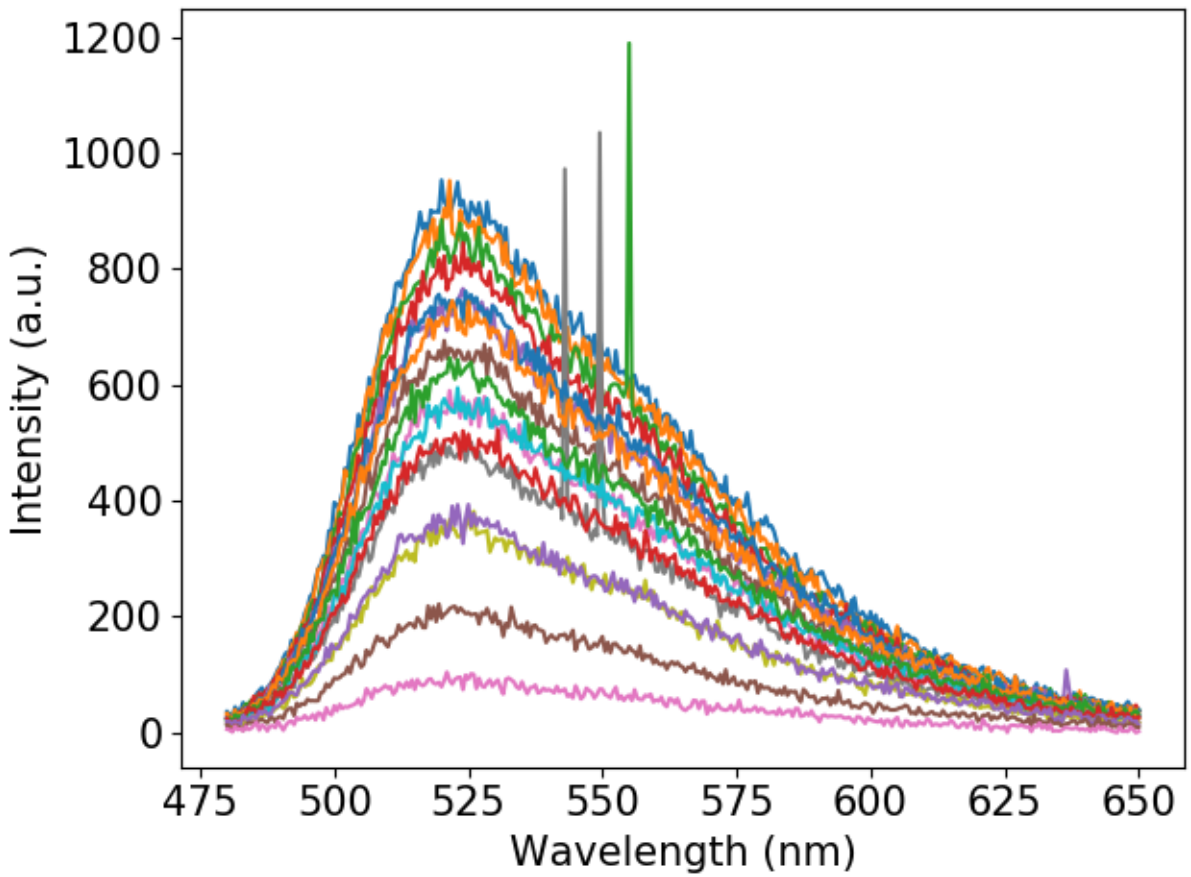
# Point by Point

```
Save fitted data...

C:\Users\User\AppData\Local\Programs\Python\Python37\lib\site-packages\numpy\core\_asarray.
  return array(a, dtype, copy=False, order=order)
==================================================
DOF1 (Mean/Std):  0.36/7.660e-03
DOF2 (Mean/Std):  0.00/0.000e+00
DOF3 (Mean/Std):  0.00/0.000e+00
DOF4 (Mean/Std):  0.00/0.000e+00
DOF5 (Mean/Std):  0.00/0.000e+00
==================================================

Calculate the simulaton data...

Loss/R^2 at 498.5 (nm) = 8.99028e-05/0.99847
Loss/R^2 at 508.0 (nm) = 1.02965e-04/0.99807
Loss/R^2 at 519.5 (nm) = 1.19140e-04/0.99773
Loss/R^2 at 528.0 (nm) = 1.06340e-04/0.99807
Loss/R^2 at 540.5 (nm) = 9.17769e-05/0.99830
Loss/R^2 at 548.0 (nm) = 1.11241e-04/0.99791
Loss/R^2 at 561.5 (nm) = 1.20330e-04/0.99764
Loss/R^2 at 569.0 (nm) = 1.06274e-04/0.99796
Loss/R^2 at 578.5 (nm) = 1.70230e-04/0.99676
Loss/R^2 at 589.5 (nm) = 2.35173e-04/0.99535
Loss/R^2 at 599.5 (nm) = 8.79090e-05/0.99846
Loss/R^2 at 619.0 (nm) = 3.13722e-04/0.99468
Loss/R^2 at 640.5 (nm) = 5.02533e-04/0.99149
where R^2 is the Pearson product-moment correlation coefficient
```

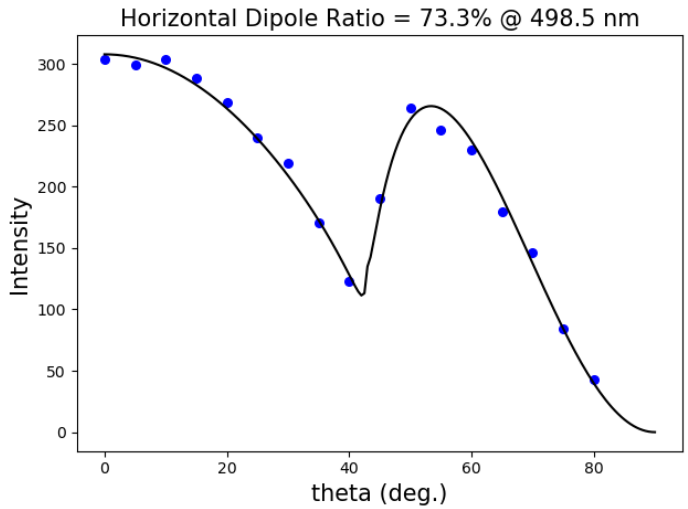
Average data

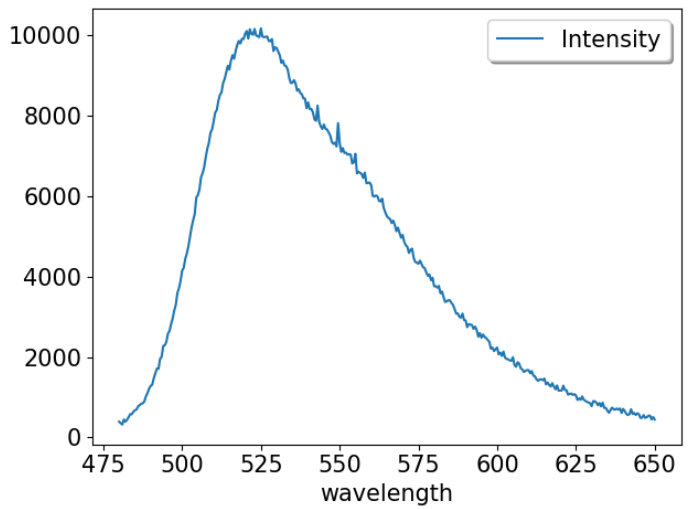
Selected wavelegth

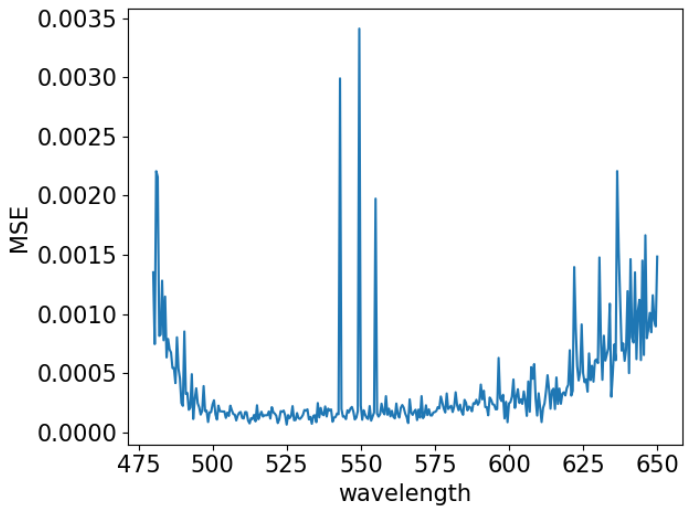
# Measured Data

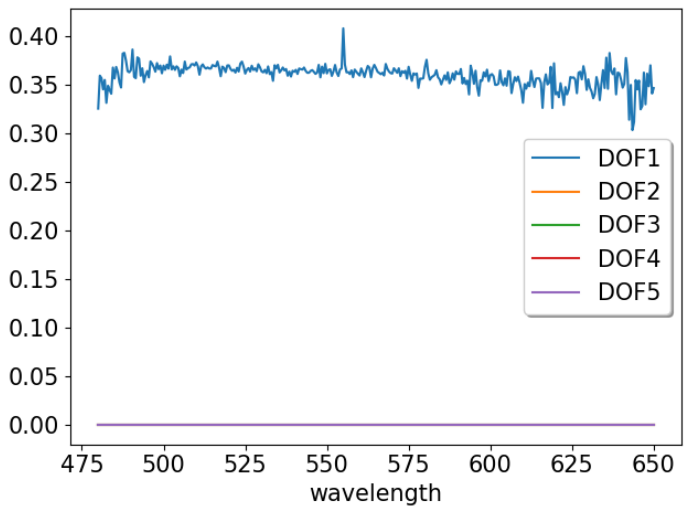


# Fitted Data


Horizontal Dipole Ratio = 73.3% @ 498.5 nm
Intensity
theta (deg.)
0
50
100
150
200
250
300
0
20
40
60
80


Intensity
wavelength
0
2000
4000
6000
8000
10000
475
500
525
550
575
600
625
650


MSE
wavelength
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
475
500
525
550
575
600
625
650


DOF1
DOF2
DOF3
DOF4
DOF5
wavelength
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
475
500
525
550
575
600
625
650