Title
Analysis of the Transition Dipole Moment Orientation from Nanoparticles

Author
LIN, TUNG TUNG

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Analysis of the Transition Dipole Moment Orientation from Nanoparticles

A dissertation submitted in partial satisfaction of the requirements for the degree

Master of Materials Science and Engineering

by

Tung Tung Lin

2021

ABSTRACT OF THE DISSERTATION

Analysis of the Transition Dipole Moment Orientation from Nanoparticles

by

Tung Tung Lin

Master of Materials Science and Engineering

University of California, Los Angeles, 2021

Professor Bruce Dunn, Co-Chair

Professor Carissa Eisler, Co-Chair

The transition dipole moment (TDM), a vector whose quantity describes the strength and the direction of the electronic transition between the emissive and ground state of an emissive material, determines the performance of all optoelectronic devices. Since the orientation of the TDM plays a major role in affecting directional optical properties, it is crucial to know accurate TDM orientation for creating OLEDs that efficiently outcouple light and luminescent concentrators that trap light into total internal reflection angles. Back focal plane (BFP) imaging is a commonly used method to reveal the orientation of the TDM. Because the emission pattern depends strongly on the TDM orientation, one can calculate the TDM from the BFP image. Despite how common this measurement has become for determining the TDM orientation, the error is often not reported even though many factors affect the accuracy of TDM angle-fitting results. In this thesis, the parameters in the TDM angle-fitting process, including theoretical data-

generating, emission pattern centering, background subtracting, BFP signal normalizing, and BFP signal fitting-range deciding, are discussed and to see how they affect the angle-fitting outcomes. Then methods that could ensure the accurate determination of the orientation of TDM are summarized.

Additionally, the limit of the accuracy and uncertainty of fitting TDM angles under different conditions are analyzed, including different refractive indexes and thicknesses of the emission layer, different numbers of dipoles, different positions where dipoles are, and different amounts of background noise of optical instruments. For a single dipole, such as dyes or nanoparticles, the uncertainty increases when the angle of the TDM becomes smaller. Besides, the maximum uncertainty appears when the single dipole is close to the dielectric interface; the minimum uncertainty happens when the dipole is right at the middle of the emission layer. For multiple dipoles, such as thin-film semiconductors, the uncertainty is at the same level as a single dipole in the middle of the emission layer, which means one dipole can represent all dipoles. Finally, a model which could generate mimetic experimental BFP emission patterns is introduced, so that the accuracy and uncertainty of fitting TDM angles under certain amounts of background noise can be predicted. With the insights provided by this work, designing high-efficient optoelectronic devices can be achieved.

The dissertation of Tung Tung Lin is approved.

Ximin He

Carissa Eisler, Committee Co-Chair

Bruce Dunn, Committee Co-Chair

University of California, Los Angeles

2021

Table of Contents

List of Figures:

List of Acronyms:

1. TDM: transition dipole moment
2. BFP: back focal plane
3. LSC: luminescence solar concentrator
4. OLED: organic light-emitting diode
5. EIA: Energy Information Administration
6. FF: fit factor

1. Introduction

1.1 Motivation - The transition dipole moment of materials determines the ultimate optoelectronic efficiency

In the past ten years, the directional optical properties of low dimensional structures such as nanoparticles, dyes, and 2D materials have emerged as an important tool to enhance the performance of optoelectronic devices. Due to electronic and dielectric confinement, properties such as light emission and exciton diffusion can be shaped. These properties are determined by the transition dipole moment (TDM): a vector whose quantity describes the strength and the direction of the electronic transition between the emissive and ground state of an emissive material [01]. Since the orientation of the photons from excitonic recombination is mostly perpendicular to the TDM [01], the direction of emitting light can be controlled by manipulating the orientation of TDM.

One of the main applications is organic light-emitting diodes (OLEDs). A horizontal orientation of TDM can enable the OLED external quantum efficiencies of over 30% because of stronger forward light emission [02]. (Figure 1) According to the Annual Energy Outlook 2021 estimated by U.S. Energy Information Administration (EIA), in 2020 the electricity used for lighting by U.S. residential and commercial sectors is about 219 billion kilowatt-hours (kWh) in total, which is about 8% of total electricity consumption [03]. By manipulating the TDM orientations from the emissive materials used in the lighting devices, the amount of electricity used, and therefore the amount of carbon-based emissions created, could be significantly reduced.

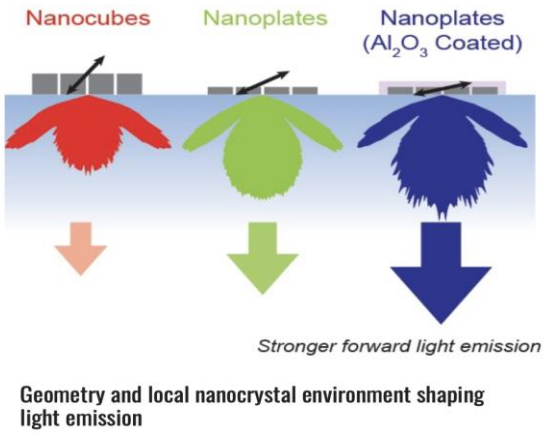


Figure 1 Light emissions from TDMs with different angles.

Another application is luminescent solar concentrators (LSCs). Figure 2 shows the relationship between the trapping efficiency of the dye in an LSC [05]. It is clear to see that when the TDM of the dye is perpendicular to the surface plane, the trapping efficiency of the LSC is higher than other angle distribution of dipoles because the light travels parallelly and can be trapped in the LSC. Therefore, to design high-efficiency optical devices, it is crucial to know the orientation of the TDM from materials accurately.

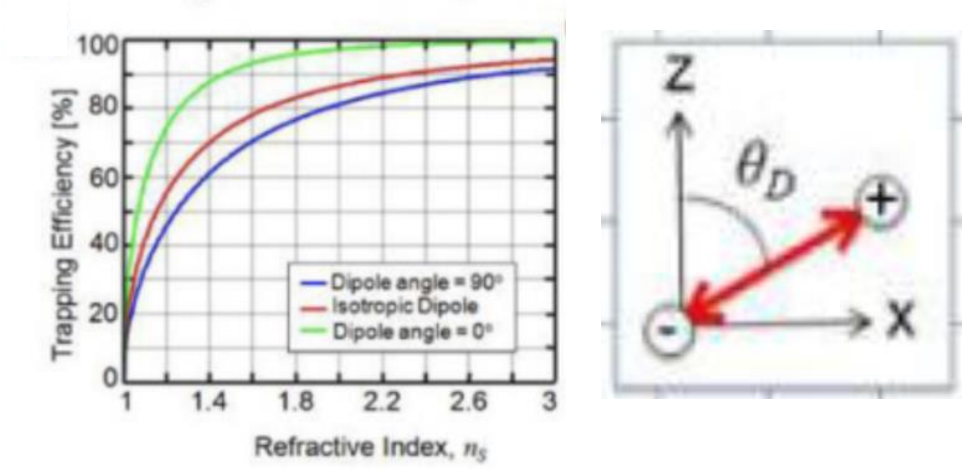


Figure 2 Left: The calculated trapping efficiency as a function of the refractive index of the dye medium. Right: Definition of TDM angle.

1.2 State of Art - What is currently used to determine the transition dipole moment?

Since the orientation of the TDM angle determines the direction of emitted light, the orientation of TDM can be determined by measuring the intensity of emitted light distribution. Nowadays, there are two main concepts to measure the orientation of TDM: measuring the intensity of emitted light distribution using a rotating substrate or measuring the emission pattern at the back focal plane (BFP) of a microscope objective.

1.2.1 Determining Angular Emission Through a Half-Cylinder Prism

As shown by Jurow et. al [01], the oriented light emission can be measured by mounting the sample to a half-cylinder prism (through spin coating or plasma-assisted atomic layer deposition) and then measuring the intensity of emitted light by rotating the substrate and directing the emission to a spectrometer. (Figure 3) [06]

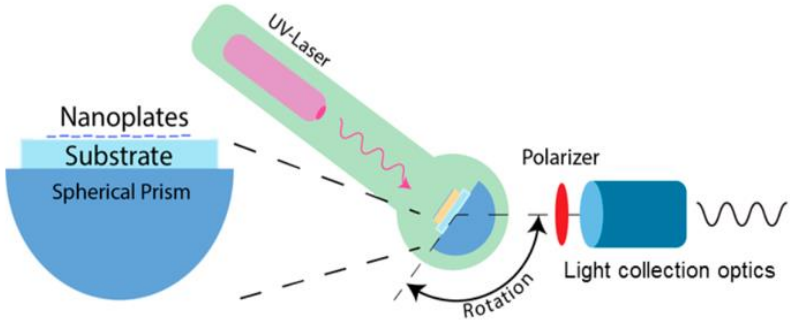


Figure 3 Measure angular dependent photoluminescence by mounting a film of perovskite nanoplates to a prism and analyzing the emitted light.

With this information, the alignment constant, ξ, can be calculated, then the signal can be changed into a function of angle according to Equation 1 [01].

Equation 1

$$\varphi = arcsin\left(\sqrt{\frac{n^4 \cdot \xi}{1+\xi(n^4-1)}}\right)$$

where φ is the angle of the TDM and n is the refractive index of the emitting medium.

ξ is bounded between 0 and 1: $\xi = 0$, when TDM perfectly aligns parallel on a subtract; $\xi = 1$, when TDM aligns vertically to a subtract.

1.2.2 Back Focal Plane Imaging

Since the previously discussed technique requires high precision in alignment and a longer time to integrate, which could cause sensitive samples to degrade, back focal plane (BFP) imaging is more commonly used for finding the orientation of TDM. BFP imaging, also known as Fourier plane imaging, can obtain angular emission patterns from emissive materials in a short amount of time (milliseconds to seconds). The typical optical microscope for BFP imaging includes a high numerical aperture objective lens, Bertrand lenses, tube lenses, flip mirrors, filters, a polarizer, and a charge-coupled device (CCD) camera. Usually, the microscope is inverted, which means the sample is on the top, and lenses and CCD camera are on the bottom to allow for an immersed objective setup. Figure 4 shows a typical setup for BFP imaging [07][06]. The sample is excited by a laser and emits photoluminescence, which is collected by the aperture. By focusing on this plane with the Bertrand lens, the angular emission pattern is projected onto the camera. Additionally, various filters are used to reduce background and laser signal and a polarizer can be used to filter out a specific part of the emission, allowing for the difference in TDM to be exaggerated.

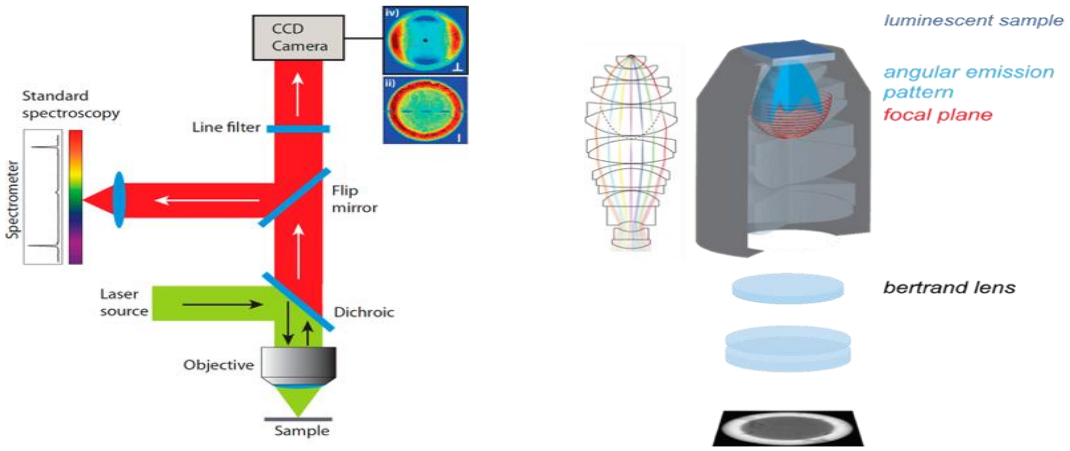


Figure 4 Illustration of BFP measurement setups.

There are some possible sources of error in this system that affect the accuracy of determining the intensity of emission as a function of angle. First, the placement of the relay lenses can make alignment more difficult. Slight misalignment of lenses can create an inaccurate BFP image. Secondly, spherical aberration is also an issue, and it comes from the lens field acting inhomogeneously for off-axis rays. The further off-axis the beam of light is, the closer the light focuses on the lens. Finally, the choice of the objective lens has a strong impact on the uncertainty of the measurement. Since the range of the BFP that can be collected is related to the size of the numerical aperture angle of the objective lens, the bigger the numerical aperture angle is, the bigger range of the BFP imaging can be collected. However, the bigger the aperture angle is, the worse the aberration becomes. Equation 2 shows the relationship between the aperture angle and the size of the disk of the least confusion of spherical aberration.

Equation 2

$$d_s = 2C_s\alpha^3$$

Therefore, it is crucial to optimize the optic system to get accurate BFP imaging.

In BFP imaging, each of the points on a pattern represents a unique angle of emission, and each angle has a unique pattern. (Figure 5) Therefore, the orientation of TDM can be determined by emission patterns on the BFP.

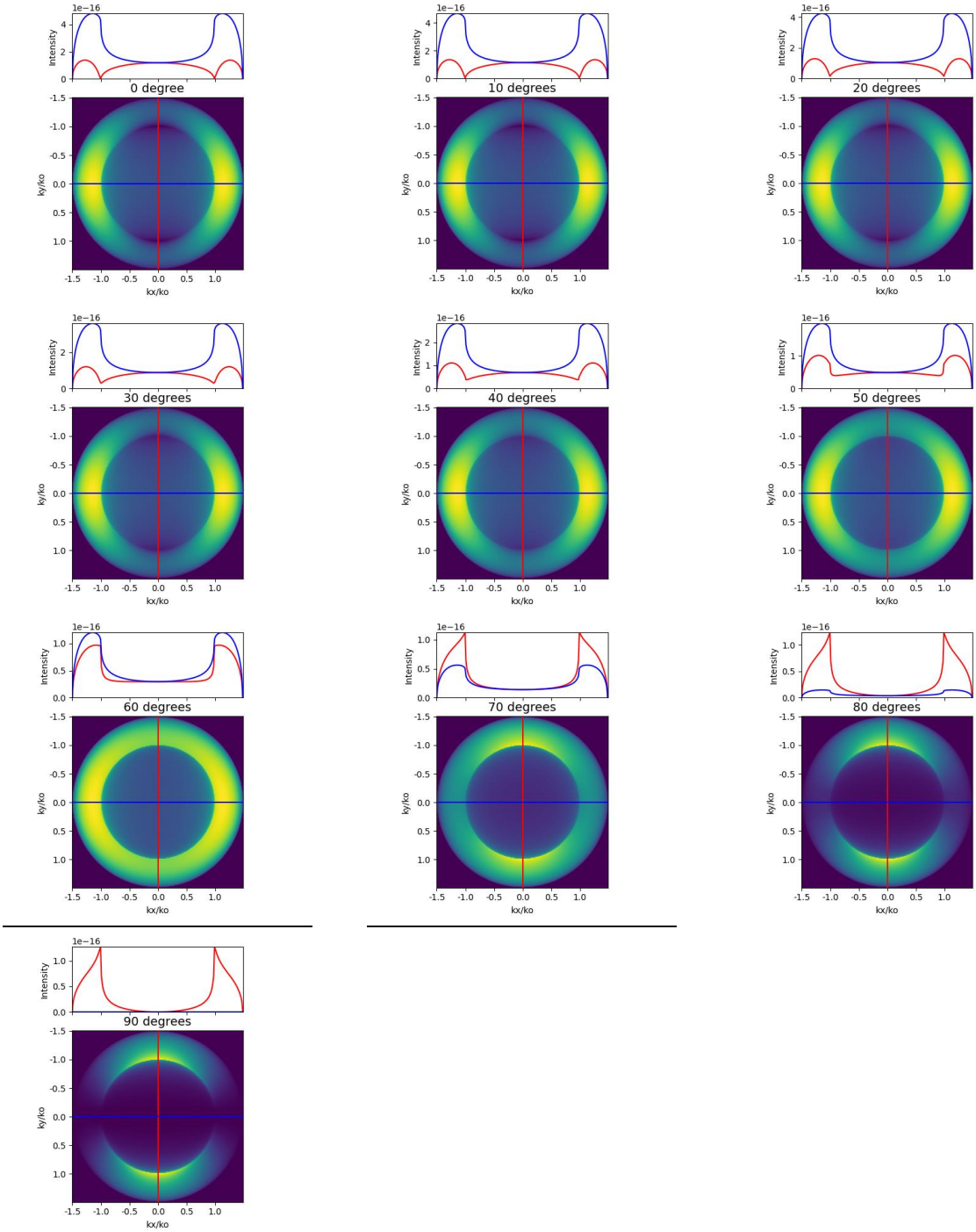


Figure 5 BFP images and signal intensity cross-sections from TDMs with different angles. n_1=1, n_2=2, n_3=1.5, D=10nm, λ=500nm, size=200X200.

The center of this angular emission pattern represents normal emission, which means the direction parallel to the optical axis, and the outer edge represents the maximum collected angle, which is determined by the numerical aperture of the objective lens.

1.3 The Accuracy and Uncertainty of Fitting TDM Angles

Despite these uncertainties, this technique is well-developed and various research has been done to successfully study many systems. Lieb et al. used this technique to determine single-molecular orientations, Taminiau et al. used it to quantify the natural magnetic dipole transitions in lanthanide ions, Schuller et al. used it to reveal the orientation of luminescent excitons in layered materials (MoS_2 and PTCDA), Gao et al. used it to resolve the dipole orientation in CdSe nanoplatelet [08][09][10][11]. Although this research shows great examples of applying BFP imaging to determine the orientations of dipole moment, **there has been little investigation into the accuracy and inherent error of determining TDM angle from this measurement.**

In this thesis, I determine the accuracy and precision of BFP imaging by asking the following questions:

- How does the imaging equipment of the BFP setup affect the TDM angle-fitting results?
- How does the local environment where the dipoles are (such as reflective index and thickness of the emission layer, background noise) affect the angle-fitting results?
- What is the range of fitting TDM angle's uncertainty under different conditions?

These questions lead to a valuable standard to calculate orientations of dipole moment from BFP imaging with high accuracy. With precise knowledge of the TDM angle, extremely efficient optoelectronic devices can be achieved by manipulating dipoles in emission layers.

2. Theory

2.1 The Origin of Transition Dipole Moment

When a material absorbs light whose energy is high enough to knock out orbital electrons, electrons will travel to higher energy states. When an electron relaxes back down to the ground state, a photon can be released as part of excitonic recombination. The transition dipole moment (TDM) represents a vector quantity that describes the polarization of electronic oscillation during this process. The emitted light has a cosine angular distribution whose peak is oriented perpendicular to the TDM. Therefore, most of the light is emitted perpendicularly to the TDM. In a uniform optical environment, if TDMs are aligned, photons are emitted anisotropically. However, if TDMs are oriented randomly, the ensemble average will exhibit equal components in (x, y, z), and the light emission is isotropic [06].

2.2 Polarization of Light

As mentioned in the previous section of the BFP imaging measurement setup, polarizers are included in a typical optical microscope for BFP imaging. Polarizers can filter light waves with specific polarization and block light waves of other polarizations, therefore, it is easier to determine the TDM from BFP through a polarizer. Figure 6 shows BFP images with and without a polarizer [04]. With a polarizer, the patterns' difference between parallel and vertical excitation can be told easily.

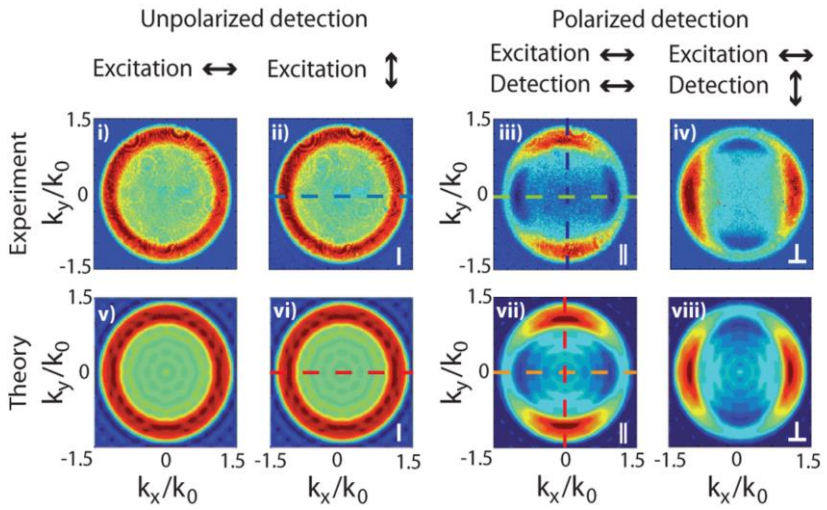


Figure 6 Experimental and calculated BFP images with and without a polarizer.

Polarization of light can be classified into s- or p-polarized light. P-polarized (from the German parallel) light has an electric field polarized parallel to the plane of incidence but perpendicular to the plane of incidence, while s-polarized (from the German senkrecht) light is perpendicular to the plane of incidence but parallel to the incident surface. Figure 7 shows a system with s- and p-polarized light [12].

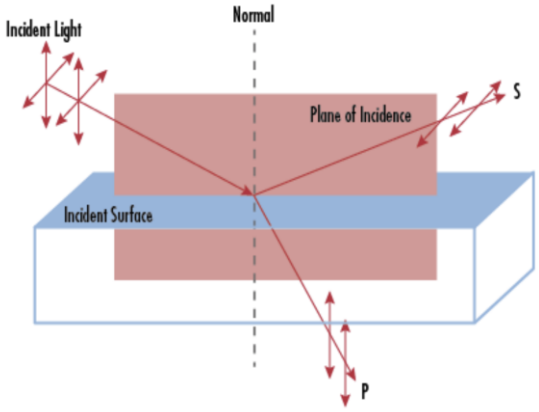


Figure 7 P and S are linear polarizations defined by their relative orientation to the plane of incidence.

3. Model

3.1 Model of Back Focal Plane Imaging

To determine the TDM of a sample, we use a simple model (Figure 8) to describe the local optical environment [10]. The system discussed here contains three layers, from the top to the bottom are air, thin film (emission layer), and the substrate, respectively.

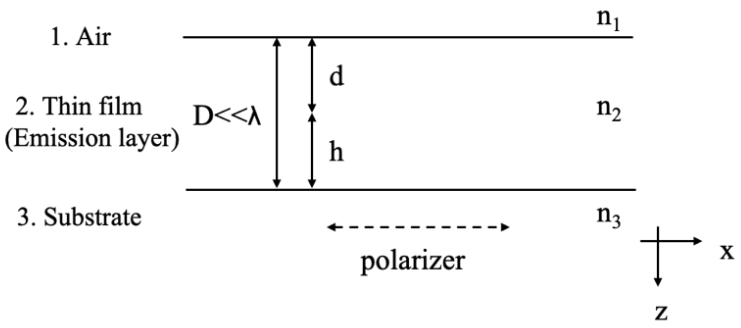


Figure 8 Illustration of the three-layer system considered theoretically.

The emission projected on the imaging plane at the direction of (k_x, k_y) with a polarizer is aligned in the direction of X is calculated as [06], [10], [11]:

Equation 3

$$N^{POL}(k_x, k_y) = C((\rho_x^s + \rho_x^p) \cdot \frac{1 - sin^2(TDM)}{2} + \rho_z^p \cdot sin^2(TDM))$$

where N is the calculated intensity of emission, ρ is the local density of optical state (LDOS), and TDM is the angle between the transition dipole moment and the surface (Figure 9). C is a constant that is related to experimental conditions, such as excitation intensity and integration time. It only affects the intensity but not the energy-momentum distribution. [09]

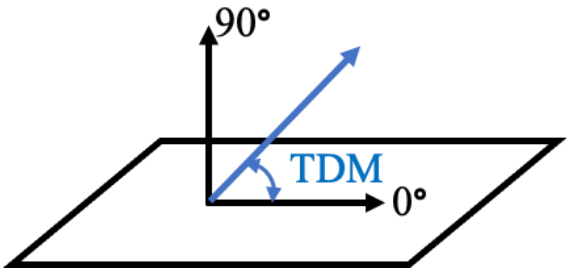


Figure 9 Definition of the TDM angle used in the three-layer system.

The LDOS for an emitter in this system is described by the following equations:

Equation 4

$$\rho_x^s = \left(\frac{1}{8\pi k_0^2}\right) \cdot \left(\frac{k_0}{k_{z3}}\right) \cdot \left| \frac{t_{32}^s \cdot e^{i \cdot k_{z2}^s \cdot \frac{D}{2}} \cdot (1 + r_{21}^s \cdot e^{2i \cdot k_{z2}^s \cdot \frac{D}{2}}) \cdot k_y}{(1 - r_{21}^s \cdot r_{23}^s \cdot e^{2i \cdot k_{z2}^s \cdot D}) \cdot \sqrt{k_x^2 + k_y^2}} \right|^2$$

Equation 5

$$\rho_x^p = \left(\frac{1}{8\pi k_0^2}\right) \cdot \left(\frac{k_0}{k_{z3}}\right) \cdot \left| \frac{t_{32}^p \cdot e^{i \cdot k_{z2}^p \cdot \frac{D}{2}} \cdot \frac{k_{z2}^p}{n_2 \cdot k_0} (1 - r_{21}^p \cdot e^{2i \cdot k_{z2}^p \cdot \frac{D}{2}}) \cdot k_x}{(1 - r_{21}^p \cdot r_{23}^p \cdot e^{2i \cdot k_{z2}^p \cdot D}) \cdot \sqrt{k_x^2 + k_y^2}} \right|^2$$

Equation 6

$$\rho_z^p = \left(\frac{1}{8\pi k_0^2}\right) \cdot \left(\frac{k_0}{k_{z3}}\right) \cdot \left| \frac{t_{32}^p \cdot e^{i \cdot k_{z2}^p \cdot \frac{D}{2}} \cdot \frac{k_x}{n_2 \cdot k_0} (1 + r_{21}^p \cdot e^{2i \cdot k_{z2}^p \cdot \frac{D}{2}})}{1 - r_{21}^p \cdot r_{23}^p \cdot e^{2i \cdot k_{z2}^p \cdot D}} \right|^2$$

where:

Equation 7

$$k_{zi} = \sqrt{n_i^2 \cdot k_0^2 - \left(\sqrt{k_x^2 + k_y^2}\right)^2}$$

Equation 8

$$k_0 = \frac{2\pi}{\lambda}$$

where k_{zi} is the momentum along z in layer i, k_0 is the total momentum, λ is the wavelength. The reflection and transmission of a plane wave between interfaces i and j are described by the Fresnel reflection r and transmission t coefficient:

Equation 9

$$t_{ij}^{p} = \frac{2n_i \cdot n_j \cdot k_{zi}}{n_j^2 \cdot k_{zi} + n_i^2 \cdot k_{zj}}$$

Equation 10

$$t_{ij}^{s} = \frac{2k_{zi}}{k_{zi} + k_{zj}}$$

Equation 11

$$r_{ij}^{p} = \frac{n_j^2 \cdot k_{zi} - n_i^2 \cdot k_{zj}}{n_j^2 \cdot k_{zi} + n_i^2 \cdot k_{zj}}$$

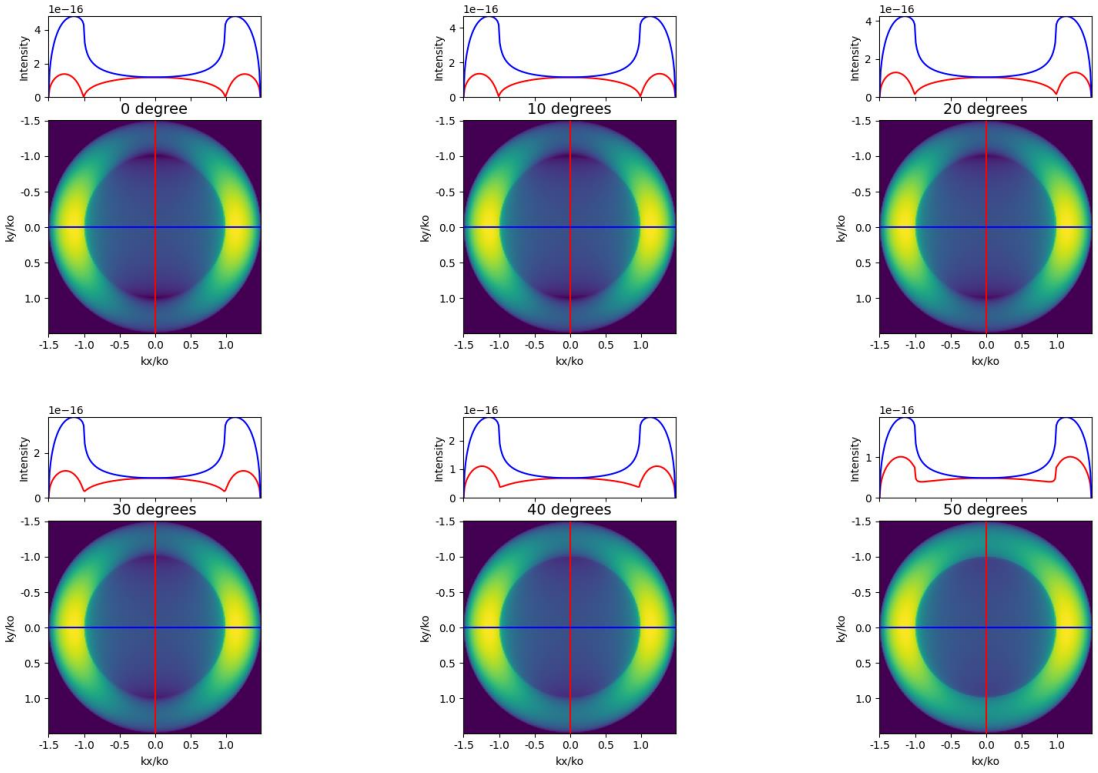
Equation 12

$$r_{ij}^{s} = \frac{k_{zi} - k_{zj}}{k_{zi} + k_{zj}}$$

where n_i is the refractive index in the ith layer, the symbol s, and p denote the polarization of the electromagnetic wave with respect to the plane of incidence, t_{ij}^{p} is the p-polarized transmission coefficient from layer i into layer j, t_{ij}^{s} is the s-polarized transmission coefficient from layer i

into layer j, r_{ij}^{p} is the p-polarized reflection coefficient from layer i into layer j, r_{ij}^{s} is the s-polarized transmission coefficient from layer i into layer j.

As stated earlier, the TDM plays a major role in the angular emission of a sample. As shown in Figure 5, the emission patterns change with different TDM angles. The intensity change can be shown obviously by the cross-section of the emission patterns in k_x and k_y directions (Figure 10). It is clear to see that the dips and peaks of the cross-section change with the TDM angle, especially in k_x direction. Therefore, during the TDM angle-fitting process, the comparison of these dips and peaks parts is crucial for accurate results. Additionally, it is worth noting that the changes between dips and peaks at smaller angles are very small, which will lead to a higher uncertainty of the angle-fitting result for smaller angles.



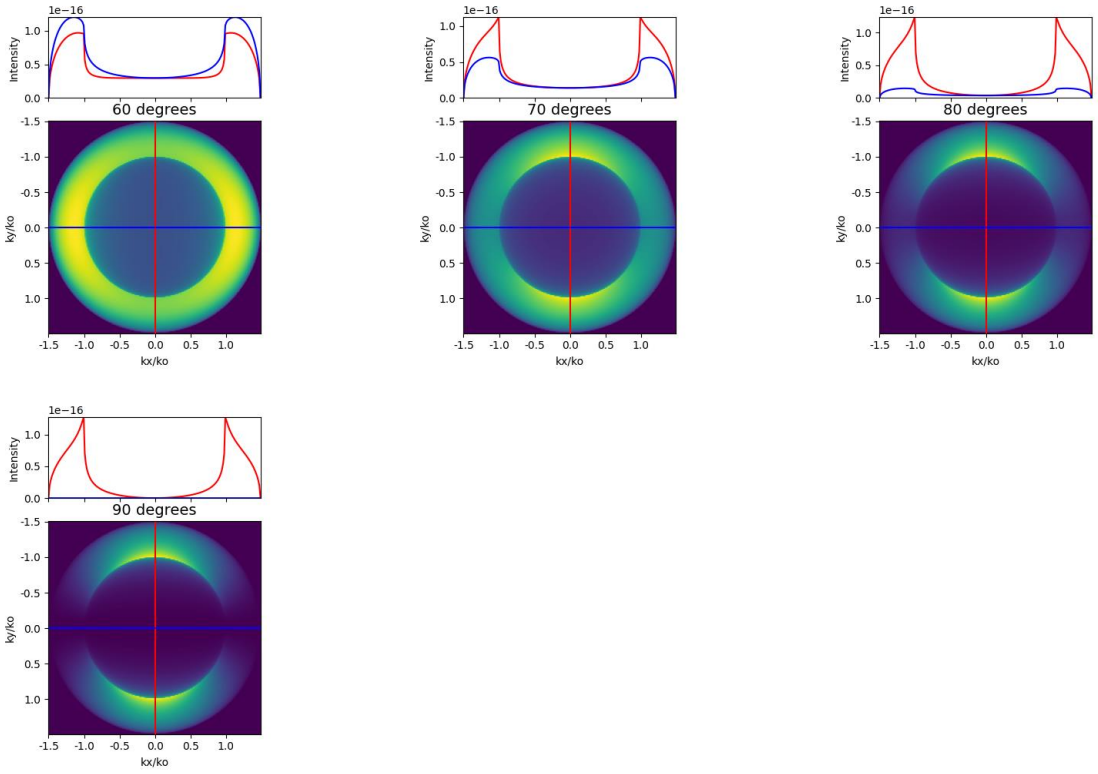


Figure 5 BFP images and signal intensity cross-sections from TDMs with different angles.

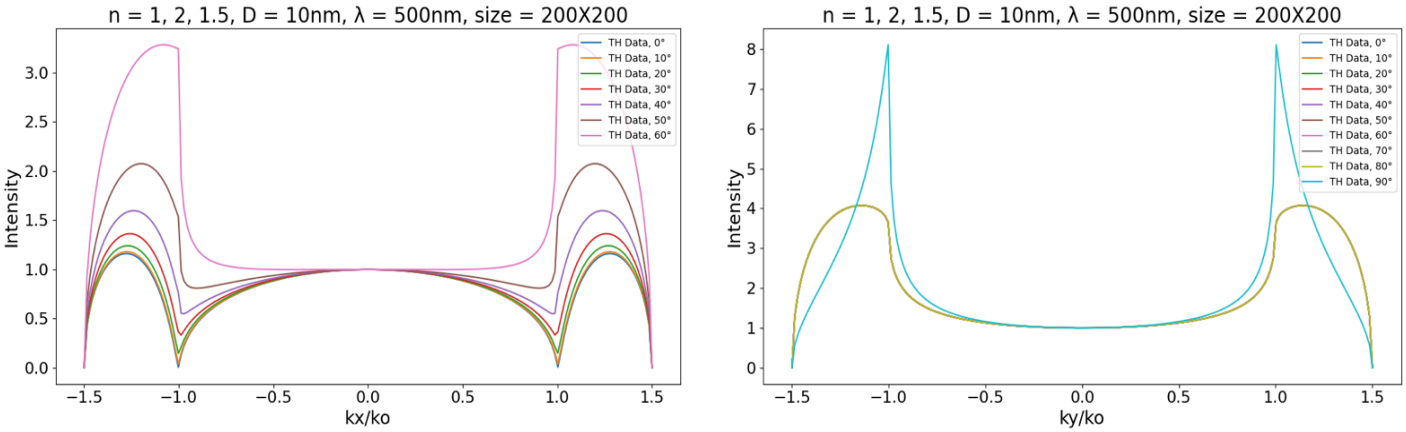


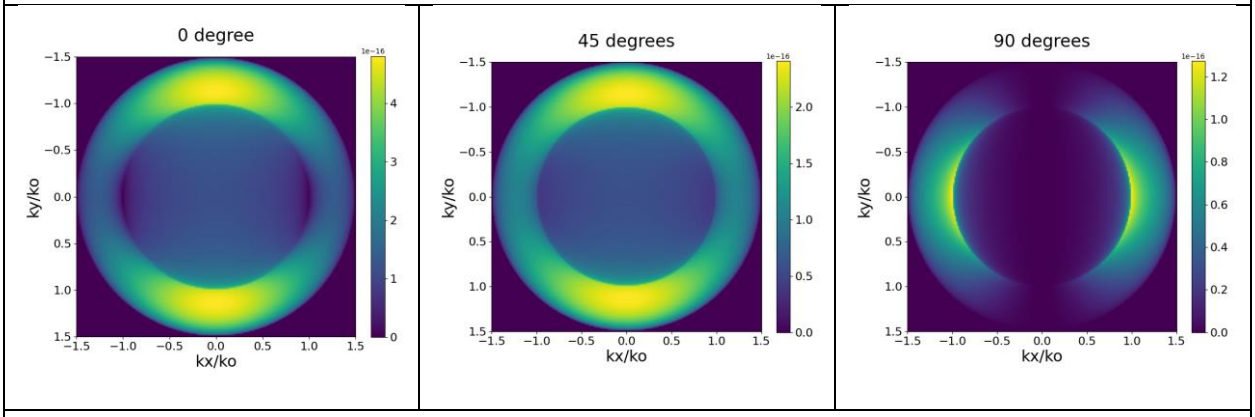
Figure 10 kx-, ky- cross-sections from TDM with different angles. n_1, n_2, n_3 are 1, 2, 1.5, respectively.

Additionally, one must consider the range of k_x and k_y during the TDM angle-fitting process. Here, the k_x and k_y of simulation are bounded between -1.5 to 1.5, which is equal to the refractive index of the substrate. However, for experimental emission patterns, the absolute

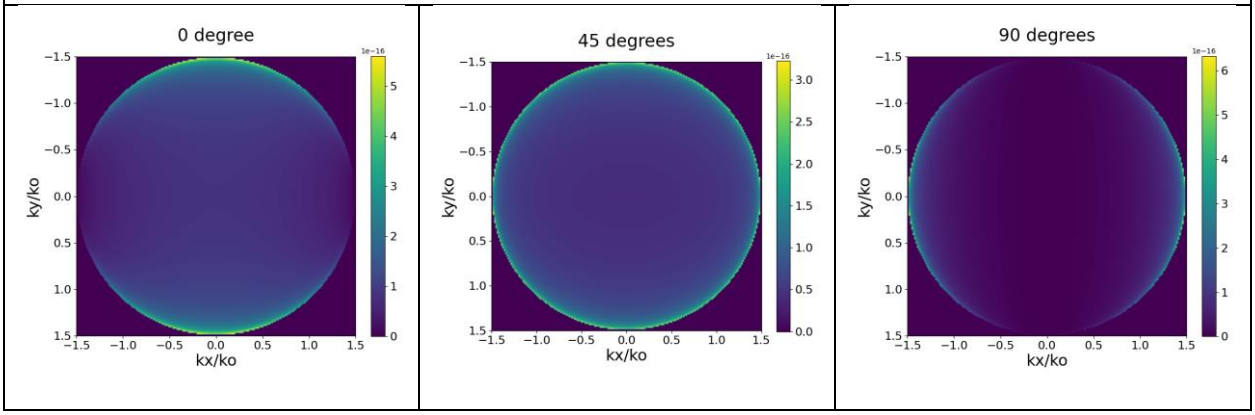
values of k_x and k_y are usually smaller than 1.5 since numerical apertures used in optical systems cannot collect light from all angles in reality.

It is also worth mentioning that the emission pattern in BFP is not only affected by the orientation of TDM but also affected by the local environment where dipoles are because the refractive index of the environment decides how light travels.

n1 = 1 (air), n2 = 2, n3 = 1.5 (glass), D=10nm, λ=500



n1 = 1.5, n2 = 2, n3 = 1.5, D=10nm, λ=500



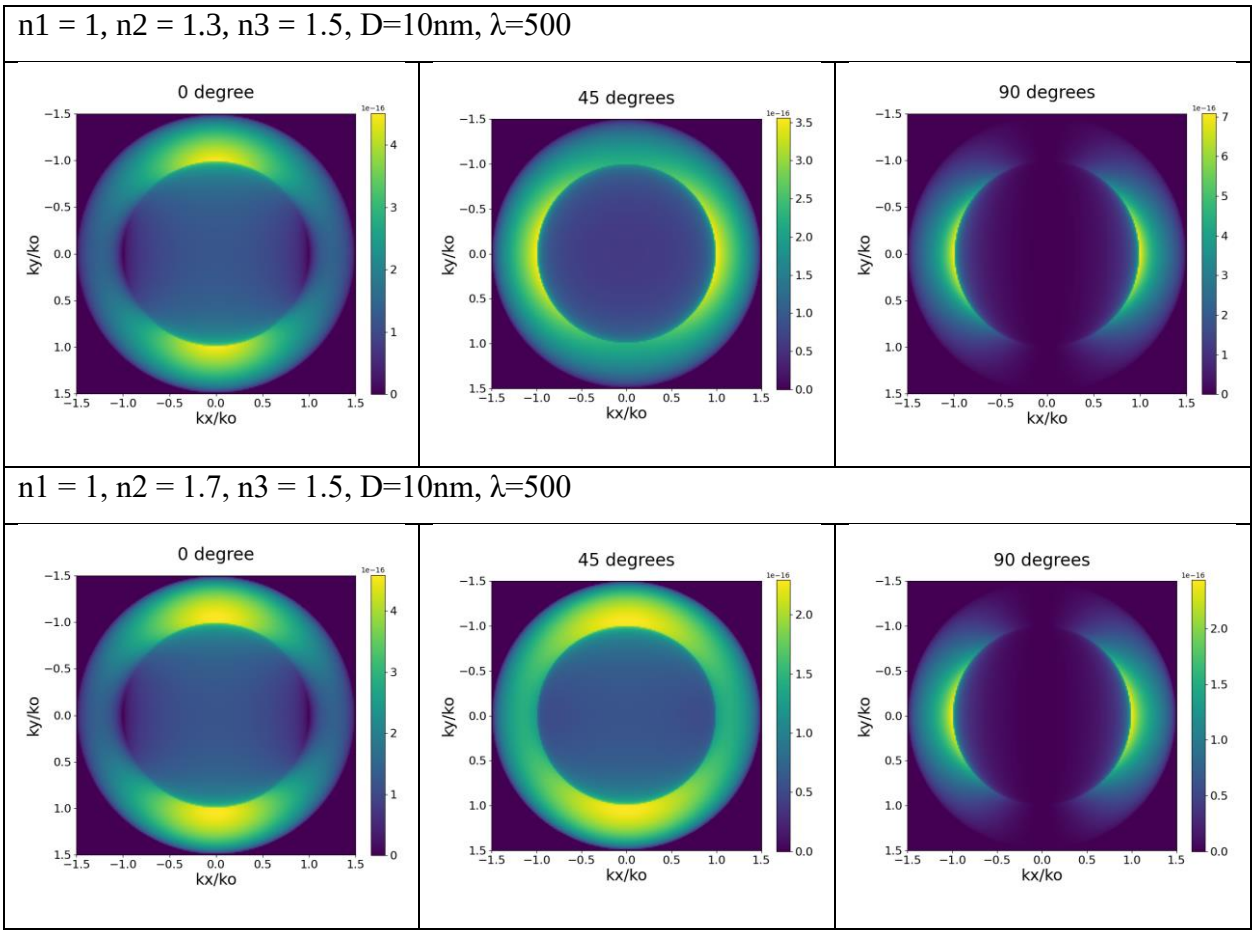


Figure 11 Emission patterns from TDM within different environments. n_1, n_2, n_3 are refractive indexes of the top layer, emission layer, and substrate.

3.2 Best-Fit Angle Calculation

The TMD angle of experimental data is calculated by χ^2, which is the sum of differences between the observed (adjusted by fit factor) and expected values squared and divided by the expected values (Equation 13). The best-fit occurs at χ^2_{min}. The fit factor is used for minimizing the effects from background noise since experimental conditions could bring fluctuation to the data, further discussion is in Chapter 4.4.

Equation 13

$$\chi^2 = \frac{(theoretical\ data\ -\ experimental\ data\ \cdot\ fit\ factor)^2}{theoretical\ data}$$

The error can be calculated by the following equation with α, which determines the confidence regions. When $\alpha\ =\ 1$, the confidence region is 68% (one sigma).

Equation 14

$$\chi^2(TDM\ angle\ uncertainty)\ -\ \chi^2_{min}\ =\ \alpha$$

The following example shows the process of TDM angle-fitting based on the discussion above.

A. Normalize both experimental and theoretical emission patterns (The method for normalization is discussed in Chapter 4.4)

B. Calculate the χ^2 by Equation 13 (use the data points within k_x and k_y = +-1.4)

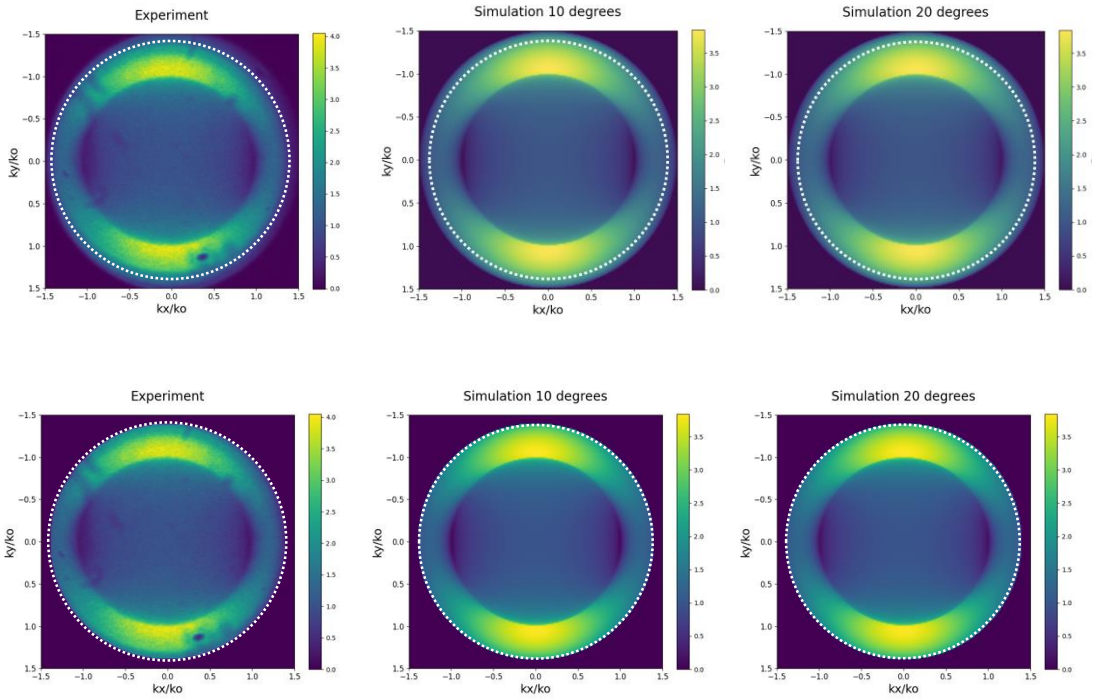


Figure 12 Emission patterns of experimental data and calculated data. (the radius of the white dash-circle is 1.4)

C. Calculate the TDM best-fit angle error by Equation 14

(best fitting TDM angle is 17°, the upper-bound angle is 18.3°, the lower-bound angle is 15.6°)

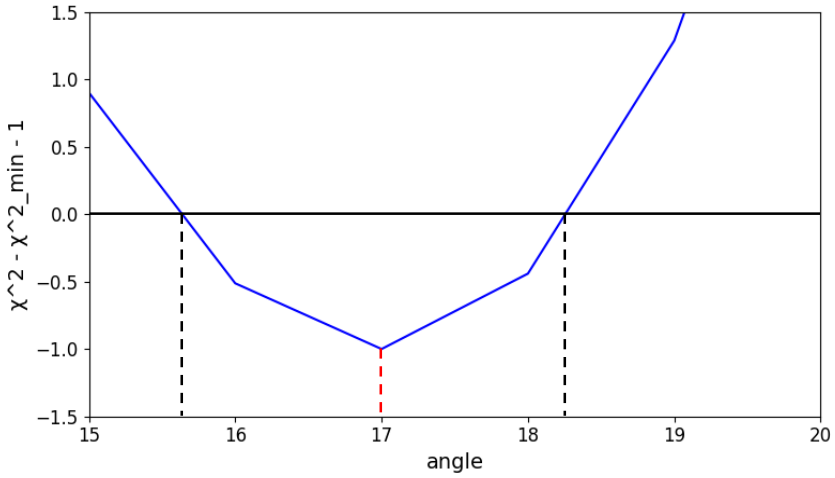


Figure 13 Best-fitting TDM angle calculated by Equation 13 and 14.

4. Effects of Parameters and Processes on Angle Analysis of TDM

In this chapter, the process of fitting simulated BFP images discussed. This includes theoretical data-generating, emission pattern centering, background subtracting, BFP signal normalizing, and BFP signal fitting-range deciding. Each step in the process has a significant impact on the TDM angle-fitting accuracy, **therefore, it is crucial to understand how to conduct the TDM angle analysis process correctly and avoid any contribution to the uncertainty**.

4.1 Theoretical Data Generating

According to the model discussed in Chapter 3, emission patterns of TDM with different angles can be calculated (Python script of the model is in Appendices A.1.1). The size of an emission pattern is determined by the number of pixels/data points used in the calculation. This corresponds to a combination of the resolution of the camera and the magnification of the lenses. In the theoretical data generating process, it is important to calculate simulation data directly from the equations of the model, instead of resizing the data by interpolation as it loses important data and increases uncertainties of TDM angle analysis.

Figure 14 gives an example of this situation. If we calculate a BFP simulation emission pattern that has 1000X1000 data points and then interpolate the data points into 184X184 (green line), the best-fit angle and the uncertainty of the TDM are different than directly calculating an emission pattern with size 184X184 from the model (red line). This is because the interpolated data set will lose or inaccurately show the important fine features (for example, the dips near kx = 1 in Figure 15). Therefore, to achieve accurate TDM angle analysis, we need to make sure to use the right way to generate BFP simulation signals correctly at the very first step.

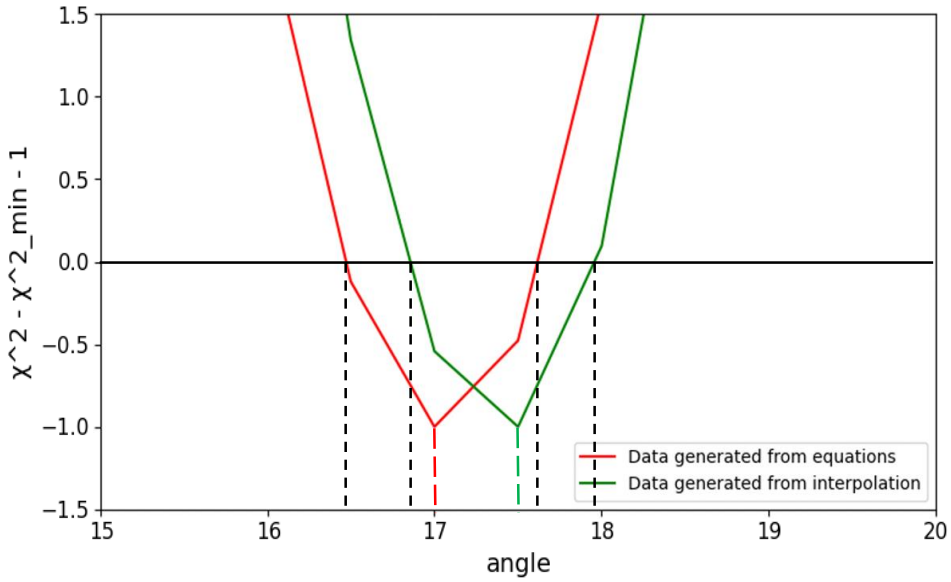


Figure 14 Best-fitting TDM angle calculated by Equation 13 and 14. For data generated from equations: best-fit angle is 17°, lower-bound angle is 16.5°, upper-bound angle is 17.6°; for data generated from interpolation: best-fit angle is 17.5°, lower-bound angle is 16.8°, upper-bound angle is 17.95°.

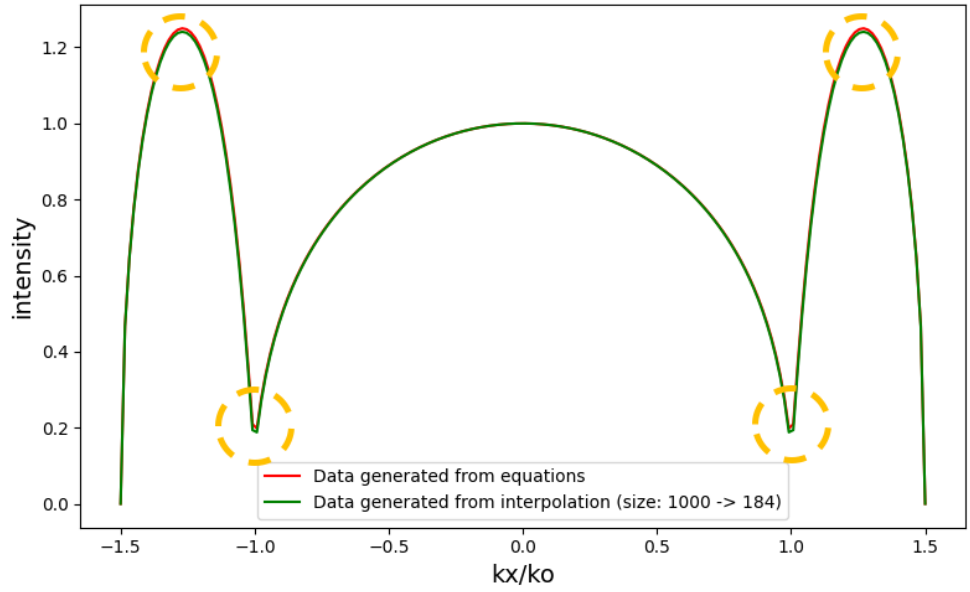


Figure 15 kx cross-sections from the model (directly calculated by equations) and interpolation data. The orange circles mark the slight differences at peaks and dips.

4.2 Data Centering - How to extract the BFP signal from the camera?

To calculate the best-fit angle by comparing each single data point from both theoretical and experimental emission patterns, we must precisely determine the center of the BFP signal, where $(k_x, k_y) = (0, 0)$. Since the intensity of the emission pattern is much higher than the background, this property can be used to find the center of the emission pattern. Figure 16 shows the data centering process. First, we calculate the sum of the intensity of every single data point in the direction of k_x (blue line) and k_y (red line). Second, we calculate the difference between adjacent rows and columns. The edge of the emission is at the point when the difference changes dramatically. Through the edge positions of the emission pattern, the center of the pattern is known.

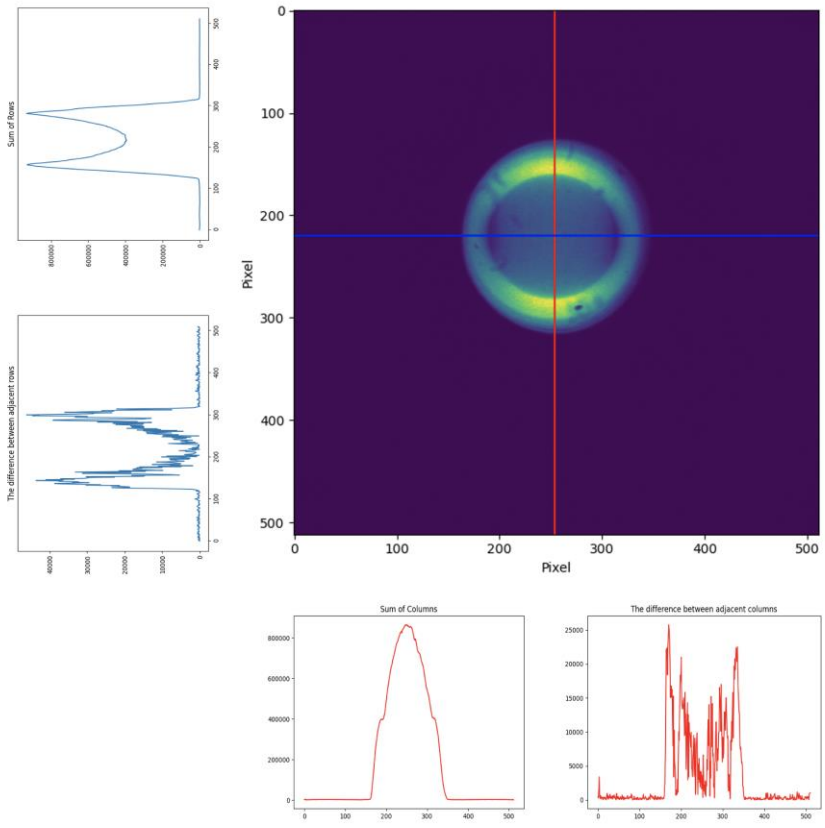


Figure 16 Illustration of the data-centering process.

Figure 17 shows the cut data, demonstrating that the BFP signal is correctly centered.

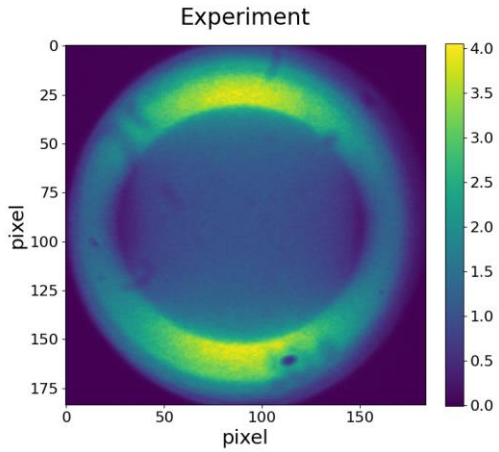


Figure 17 BFP emission pattern of the cut data.

The crucial step in this process is to make sure that the theoretical and experimental patterns overlap well, especially the exaggerated dips and peaks often seen in these datasets. Figure 18 shows the kx cross-section of the theoretical and experimental emission patterns.

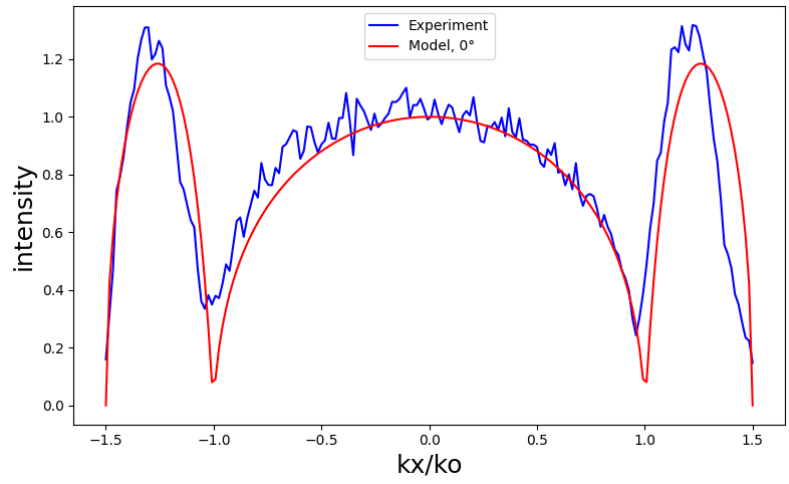


Figure 18 Comparison of kx cross-sections from experimental data and calculated data.

4.3 Background Subtraction

Due to experimental conditions, there is background noise in the BFP imaging. Most of the noise is inherent to electronic image sensors such as the CCD camera. A CCD camera is a metal oxide chip sensor that transports electrically charged signals into a light image through the photoelectric effect [13]. Common noises from CCD cameras are photon noise, dark noise, and read noise. Photon noise is related to the photon arrival rate to the CCD; dark noise is related to the number of electrons thermally generated within the silicon structure of the CCD; read noise is related to the process of converting CCD charge carriers into voltage signal [14]. To get an accurate best-fit angle, background subtraction is needed for experimental raw data because the angle fit depends on the signal change from 0.

The first step is to isolate the emission pattern. Through the data centering process, the edges of the BFP emission pattern are known. According to this, we can remove the data points in this area, which shows in Figure 19. Second, we estimate what the background contribution to the BFP image will be based on the surrounding pixels. I have researched several ways for estimating the background: linear of columns, average of rows, average of columns, griddata linear, griddata nearest, and griddata cubic. These methods are summarized as follows:

- Linear of columns: predict the background contribution to the BFP signal with the linear interpolated values calculated from columns
- Average of rows: predict the background contribution to the BFP signal with the average of rows' value
- Average of columns: predict the background contribution to the BFP signal with the average of columns' value

- Griddata linear: predict the background contribution to the BFP signal with the values getting from the linearly interpolated value on each simplex (the simplex is tessellated from the input point)
- Griddata nearest: predict the background contribution to the BFP signal with the value at the data point closest to the point of interpolation
- Griddata cubic: predict the background contribution to the BFP signal with the value determined from a piecewise cubic, continuously differentiable, and approximately curvature-minimizing polynomial surface

I have determined two criteria to determine the efficacy of each method. First, after background subtraction, the intensity outside the emission pattern should be as close to zero as possible (i.e. the background should be 0). Second, the shape of the cross-section of the emission pattern should keep the same before and after background subtraction, so the background subtraction does not artificially change the BFP image.

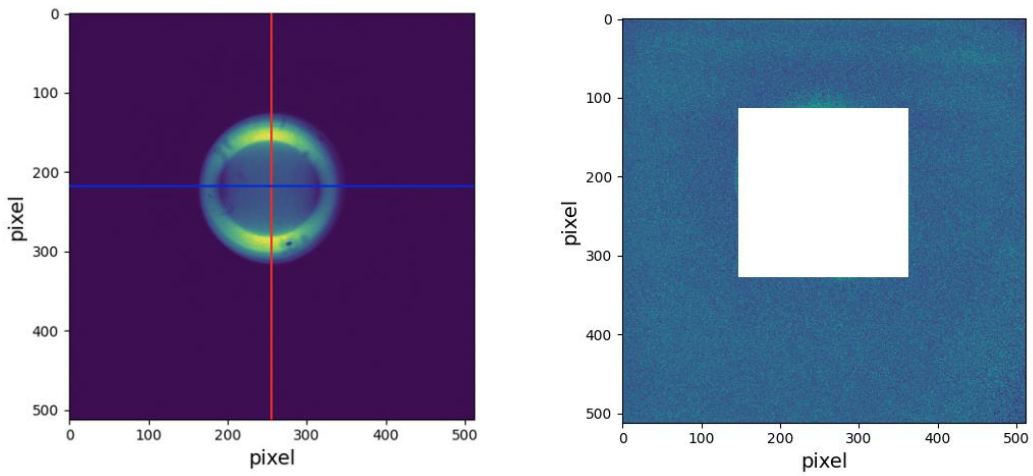


Figure 19 Illustration of isolating the emission pattern from the background.

Figure 20 shows the estimated background calculated by the six methods discussed above.

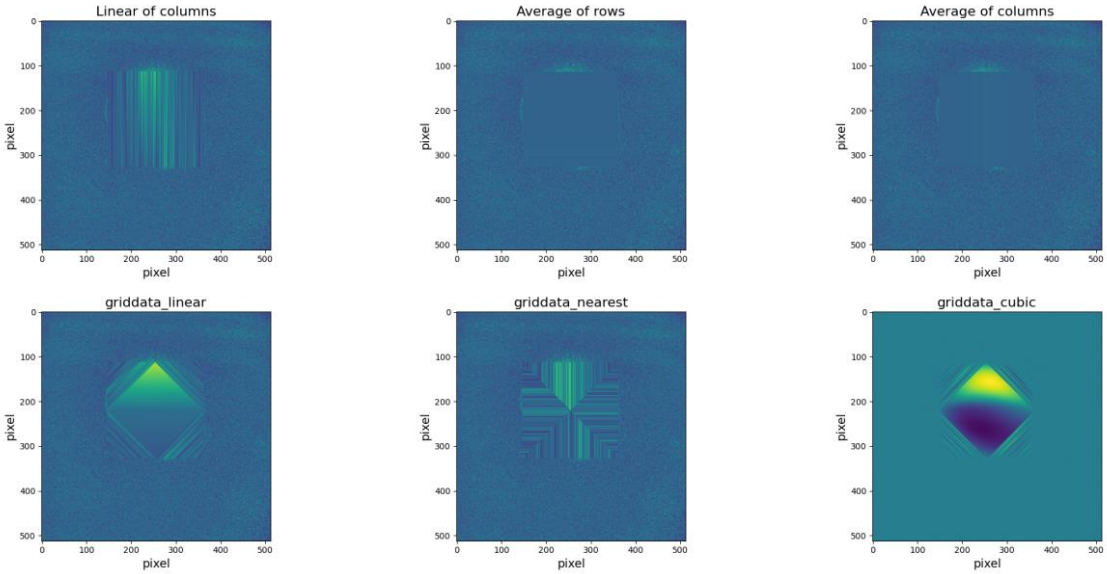
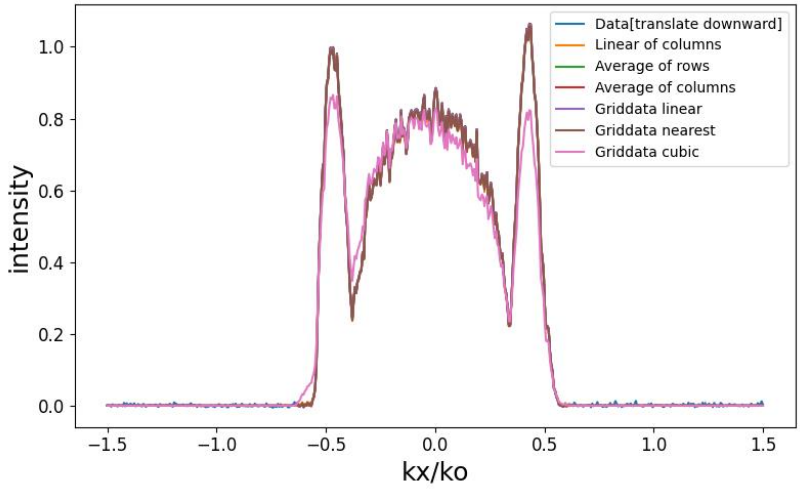


Figure 20 Estimated background calculated by linear of columns, average of rows, average of columns, griddata linear, griddata nearest, and griddata cubic methods.

Figure 21 shows cross-sections of the emission pattern before and after background subtraction.



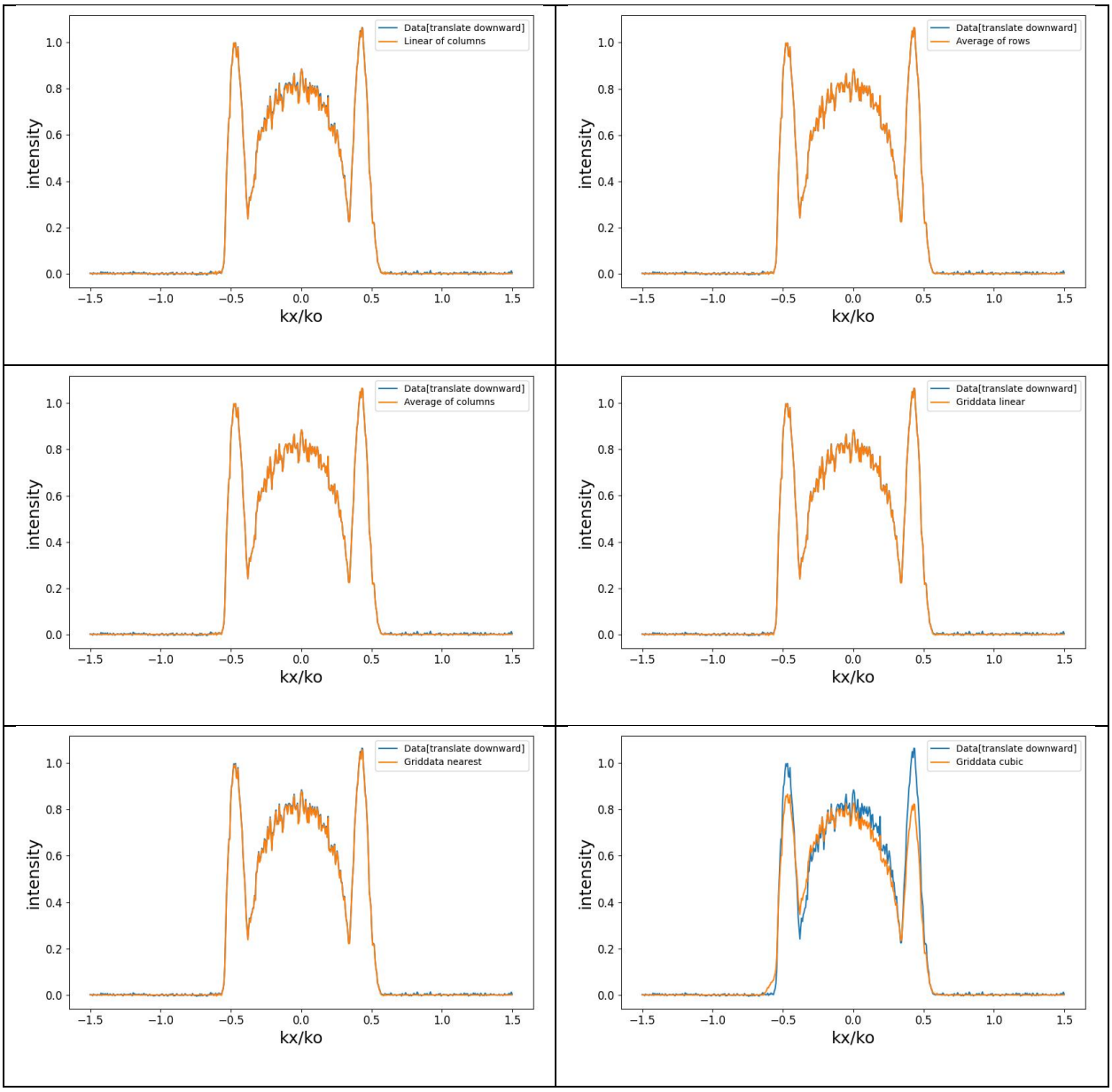


Figure 21 Comparisons of kx cross-sections between the original experimental data and data after removing background by different methods.

According to the cross-sections comparison and the criteria listed above, the best background subtraction methods in this example are “average of rows” and “average of columns” since the background simulation generated by these methods matches well with the original data. It is

worth noticing that there is not an absolute best background subtraction method since the experimental setups and environments vary case by case.

4.4 Methods to Normalize Back Focal Plane Signal

Since the intensity of the experimental emission pattern changes with experimental conditions such as excitation intensity and integration time, normalization is needed before the calculation of the best-fit angle. We normalize the signal at the center, $(k_x, k_y) = (0, 0)$, to equal 1 in both the theoretical and experimental data so they can be easily compared. Because of the noise inherent in the experimental data, we normalize to a 5X5 pixel array around $(k_x, k_y) = (0, 0)$. However, the intensity of the experimental emission pattern is affected by the experimental conditions and the background noise, so their normalization can still be imperfect even with the 5X5 pixel average (green boxes), as shown by the following cross-section (Figure 22).

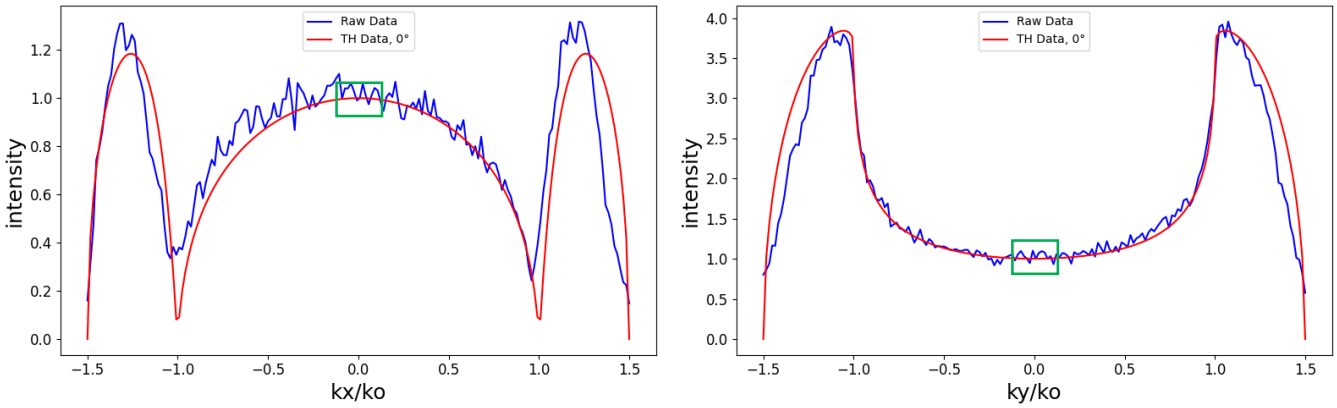


Figure 22 Comparison of kx and ky cross-sections from experimental data and calculated data.

To consider the fluctuation of the experimental data and properly normalize the experimental data, the fit factor (FF) is introduced. The fit factor is a constant, which can translate the whole experimental data upward (intensity increases) or downward (intensity decreases) to make the intensities of theoretical and experimental data at the same level. To calculate fit factors, we need

to take a look at the 5-pixel intensity range right at the middle of k_y cross-section of the BFP emission pattern, then find out the maximum and minimum intensity points.

According to the following equations, minimum and maximum fit factors can be found.

Equation 15

$$FF_{min} \cdot Intensity_{MAX} = 1$$

$$FF_{Max} \cdot Intensity_{min} = 1$$

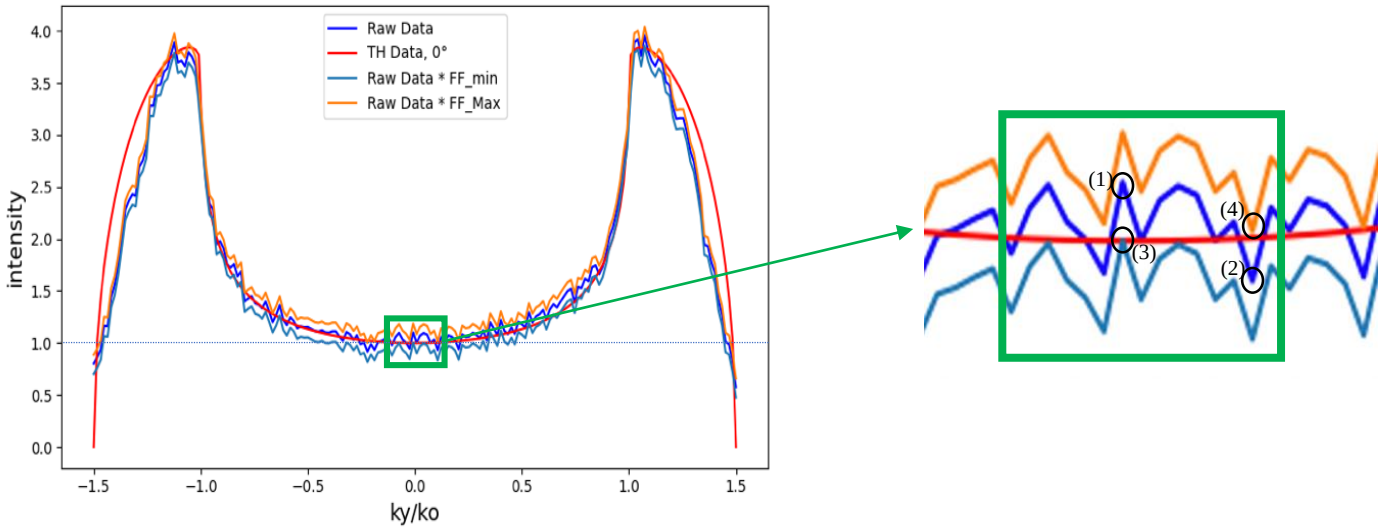


Figure 23 Comparison of ky cross-sections from raw data and data adjusted by fit factors. (1) $Intensity_{MAX}$ (2) $Intensity_{min}$ (3) FF_{min} * $Intensity_{MAX}$ = 1 (4) FF_{MAX} * $Intensity_{min}$ = 1.

Fit factors are a continuous range of constants, the intensity of raw data adjusted by these factors is at the same level as theoretical data overall. To find out the best fit factor, we can use the concept from Equation 13 discussed in Chapter 3.

Equation 16

$$\chi^2 = \frac{(k_y\ cross-section_{theoretical} - k_y\ cross-section_{experimental} \cdot fit\ factor)^2}{k_y\ cross-section_{theoretical}}$$

When χ^2 is minimum, the intensities between the raw data · fit factor and theoretical data when TDM at 0° are the closest. It is worth noticing that we use the k_y cross-section to calculate the best fit factor since the cross-sections at this direction keep the same at smaller angles (Figure 10), therefore, it is a better standard.

4.5 Radius (Area/Pixels) of Fitting Range

The number of the data points/pixels of the experimental emission pattern depends on the resolution of the camera, the magnification of the imaging lenses, and the numerical aperture of the lens used in the experiment. The number of the data points used for angle-fitting is proportional to the accuracy of the result: the uncertainty increases as the number of pixels decreases.

Figure 24 shows the biggest uncertainty/error under different conditions (refractive index and thickness of the emission layer) with different numbers of data points (100X100 and 500X500). For data size 100X100, the maximum fitting TDM angle error is 20°, however, for data size 500X500 is only 8°, which clearly shows that the more data points used in the calculation, the more accurate the TDM angle-fitting outcome will be. Additionally, as mentioned earlier, the fitting angle's error increases when the TDM angle decreases, which is also shown by this study. The figures also point out that the amount of uncertainty is more strongly related to the refractive index, rather than the thickness of the emission layer as the uncertainty generally increases along the x-axis (refractive index) than the y-axis (thickness) for all plots.

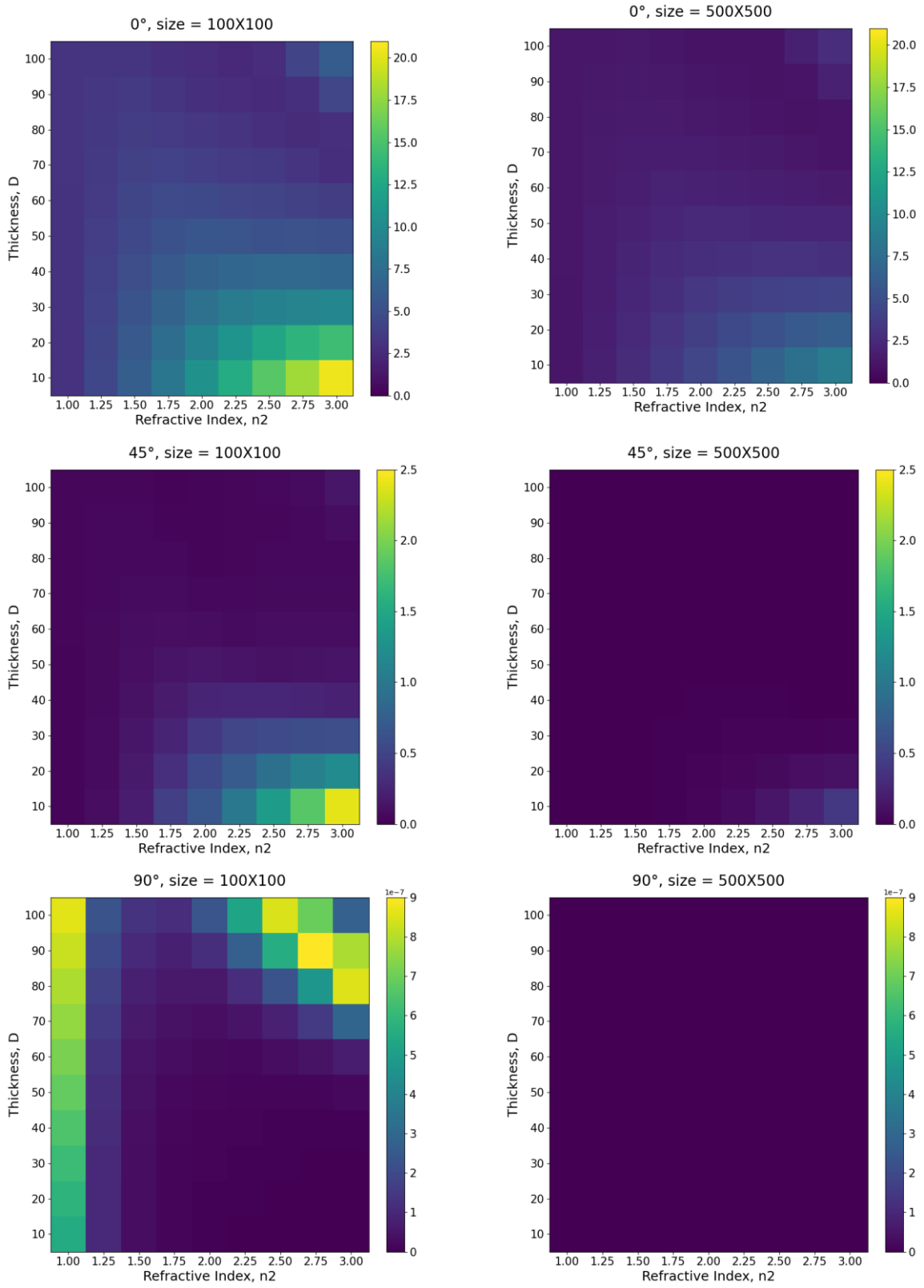


Figure 24 Maximum fitting TDM angle uncertainty under different emission layer thicknesses, refractive indexes, and data sizes. n_1=1, n_2=2, n_3=1.5, d=0.5, wavelength = 500nm.

4.6 Conclusion

In this chapter, all steps of the TDM angle analysis are discussed. Starting from BFP signal simulation data-generating, it is important not to resize data by interpolation since it will lose important features in the image and add an extra 1° or 2° of uncertainty. Second, for background subtraction, it is crucial to choose a subtraction method that does not change the shape of the cross-section of the original data set. Third, I discussed how to normalize the experimental data set through fit factors to account for the extra noise of the data. Finally, the more data points used in the calculation, the more accurate outcomes are. Therefore, the combination of the resolution of the camera, the magnification of the imaging lenses, and the numerical aperture of the lens used in the experiment need to be optimized such that the experimental image spans as many pixels as possible. By addressing all these criteria, the uncertainty that comes from the TDM angle analysis process can be minimized.

5. Accuracy/Precision Limit of TDM Angle-Fitting

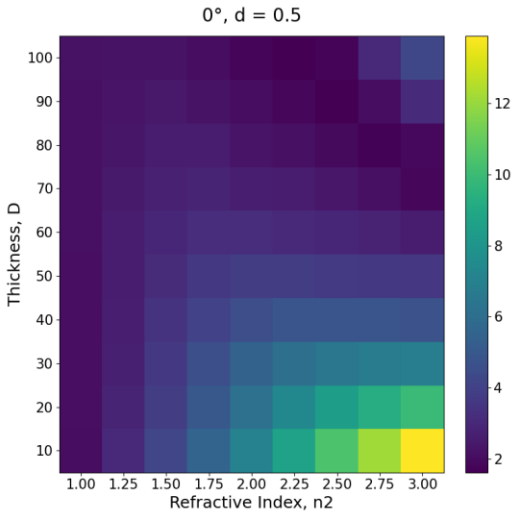
In the following chapter, the maximum uncertainty and error of fitting TDM angles for different samples are discussed, including different refractive indexes and thicknesses of the emission layer, different numbers of dipoles, different positions where dipoles are, and different amounts of background noise of optical instruments. With this information, the range of the fitting TDM angle's uncertainty and the main factor which affects the accuracy the most can be revealed.

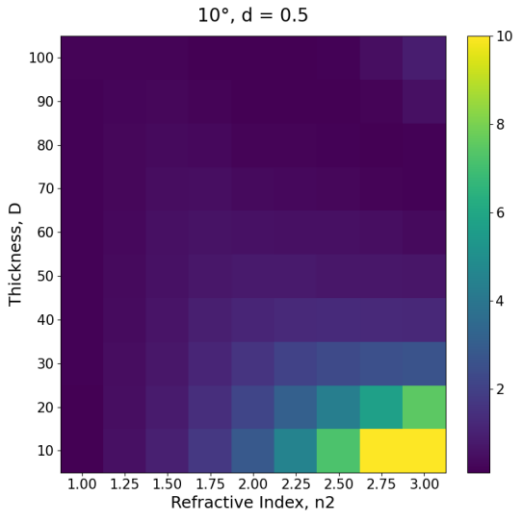
The following list gives examples of different categories of samples:

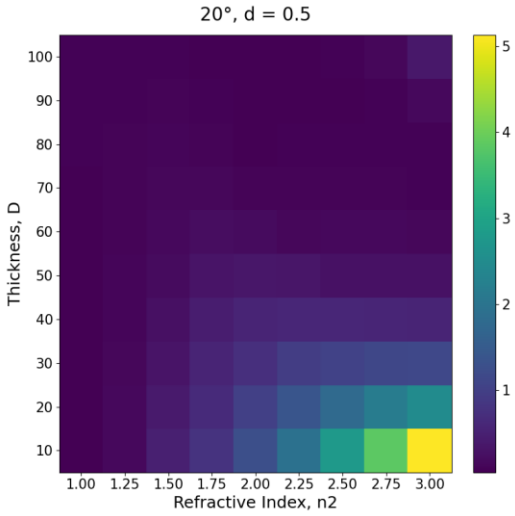
- Low refractive index: individual emitters such as dyes or nanoparticles, e.g. CdSe/ZnS (cadmium selenide/zinc sulfide) quantum dots [15]
- Medium refractive index: more continuous film of nanocrystals, e.g. CdSe (cadmium selenide) nanoplatelet films [11]
- High refractive index: thin-film of semiconductor, e.g. MoS_2 (molybdenum disulfide) [10]
- Single dipole with different emission layer positions: embedding individual emitter in a film
- Multiple dipoles: dipoles in a thicker semiconductor film

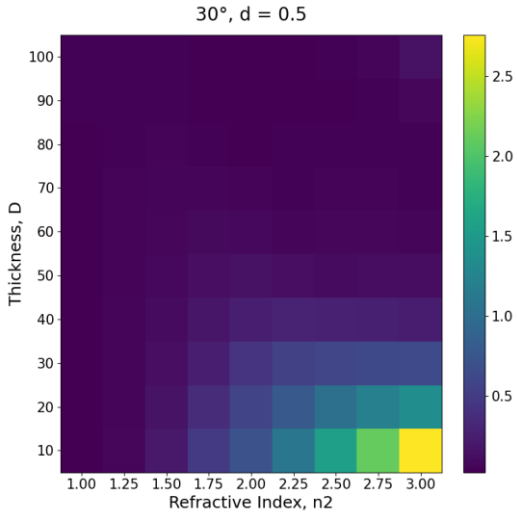
5.1 Sigle Dipole - Limit of accuracy when fitting a single dipole at the middle of the emission layer

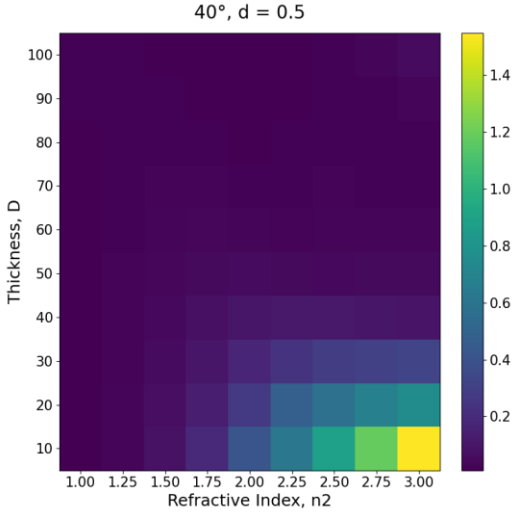
Figure 25 shows the maximum uncertainty/error of angle-fitting for a single dipole moment at the middle of the emission layer (d=0.5).

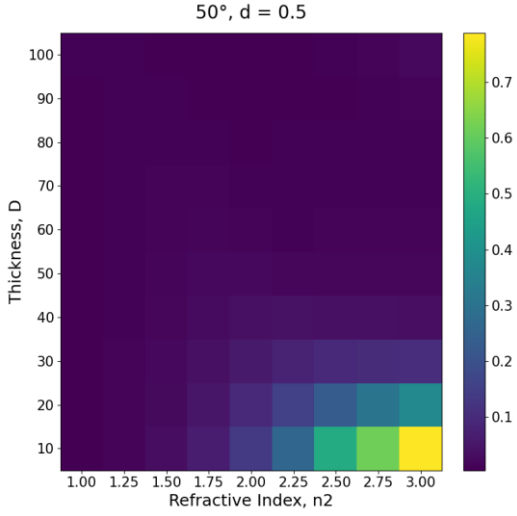

0°, d = 0.5
Thickness, D
Refractive Index, n2


10°, d = 0.5
Thickness, D
Refractive Index, n2


20°, d = 0.5
Thickness, D
Refractive Index, n2


30°, d = 0.5
Thickness, D
Refractive Index, n2


40°, d = 0.5
Thickness, D
Refractive Index, n2


50°, d = 0.5
Thickness, D
Refractive Index, n2

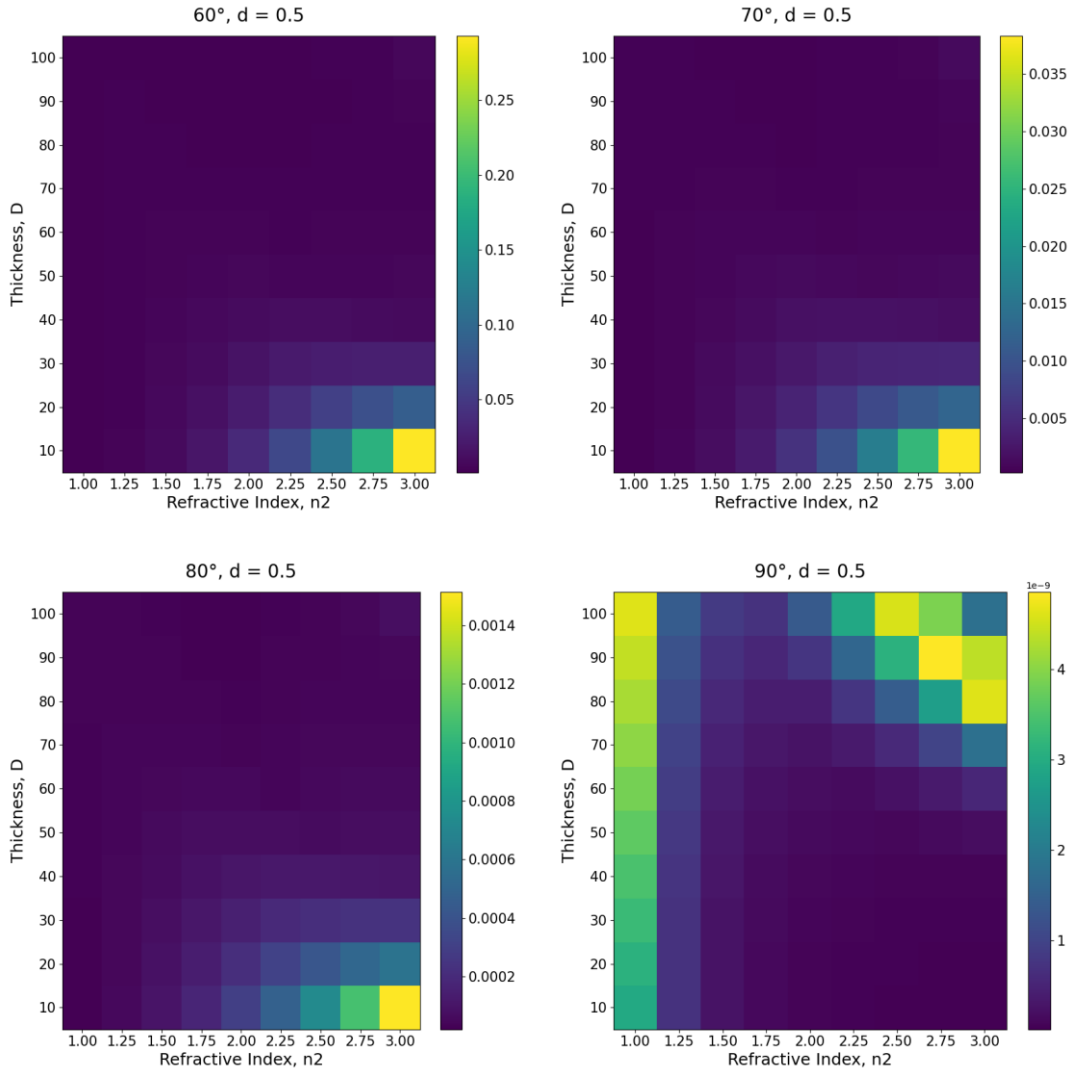


Figure 25 Maximum fitting TDM angle uncertainty from a single TDM at the middle of the emission layer (d=0.5). n_1=1, n_2=2, n_3=1.5, wavelength = 500nm, size=200X200.

When the angle of the dipole moment becomes smaller, the uncertainty increases. This is because the difference between BFP patterns of smaller angles is smaller compared to bigger angles (as shown by Figure 10 and Figure 26). Thus, it is significantly more difficult to fit smaller angles. Additionally, the error around the best-fit angle is not even, and there is more

uncertainty on the lower bound because of this same phenomenon (Figure 27). (Refer to Figure 13 for lower-bound angle definition)

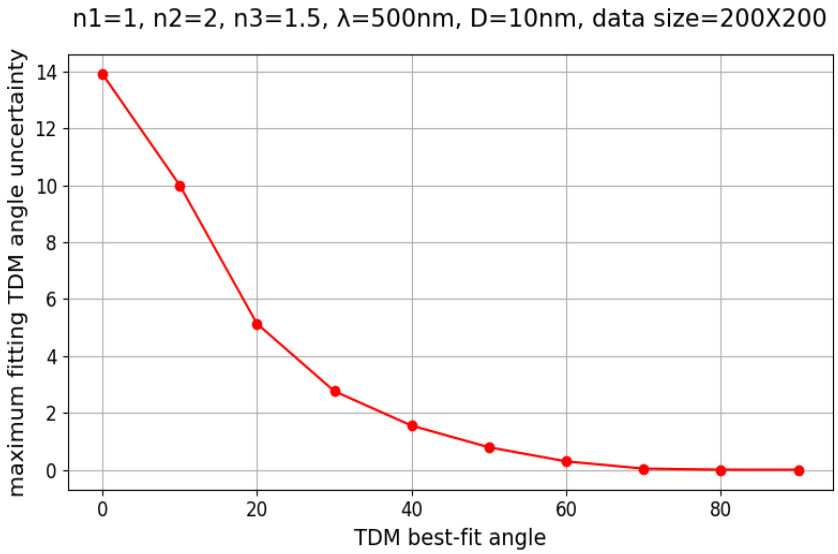


Figure 26 The relationship between the TDM best-fit angle and maximum fitting TDM angle uncertainty.

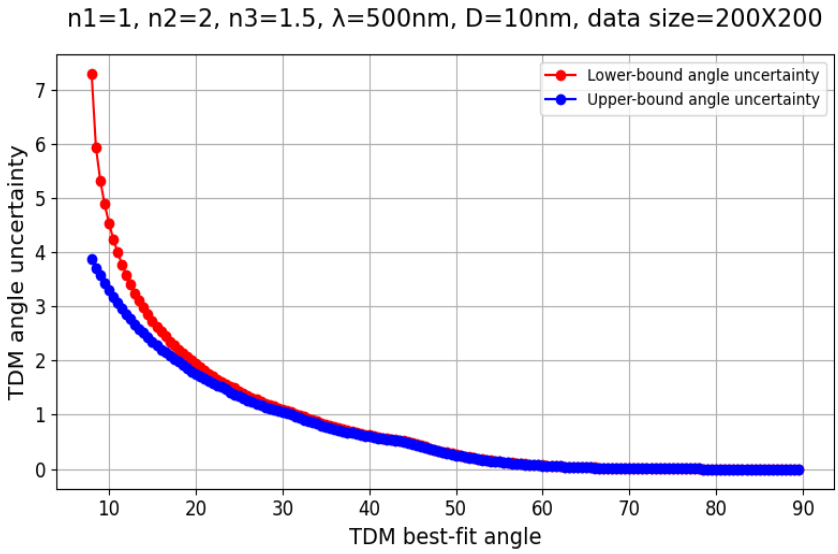
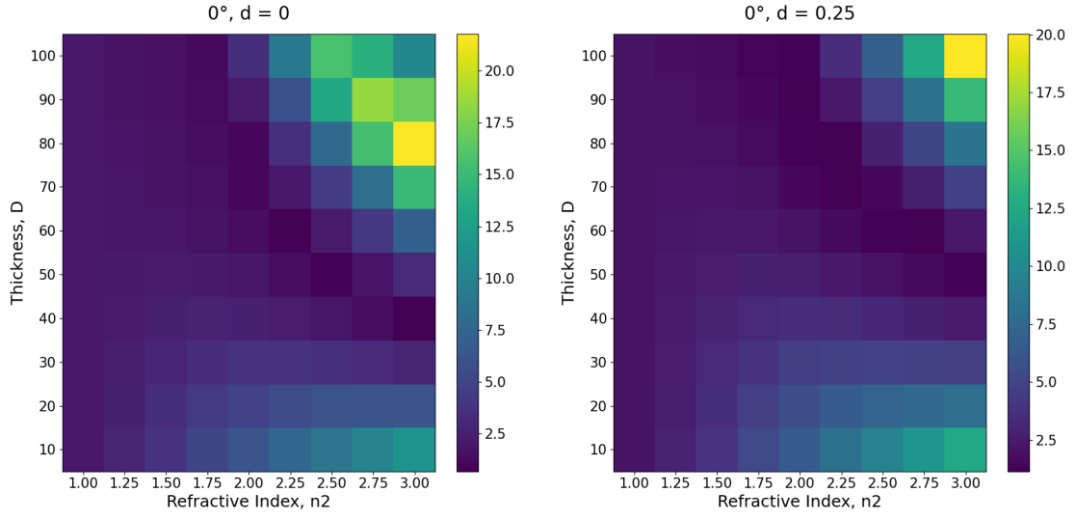


Figure 27 The relationship between the TDM best-fit angle and lower/upper-bound fitting TDM angle uncertainty.

5.2 Position of the Dipole - How the position of a dipole affects the TDM angle-fitting?

For samples that have a single emitter, such as a dye or nanoparticle embedded in a film, changing the position of the emitter within the film changes the BFP image [15]. Additionally, the uncertainty of dipole moment angle-fitting is also related to the position of the dipole.

Figure 28 shows the maximum fitting TDM angle uncertainty with different single TDM positions in the emission layer. Here, d means the position of TDM in the emission layer as shown in Figure 8, when d = 0, TDM is at the top of the thin film; d=1, TDM is at the bottom of the thin film. (Figure 8)



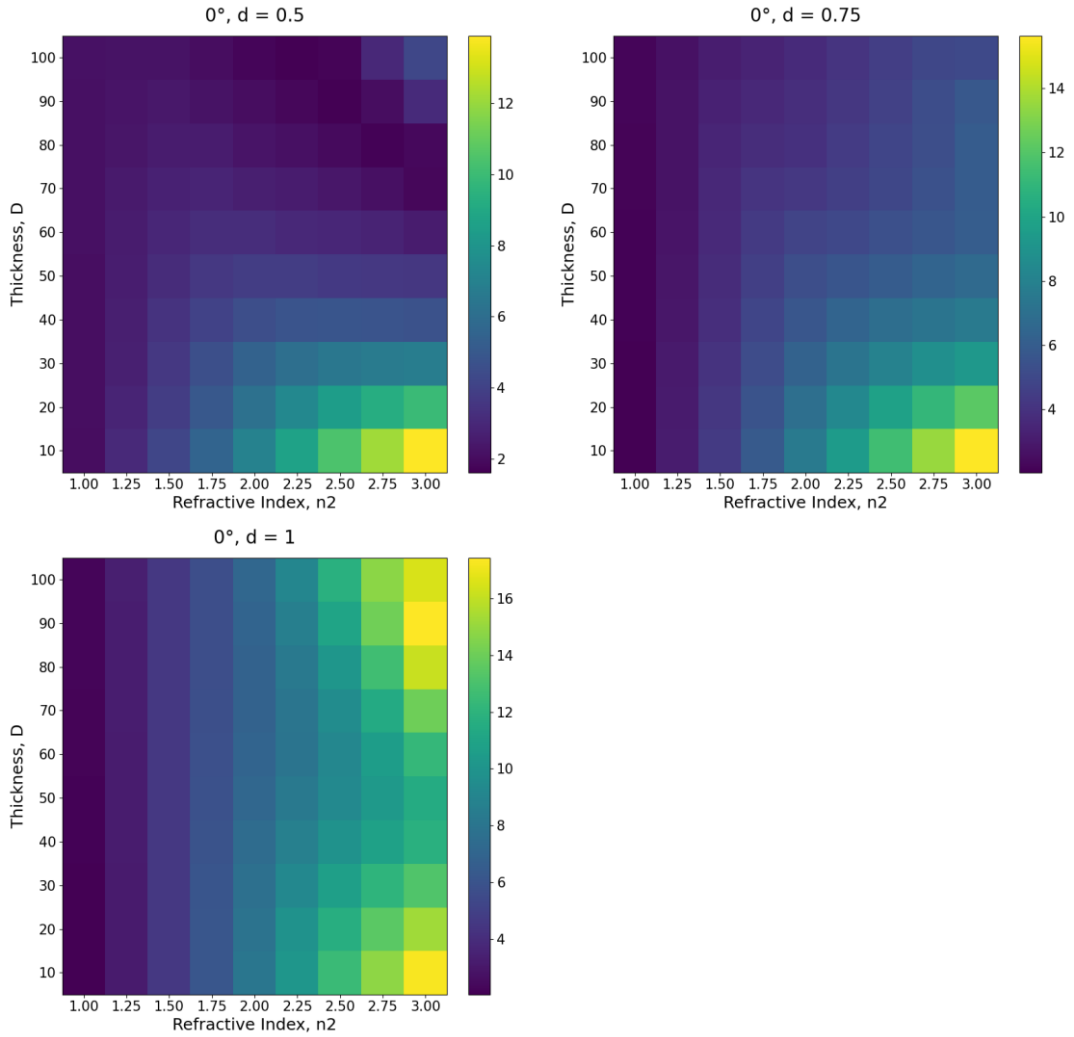


Figure 28 Maximum fitting TDM angle uncertainty from a single TDM at various positions of the emission layer. n_1=1, n_2=2, n_3=1.5, wavelengh = 500nm, size=200X200.

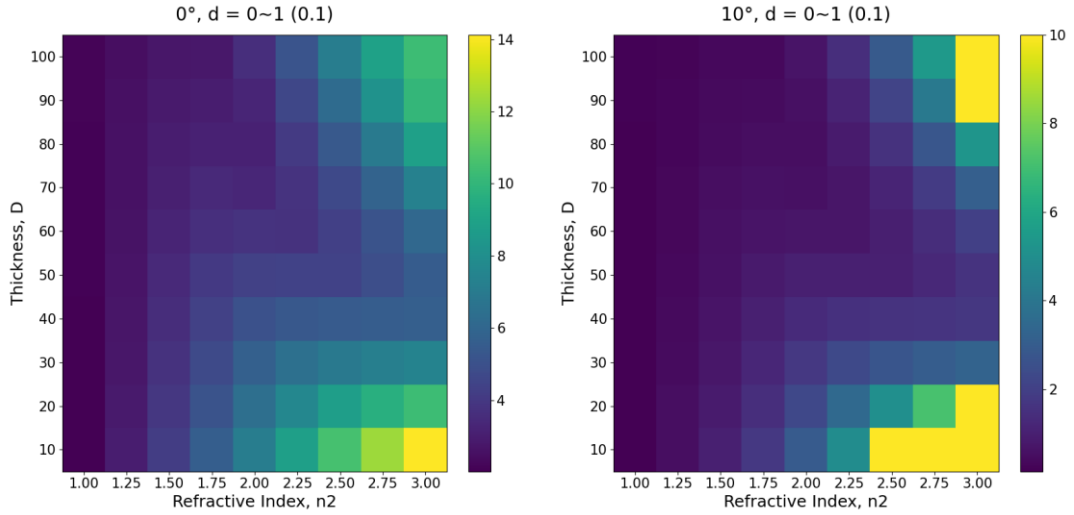
The maximum uncertainty happens when the dipole is right on the top of the emission layer (d = 0; air-thin film interface). The lowest uncertainty appears when the dipole is in the middle of the emission layer (d = 0.5). Since the dielectric interface serves as a mirror, which can redirect the fluorescence emission from TDM, a large fraction of the emission pattern is emitted close to the critical angle [15]. In this case, the photon collection efficiency decreases by the limitation of the

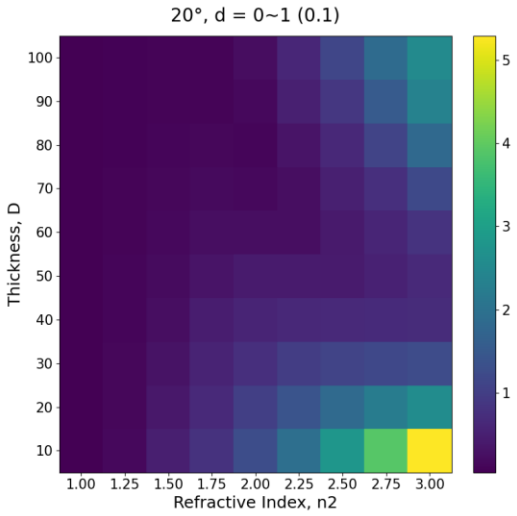
numerical aperture used in the BFP image setup, therefore, the uncertainty of the TDM which is close to the dielectric interface is greater.

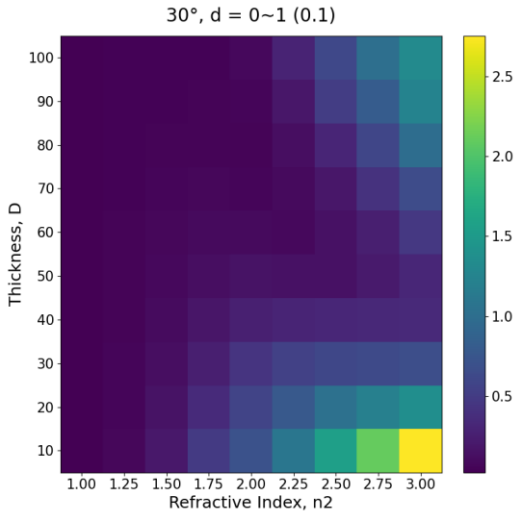
5.3 Multiple Dipoles - How the number of dipoles affect the TDM angle-fitting?

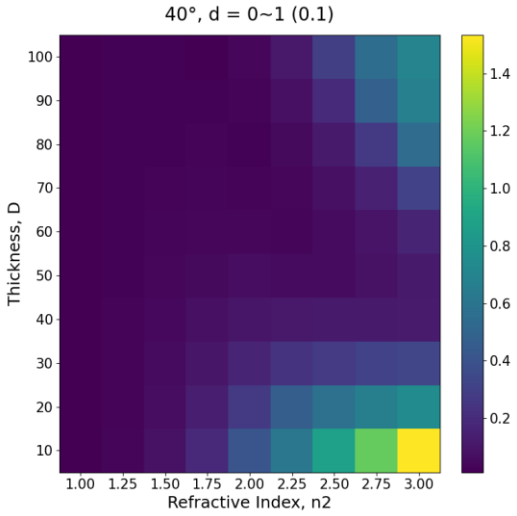
For samples investigating individual dyes or nanocrystals, using one dipole is sufficient to describe the system [15]. However, samples that are comprised of thin-film semiconductors will have many dipoles throughout the film that contribute to the BFP pattern. Here, we investigate how multiple dipoles affect the uncertainty of determining the TDM.

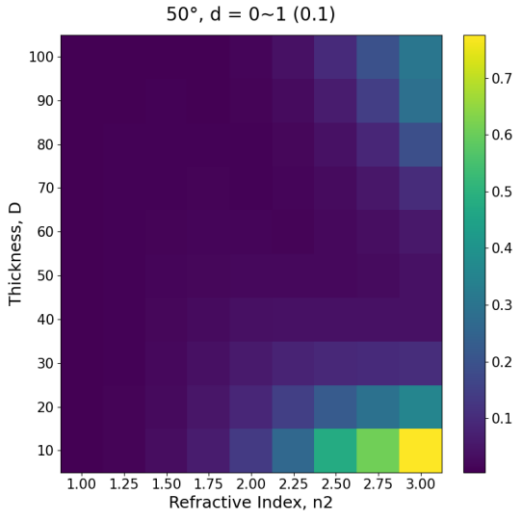
Figure 29 shows the maximum uncertainty/error of angle-fitting for systems with 11 dipole moments locate in the emission layer. These dipoles are spaced equally throughout the film (spacing = 0.1*D where D is the thickness of the emission layer).

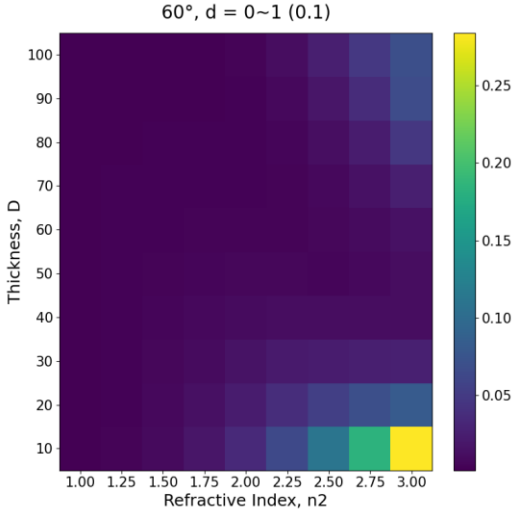


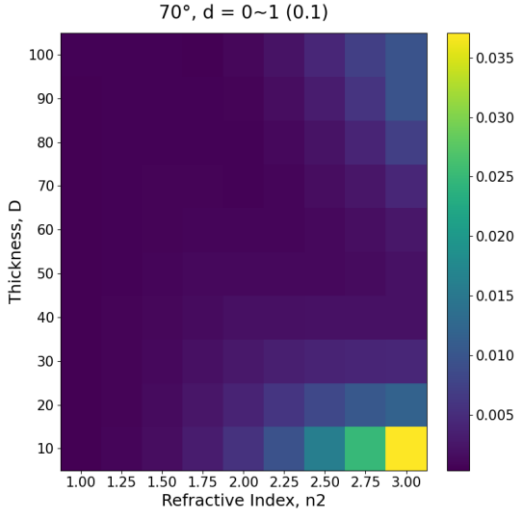

20°, d = 0~1 (0.1)
Thickness, D
Refractive Index, n2


30°, d = 0~1 (0.1)
Thickness, D
Refractive Index, n2


40°, d = 0~1 (0.1)
Thickness, D
Refractive Index, n2


50°, d = 0~1 (0.1)
Thickness, D
Refractive Index, n2


60°, d = 0~1 (0.1)
Thickness, D
Refractive Index, n2


70°, d = 0~1 (0.1)
Thickness, D
Refractive Index, n2

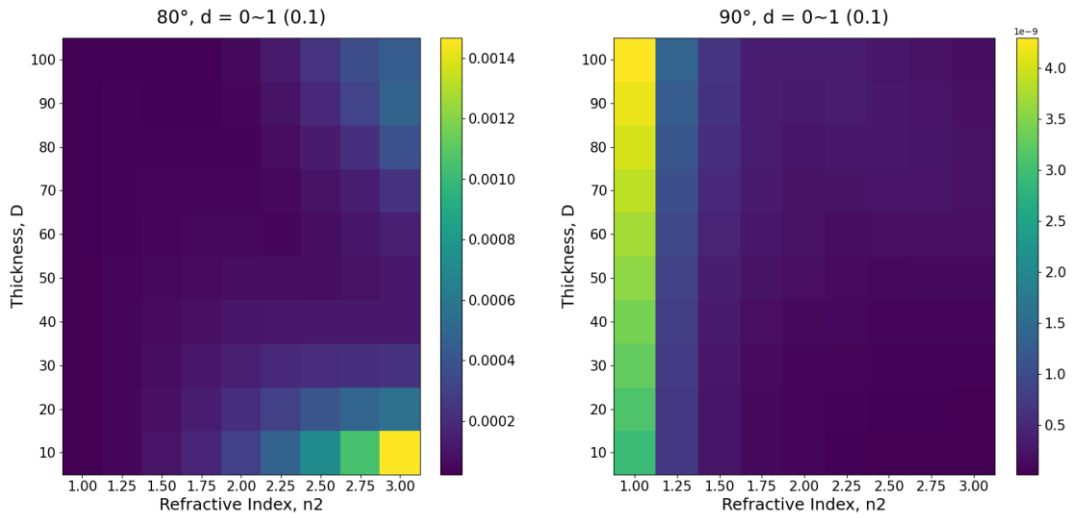


Figure 29 Maximum fitting TDM angle uncertainty from 11 TDMs spaced equally throughout the emission layer (d=0.1, 0.2, …, 1). n_1=1, n_2=2, n_3=1.5, wavelengh = 500nm, size=200X200.

Compared to the maximum uncertainties of fitting TDM angles between single and multiple (#11) dipole moments, we can see that their maximum error levels are the same, which means the uncertainty of the TDM angle fitting is not strongly affected by the number of dipoles. This is expected for thin (D <20 nm) samples, as the thickness is much smaller than the emission wavelength. Many groups who study nanocrystal thin films or 2D monolayers only use one dipole to describe these structures [10][11], and we verify here that it does not adversely affect the fitting.

It is more surprising to see that the uncertainty does not change when adding more dipoles for thicker samples on the order of the wavelength of emission (D = 100 nm). We notice that the uncertainty from 11 dipoles looks similar to the uncertainty of a single dipole near the bottom of the emission layer (d=1). This is because the closer the dipole is to the collection point, the stronger the BFP signal is. Therefore, the dipoles closest to the substrate dominates the

maximum TDM angle-fitting error patterns (Figure 30). This is an important result from a data processing standpoint: being able to model a thick film comprised of many dipoles by using just one or a few dipoles can save significant computation time.

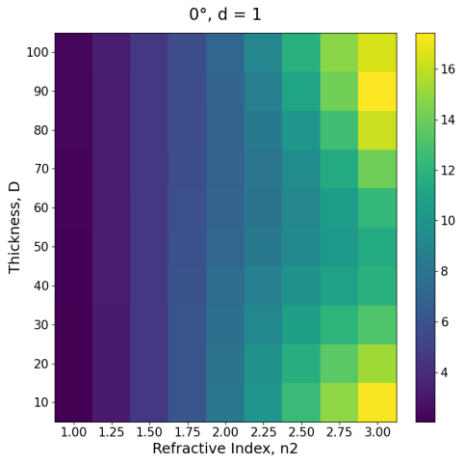


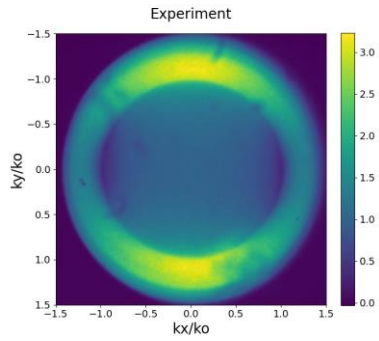
Figure 30 Maximum fitting TDM angle uncertainty from a single TDM at the bottom of the emission layer. n_1=1, n_2=2, n_3=1.5, wavelength = 500nm, size=200X200.

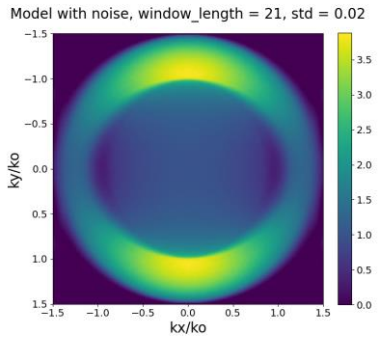
5.4 Background Noise - How the background noise affects the TDM angle-fitting

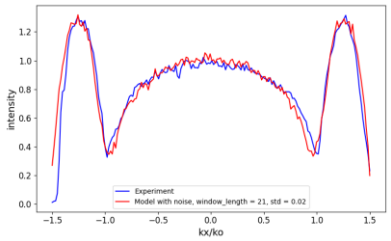
To reveal the accuracy/precision of the dipole moment angle-fitting from experimental data, it is crucial to consider noise since it is inevitable in the types of cameras used. The experimental raw data can be mimicked by adding noise or using a smoothing function to recreate the blurring seen in actual experimental images. (the Python scripts of this method can be found in Appendices A.1.6. The functions to add noise and smooth/blurring are “std” and “window length “, respectively) By using this artificial data, the uncertainty and precision of the angle-fitting under a certain amount of noise and blurring can be predicted.

Figure 31 shows an example of using this data processing to mimic actual experimental data. The appropriate amount of noise and blurring can be checked by comparing the standard deviations

of cross-sections since standard deviation can measure the amount of variation or dispersion of a set of values, which quantify the fluctuation of the data. Here, the standard deviations of experimental data and simulation in kx cross-section are compared, which demonstrates that the amount of noise and blurring is added appropriately.



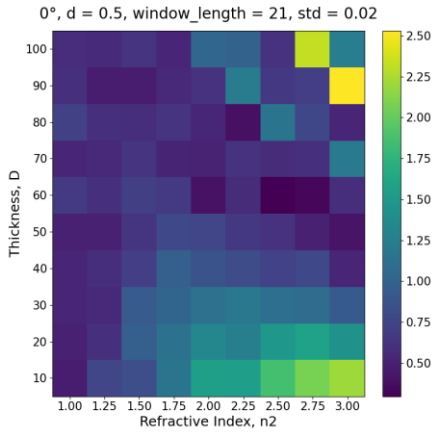


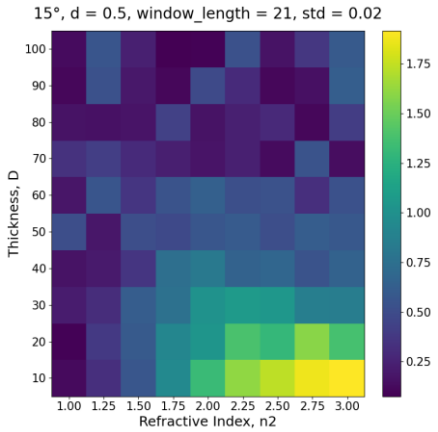


kx = +- 0.5 → std = 0.03357779 kx = +- 1.0 → std = 0.17412967 kx = +- 1.5 → std = 0.26125608	kx = +- 0.5 → std = 0.03837299 kx = +- 1.0 → std = 0.20182109 kx = +- 1.5 → std = 0.24910926	

Figure 31 Left and middle: TDM emission patterns of experimental data and model with artificial noise. Right: Comparison of kx cross-sections from experimental data and model with artificial noise.

Figure 32 shows the maximum fitting TDM angle errors calculated from the artificial data discussed right above. With this information, we know that the maximum uncertainty is 2.5° under a certain amount of noise and blurring.





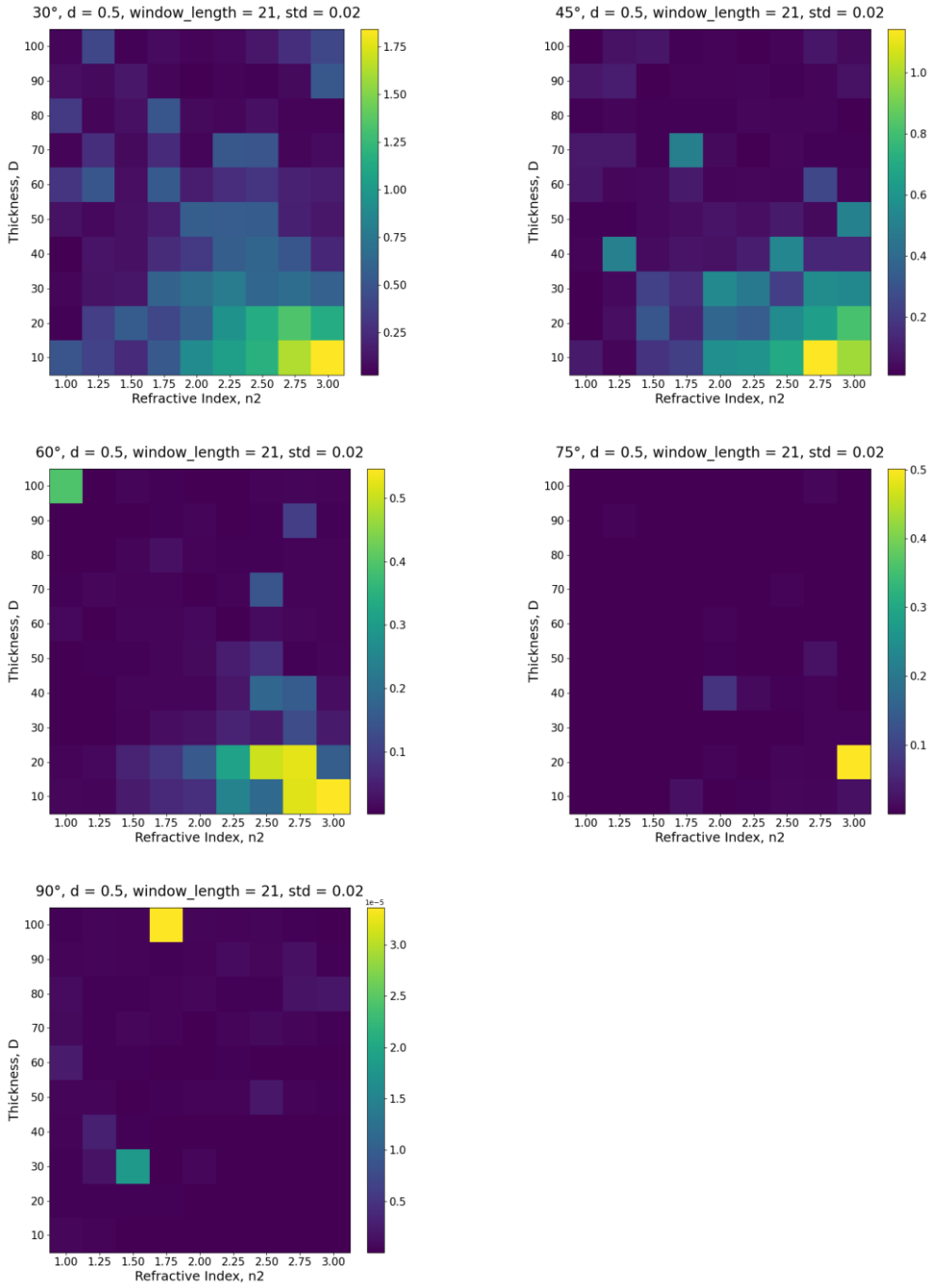


Figure 32 Maximum fitting TDM angle uncertainty from a single TDM at the middle of the emission layer with background noise. Noise is adding by function “std”; Smooth/blur is adding by function “ window length”. n_1=1, n_2=2, n_3=1.5, wavelength = 500nm, size=200X200.

5.5 Conclusion

In this chapter, I showed how the conditions of TDM itself and the local environment affect the accuracy of TDM angle-fitting results. First, we know that the uncertainty of TDM angle-fitting is greater for smaller angles. With the data size 200X200 pixels, the uncertainty of TDM angle can up to 14°. Secondly, the position of TDM also plays a huge role in affecting accuracy. When a single dipole such as dye or nanoparticle with a smaller angle is close to the dielectric interface, the TDM angle uncertainty can over 20°. This points out the position where the individual emitter is embedded is important for accurate TDM angle analysis. Third, I found that the number of the dipoles does not have a significant impact on the accuracy of the TDM angle fitting results in thin samples, which thickness is much smaller than the emission wavelength. Therefore, when fitting TDM angle for materials with multiple dipoles such as thin-film semiconductors, we can use one dipole to describe the whole system. Finally, a data processing method to mimic actual experimental data is introduced, which can predict the uncertainty and precision of the angle-fitting under a certain amount of noise and blurring. In conclusion, during the process of TDM angle fitting, with a BFP emission pattern which contains 200X200 data points, we can expect the error of the angle-fitting result is bound between 20° to 0°, which highly depends on the nature and the local environment of the TDM.

6. Conclusion and Outlook

The goal of this work was to determine how the steps in the process of fitting a transition dipole moment (TDM) orientation from back focal plane (BFP) images affect the accuracy and precision of the study. In summary, I explored factors that could affect the accuracy and precision of TDM angle analysis, including theoretical data-generating, emission pattern centering, background subtracting, BFP signal normalizing, and BFP signal fitting-range deciding. From these factors, I determined the following principles that could ensure accurate determination of the orientation of TDM:

- Generate theoretical data of TDM emission pattern from equations (the model in chapter 3) as interpolation can introduce inaccurate features in the theoretical BFP images, and add an extra 1° or 2° of uncertainty
- Ensure accurate determination of the position of the BFP signal (for example: creating an extra step to make sure that the dips and the peaks of the emission pattern's cross-section are overlapped well between theoretical and experimental data)
- Choose an appropriate method for background subtraction depends on the experiment environment by the following criteria:
  - $\Rightarrow$ intensity outside the emission pattern should be as close to zero as possible after background subtraction
  - $\Rightarrow$ the shape of the cross-section of the emission pattern should keep the same before and after background subtraction
- Apply an appropriate fit factor to adjust the value of light intensity from experimental data to the same level as simulation

- Optimize resolution of the camera, the magnification of the imaging lenses, and the numerical aperture of the lens used in the experiment to collect as many data points as possible (the maximum angle uncertainty for data size 100X100 is over 20°, for data size 500X500 is 8°)

Additionally, the maximum uncertainty of fitting TDM angles under various local environments of dipoles are discussed, including different refractive indexes and thicknesses of the emission layer, different numbers of dipoles, different positions where dipoles are, and different amounts of background noise of optical instruments. First, the uncertainty increases when the TDM angle decreases because the difference between emission patterns at smaller angles is tinny. Second, I have shown that the position of single TDM such as dye and nanoparticle in the emission layer also affects the uncertainty. With the data size: 200X200, the maximum uncertainty (21°) appears when the dipole is at the air-thin film interface (d=0); the lowest uncertainty (14°) happens when the dipole is in the middle of the emission layer (d = 0.5). Third, angle-fitting for emissive materials with multiple dipoles such as thin-film semiconductors is also investigated. When the thickness of the thin film is much smaller than the emission wavelength, a single dipole in the middle of the thin film can represent the whole system. Finally, a method to add background noise to simulation emission patterns is introduced, which provides a way to mimic experimental data and estimate the uncertainty of fitting TDM angles. In the longer term, the analysis in this work could be used to develop high-efficiency optoelectronic devices by manipulating TDMs.

Appendices

A.1 Python Scripts

A.1.1 Scripts of Back Focal Plane Calculated Signal Model

The following scripts show to get the back focal plane calculated signal by Python based on the discussion in chapter 3.

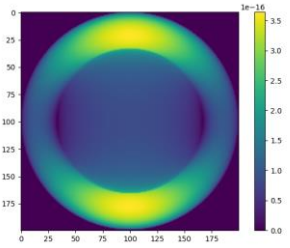
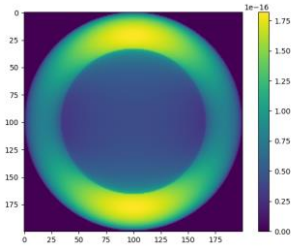
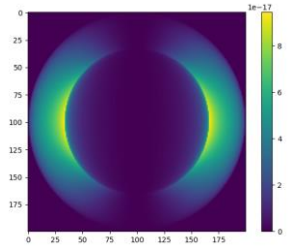
Before calculation, the size of the emission pattern (`size`), the refractive index of the three layers in the system (`n1, n2, n3`), and wavelength of the laser used to excite the emissive material(`wave_length_nm`), the thickness of the emission layer(`thickness_nm`), the position of the TDM in the emission layer(`d`), and the angle of the TDM(`angle`) need to be determined first.

```
import cmath
from cmath import pi, sin, exp
import numpy as np
import matplotlib.pyplot as plt


size = 200  # The kx, ky size of the TDM emission pattern (total: size x size)
n1 = 1  # The refractive index of air
n2 = 2  # The reflective index of the thin-film/emission layer
n3 = 1.5  # The reflective index of the substrate
wave_length_nm = 430  # wavelength (unit: nm)
wave_length = wave_length_nm * pow(10, -9)
thickness_nm = 10  #thickness of the thin-film/emission layer (unit: nm)
D = thickness_nm * pow(10, -9)
d = 0.5  # The distance between the TDM and the air-thin film interface
h = 1 - d  # The distance between the TDM and the substrate-thin film interface
angle = 0  # The angle between the TDM and the substrate
tdm = pi / 180 * angle


def kz1(kx, ky, wave_length, n1):
    return cmath.sqrt(
        (n1 ** 2) * ((2 * pi / wave_length) ** 2) - ((cmath.sqrt(kx ** 2 + ky ** 2))
** 2))

def kz2(kx, ky, wave_length, n2):
    return cmath.sqrt(
        (n2 ** 2) * ((2 * pi / wave_length) ** 2) - ((cmath.sqrt(kx ** 2 + ky ** 2))
** 2))
```

```
def kz3(kx, ky, wave_length, n3):
    return cmath.sqrt(
        (n3 ** 2) * ((2 * pi / wave_length) ** 2) - ((cmath.sqrt(kx ** 2 + ky ** 2))
** 2))

def ts_3_2(kx, ky, wave_length, n2, n3):
    return 2 * kz3(kx, ky, wave_length, n3) / (
                kz3(kx, ky, wave_length, n3) + kz2(kx, ky, wave_length, n2))

def tp_3_2(kx, ky, wave_length, n2, n3):
    return 2 * n3 * n2 * kz3(kx, ky, wave_length, n3) / (
                n2 ** 2 * kz3(kx, ky, wave_length, n3) + n3 ** 2 * kz2(kx, ky,
wave_length, n2))

def rs_2_1(kx, ky, wave_length, n1, n2):
    return (kz2(kx, ky, wave_length, n2) - kz1(kx, ky, wave_length, n1)) / (
                kz2(kx, ky, wave_length, n2) + kz1(kx, ky, wave_length, n1))

def rs_2_3(kx, ky, wave_length, n2, n3):
    return (kz2(kx, ky, wave_length, n2) - kz3(kx, ky, wave_length, n3)) / (
                kz2(kx, ky, wave_length, n2) + kz3(kx, ky, wave_length, n3))

def rp_2_1(kx, ky, wave_length, n1, n2):
    return (n1 ** 2 * kz2(kx, ky, wave_length, n2) - n2 ** 2 * kz1(kx, ky,
wave_length, n1)) / \
           (n1 ** 2 * kz2(kx, ky, wave_length, n2) + n2 ** 2 * kz1(kx, ky,
wave_length, n1))

def rp_2_3(kx, ky, wave_length, n2, n3):
    return (n3 ** 2 * kz2(kx, ky, wave_length, n2) - n2 ** 2 * kz3(kx, ky,
wave_length, n3)) / \
           (n3 ** 2 * kz2(kx, ky, wave_length, n2) + n2 ** 2 * kz3(kx, ky,
wave_length, n3))


def rho_s_x(wave_length, D, kx, ky, n1, n2, n3):
    return (1 / (8 * pi * (2 * pi / wave_length) ** 2)) * ((2 * pi / wave_length) /
kz3(kx, ky, wave_length, n3)) * \
          abs((ts_3_2(kx, ky, wave_length, n2, n3) * exp(1j * kz2(kx, ky,
wave_length, n2) * (d * D)) * \
                (1 + rs_2_1(kx, ky, wave_length, n1, n2) * exp(2j * kz2(kx, ky,
wave_length, n2) * (h * D)))) / \
               (1 - rs_2_1(kx, ky, wave_length, n1, n2) * rs_2_3(kx, ky, wave_length,
n2, n3) * \
                exp(2j * kz2(kx, ky, wave_length, n2) * D)) * (ky / cmath.sqrt(kx ** 2
+ ky ** 2))) ** 2


def rho_p_x(wave_length, D, kx, ky, n1, n2, n3):
    return (1 / (8 * pi * (2 * pi / wave_length) ** 2)) * ((2 * pi / wave_length) /
kz3(kx, ky, wave_length, n3)) \
           * abs((tp_3_2(kx, ky, wave_length, n2, n3) * exp(1j * kz2(kx, ky,
wave_length, n2) * (d * D)) *
                  kz2(kx, ky, wave_length, n2) / (n2 * (2 * pi / wave_length)) * (1 -
rp_2_1(kx, ky, wave_length, n1, n2)
                * exp(2j * kz2(kx, ky, wave_length, n2) * (h * D)))) / (1 - rp_2_1(kx,
ky, wave_length, n1, n2)
                * rp_2_3(kx, ky, wave_length, n2, n3) * exp(2j * kz2(kx, ky,
wave_length, n2) * D)) * \
                 (kx / cmath.sqrt(kx ** 2 + ky ** 2))) ** 2
```

```
def rho_p_z(wave_length, D, kx, ky, n1, n2, n3):
    return (1 / (8 * pi * (2 * pi / wave_length) ** 2)) * ((2 * pi / wave_length) /
kz3(kx, ky, wave_length, n3)) * \
           abs((tp_3_2(kx, ky, wave_length, n2, n3) * exp(1j * kz2(kx, ky,
wave_length, n2) * (d * D)) * kx / (
                        n2 * (2 * pi / wave_length)) * \
                (1 + rp_2_1(kx, ky, wave_length, n1, n2) * exp(2j * kz2(kx, ky,
wave_length, n2) * (h * D)))) / \
               (1 - rp_2_1(kx, ky, wave_length, n1, n2) * rp_2_3(kx, ky, wave_length,
n2, n3) * exp(
                   2j * kz2(kx, ky, wave_length, n2) * D))) ** 2

def r1(kx, ky):
    return rho_s_x(wave_length, D, kx, ky, n1, n2, n3)
def r2(kx, ky):
    return rho_p_x(wave_length, D, kx, ky, n1, n2, n3)
def r3(kx, ky):
    return rho_p_z(wave_length, D, kx, ky, n1, n2, n3)


kxs = np.linspace(-1.5 * (2 * pi / wave_length), 1.5 * (2 * pi / wave_length), size)
kys = np.linspace(-1.5 * (2 * pi / wave_length), 1.5 * (2 * pi / wave_length), size)
x_1, y_1 = np.meshgrid(kxs, kys)


res1 = np.eye(size, size, dtype=complex)
for i in range(size):
    for j in range(size):
        res1[i][j] = r1(x_1[i][j], y_1[i][j])

res2 = np.eye(size, size, dtype=complex)
for i in range(size):
    for j in range(size):
        res2[i][j] = r2(x_1[i][j], y_1[i][j])

res3 = np.eye(size, size, dtype=complex)
for i in range(size):
    for j in range(size):
        res3[i][j] = r3(x_1[i][j], y_1[i][j])


n_intensity = (res1 + res2) * ((1 - (sin((tdm)) ** 2)) / 2) + (res3) * ((sin((tdm)) **
2))

n_intensity = n_intensity.real

# Normalization: normalized the emission pattern by the center point of the emission
pattern
# n_intensity = n_intensity / n_intensity[int(size / 2): int(size / 2) + 1, int(size /
2): int(size / 2) + 1]

# plot the data
plt.imshow(n_intensity)
plt.colorbar()
plt.show()

# save the data
np.savetxt('/route to save the data.csv', n_intensity, delimiter=',')
```

The following figure shows example outputs of different TDM angles.

0 degree	45 degrees	90 degrees
		

A.1.2 Scripts of Transition Dipole Moment Angle-Fitting Calculation

The following scripts show to calculate the TDM best-fit angle by Python based on the discussion in chapter 3.

Before calculation, the range of the data points on the emission pattern used for TDM best-fit angle calculation (`radius`), the refractive index of the three layers in the system (`n1, n2, n3`), and the wavelength of the laser used to excite the emissive material(`wave_length_nm`), the thickness of the emission layer (`thickness_nm`), and the fit factor (`ff`) need to be determined first.

```
import cmath
from cmath import pi, sin, exp
import numpy as np
import pandas as pd

data = pd.read_csv('/route to open the data.csv', header=None)
data = np.array(data)

size = len(data)

# ------------------------------------------------------------------------------------------
------------------------------- #
# Generate a mask to get data points within a specific radius on the emission pattern

radius = 1.4    # the radius of the mask


# generate two 2D arrays with values bounded between -1.5~1.5
kx = np.linspace(-1.5, 1.5, num=size, dtype=float)
kx = np.tile(kx, (size, 1))
```

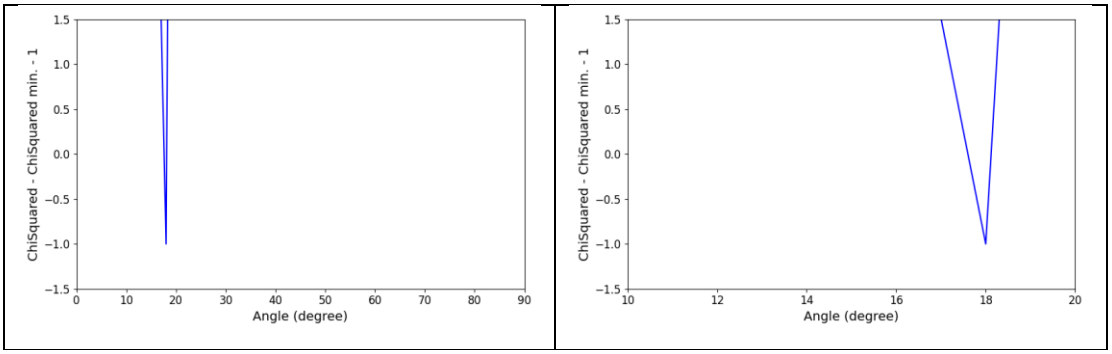
```
ky = np.linspace(-1.5, 1.5, num=size, dtype=float)
ky = ky.reshape((size, 1))
ky = np.tile(ky, (1, size))

# do calculation on kx and ky arrays
res = np.zeros(shape=(size, size))
for i in range(kx.shape[0]):
    for j in range(kx.shape[1]):
        x = kx[i][j]
        y = ky[i][j]
        res[i][j] = np.sqrt(x**2 + y**2)

dataOnCircle = np.where(res > radius, 0, 1)  # numpy.where(condition[, x, y]), if the
condition is true, return x; else, return y

# ------------------------------------------------------------------------------------
------------------------------- #
# TDM angle analysis

wave_length_nm = 430  # wavelength (unit: nm)
wave_length = wave_length_nm * pow(10, -9)

sum_of_chi_squared = []
for i in range(0, 91):
    tdm = pi / 180 * (1 * i + 0)

    def n_intensity(kx, ky):
        n1 = 1  # The refractive index of air
        n2 = 2  # The reflective index of the thin-film/emission layer
        n3 = 1.5    # The reflective index of the substrate
        thickness_nm = 5  # thickness of the thin-film/emission layer (unit: nm)
        D = thickness_nm * pow(10, -9)


        def kz1(kx, ky, wave_length, n1):
            return cmath.sqrt((n1 ** 2) * ((2 * pi / wave_length) ** 2) -
((cmath.sqrt(kx ** 2 + ky ** 2)) ** 2))

        def kz2(kx, ky, wave_length, n2):
            return cmath.sqrt((n2 ** 2) * ((2 * pi / wave_length) ** 2) -
((cmath.sqrt(kx ** 2 + ky ** 2)) ** 2))

        def kz3(kx, ky, wave_length, n3):
            return cmath.sqrt((n3 ** 2) * ((2 * pi / wave_length) ** 2) -
((cmath.sqrt(kx ** 2 + ky ** 2)) ** 2))

        def ts_3_2(kx, ky, wave_length, n2, n3):
            return 2 * kz3(kx, ky, wave_length, n3) / (kz3(kx, ky, wave_length, n3) +
kz2(kx, ky, wave_length, n2))

        def tp_3_2(kx, ky, wave_length, n2, n3):
            return 2 * n3 * n2 * kz3(kx, ky, wave_length, n3) / (
                    n2 ** 2 * kz3(kx, ky, wave_length, n3) + n3 ** 2 * kz2(kx, ky,
wave_length, n2))

        def rs_2_1(kx, ky, wave_length, n1, n2):
            return (kz2(kx, ky, wave_length, n2) - kz1(kx, ky, wave_length, n1)) / (
                    kz2(kx, ky, wave_length, n2) + kz1(kx, ky, wave_length, n1))

        def rs_2_3(kx, ky, wave_length, n2, n3):
            return (kz2(kx, ky, wave_length, n2) - kz3(kx, ky, wave_length, n3)) / (
                    kz2(kx, ky, wave_length, n2) + kz3(kx, ky, wave_length, n3))
```

```
        def rp_2_1(kx, ky, wave_length, n1, n2):
            return (n1 ** 2 * kz2(kx, ky, wave_length, n2) - n2 ** 2 * kz1(kx, ky,
wave_length, n1)) / \
                   (n1 ** 2 * kz2(kx, ky, wave_length, n2) + n2 ** 2 * kz1(kx, ky,
wave_length, n1))

        def rp_2_3(kx, ky, wave_length, n2, n3):
            return (n3 ** 2 * kz2(kx, ky, wave_length, n2) - n2 ** 2 * kz3(kx, ky,
wave_length, n3)) / \
                   (n3 ** 2 * kz2(kx, ky, wave_length, n2) + n2 ** 2 * kz3(kx, ky,
wave_length, n3))

        def rho_s_x(wave_length, D, kx, ky, n1, n2, n3):
            return (1 / (8 * pi * (2 * pi / wave_length) ** 2)) * (
                    (2 * pi / wave_length) / kz3(kx, ky, wave_length, n3)) * \
                   abs((ts_3_2(kx, ky, wave_length, n2, n3) * exp(1j * kz2(kx, ky,
wave_length, n2) * D / 2) * \
                        (1 + rs_2_1(kx, ky, wave_length, n1, n2) * exp(2j * kz2(kx,
ky, wave_length, n2) * D / 2))) / \
                       (1 - rs_2_1(kx, ky, wave_length, n1, n2) * rs_2_3(kx, ky,
wave_length, n2, n3) * \
                        exp(2j * kz2(kx, ky, wave_length, n2) * D)) * (ky /
cmath.sqrt(kx ** 2 + ky ** 2))) ** 2

        def rho_p_x(wave_length, D, kx, ky, n1, n2, n3):
            return (1 / (8 * pi * (2 * pi / wave_length) ** 2)) * (
                    (2 * pi / wave_length) / kz3(kx, ky, wave_length, n3)) \
                   * abs((tp_3_2(kx, ky, wave_length, n2, n3) * exp(1j * kz2(kx, ky,
wave_length, n2) * D / 2) * \
                          kz2(kx, ky, wave_length, n2) / (n2 * (2 * pi / wave_length))
* \
                          (1 - rp_2_1(kx, ky, wave_length, n1, n2) * exp(2j * kz2(kx,
ky, wave_length, n2) * D / 2))) / \
                         (1 - rp_2_1(kx, ky, wave_length, n1, n2) * rp_2_3(kx, ky,
wave_length, n2, n3) * \
                          exp(2j * kz2(kx, ky, wave_length, n2) * D)) * (kx /
cmath.sqrt(kx ** 2 + ky ** 2))) ** 2

        def rho_p_z(wave_length, D, kx, ky, n1, n2, n3):
            return (1 / (8 * pi * (2 * pi / wave_length) ** 2)) * (
                    (2 * pi / wave_length) / kz3(kx, ky, wave_length, n3)) * \
                   abs((tp_3_2(kx, ky, wave_length, n2, n3) * exp(1j * kz2(kx, ky,
wave_length, n2) * D / 2) * kx / (
                           n2 * (2 * pi / wave_length)) * \
                        (1 + rp_2_1(kx, ky, wave_length, n1, n2) * exp(2j * kz2(kx,
ky, wave_length, n2) * D / 2))) / \
                       (1 - rp_2_1(kx, ky, wave_length, n1, n2) * rp_2_3(kx, ky,
wave_length, n2, n3) * exp(
                           2j * kz2(kx, ky, wave_length, n2) * D))) ** 2

        return (rho_s_x(wave_length, D, kx, ky, n1, n2, n3) + rho_p_x(wave_length, D,
kx, ky, n1, n2, n3)) * (
                (1 - (sin((tdm)) ** 2)) / 2) + \
               (rho_p_z(wave_length, D, kx, ky, n1, n2, n3)) * ((sin((tdm)) ** 2))


    kxs = np.linspace(-1.5 * (2 * pi / wave_length), 1.5 * (2 * pi / wave_length),
size)
    kys = np.linspace(-1.5 * (2 * pi / wave_length), 1.5 * (2 * pi / wave_length),
size)
    x_1, y_1 = np.meshgrid(kxs, kys)

    arr2D = np.eye(size, size, dtype=complex)
```

```
    for i in range(size):
        for j in range(size):
            arr2D[i][j] = n_intensity(x_1[i][j], y_1[i][j])

    # just get real part
    theory = arr2D.real

    # normalize theory (by the center point of the emission pattern, [kx, ky]=[0, 0])
    theory_n = theory / theory[int(size/2):int(size/2) + 1, int(size/2):int(size/2) +
1]

    # ------------------------------------------------------------------------------
------------------------------- #
    # Calculate chiSquared

    # normalize data (by the center point of the emission pattern, [kx, ky]=[0, 0])
    data_n = data / data[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]

    # ------------------------------------------------------------------------------
------------------------------- #
    # find: data_N_dataOnCircle and theory_N_dataOnCircle (only use the data points
within a specific radius in the emission pattern)
    data_n_dataOnCircle = np.zeros(shape=(size, size))

    for i in range(dataOnCircle.shape[0]):
        for j in range(dataOnCircle.shape[1]):
            x = dataOnCircle[i][j]
            y = data_n[i][j]
            data_n_dataOnCircle[i][j] = (x * y)

    theory_n_dataOnCircle = np.zeros(shape=(size, size))

    for i in range(dataOnCircle.shape[0]):
        for j in range(dataOnCircle.shape[1]):
            x = dataOnCircle[i][j]
            y = theory_n[i][j]
            theory_n_dataOnCircle[i][j] = (x * y)

    # ------------------------------------------------------------------------------
------------------------------- #
    # calculate chiSquared

    ff = 0.98
    chiSquare = ((theory_n_dataOnCircle - data_n_dataOnCircle * ff)**2 /
theory_n_dataOnCircle)

    # turn nan to 0
    chiSquare[np.isnan(chiSquare)] = 0

    sumOfChiSquare = np.sum(chiSquare)

    sum_of_chi_squared.append(sumOfChiSquare)

# ------------------------------------------------------------------------------
------------------------------- #
# Calculate [chi^2 - (chi_min)^2 - 1]

# y_axis = chi^2 - (chi_min)^2 - 1
sum_of_chi_squared = np.asarray(sum_of_chi_squared)
y_axis = sum_of_chi_squared-sum_of_chi_squared.min(axis=0)-1
y_axis = np.asarray(y_axis)

print(y_axis)
```

```
# ------------------------------------------------------------------------------------------
------------------------------- #
# Plot chi2-chi2min-1 figure

import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(8, 5))

x_axis = np.linspace(0, 90, 91)  # (start angle, end angle, the number of angle)
x_axis = np.asarray(x_axis)

ax.plot(x_axis, y_axis, 'b')

ax = plt.gca()
ax.set_xlim([0, 90])    # set the range for x-axis
ax.set_ylim([-1.5, 1.5])     # set the range for y-axis

ax.tick_params(axis="x", labelsize=12)
ax.tick_params(axis="y", labelsize=12)

ax.set_xlabel('Angle (degree)', fontsize=14)
ax.set_ylabel('ChiSquared - ChiSquared min. - 1', fontsize=14)

plt.show()
```

The following figure shows example outputs of TDM angle-fitting.



A.1.3 Scripts of Fit Factor Calculation

The following scripts show how to calculate the fit factor by vertical (ky) cross-section of the emission pattern. Other information about the fit factor is discussed in Chapters 3.2 and 4.4.

```
# Calculate fit factor (ff) based on vertical (ky) cross-section

import numpy as np
import pandas as pd

data = pd.read_csv('/route to open the data.csv', header=None)
```

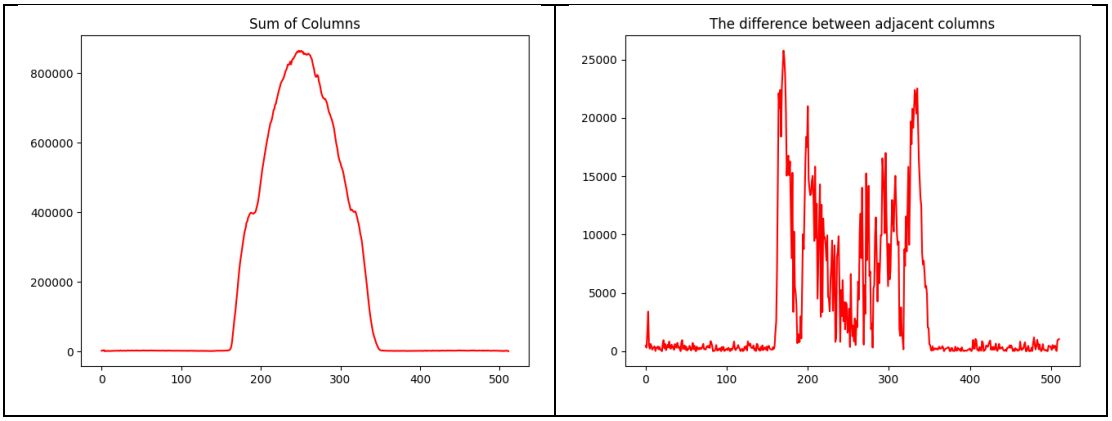
```
data = np.array(data)

size = len(data)

# normalize
data_n = data / data[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]


# find the number of the data point/pixel when k = +- 0.5
b = np.linspace(0, size, num=7)
P_k05 = int(b[4])
N_k05 = int(b[2])


# Get the data points/pixels between (kx=0, ky=-0.5~0.5)
data_05 = data_n[N_k05:P_k05, int(size/2):int(size/2) + 1]  #[row1: row2; column1:
column2]


theory = pd.read_csv('/route to open the data (this data is TDM 0 degree emission
pattern from simulation, which size is the same as experimental data).csv',
header=None)
theory = np.array(theory)
theory_n = theory / theory[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]
# Get the data points/pixels between (kx=0, ky=-0.5~0.5)
theory_05 = theory_n[N_k05:P_k05, int(size/2):int(size/2) + 1]


# ------------------------------------------------------------------------------
-----------------#
# Calculates ChiSquared
sum_of_chi_square = []

for i in range(0, 31):
    chiSquared = ((theory_05 - data_05 * (0.8 + i * 0.01)) ** 2 / theory_05)
    # turn nan to 0
    chiSquared[np.isnan(chiSquared)] = 0

    sumOfChiSquare = np.sum(chiSquared)

    sum_of_chi_square.append(sumOfChiSquare)

sum_of_chi_square = np.asarray(sum_of_chi_square)

# ------------------------------------------------------------------------------
------------------------------- #
# Calculates Sum of [chi^2 - (chi_min)^2 - 1]

# y_axis = chi^2 - (chi_min)^2 - 1
y_axis = sum_of_chi_square-sum_of_chi_square.min(axis=0)-1

print(y_axis)

# turn ndarray into list
y_axis = list(y_axis)

y_axis_min = y_axis.index(-1)

# calculate fit factor
ff = 0.8 + 0.01 * y_axis_min

print(ff)
```

A.1.4 Scripts of Emission Pattern Centering

The following scripts show how to get the corresponding data points/pixels number of the edge of the emission pattern. With this information, the TDM back focal plane imaging can be cut appropriately for the TDM angle analysis.

```
# calculate corresponding pixel number when kx = +-1.5
# 1. Find the sum of each column

import numpy as np
import pandas as pd

data = pd.read_csv('/route to open the data.csv', header=None)
data = np.array(data)

size = len(data)

# normalize
data_n = data / data[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]

# ------------------------------------------------------------------------------------------
------------------------------- #
# calculate the sum of each column
sumOfColumn = data.sum(axis=0)

# plot the sum of each column
import matplotlib.pyplot as plt

x = np.array(np.arange(0, size, 1))
y = sumOfColumn

plt.plot(x, y)
plt.title('Sum of Columns')
plt.show()

# ------------------------------------------------------------------------------------------
------------------------------- #
# calculate the difference between adjacent columns
difference = abs(np.diff(sumOfColumn))

# plot the difference between adjacent columns
X = np.array(np.arange(0, size-1, 1))

plt.plot(X, difference)
plt.title('The difference between adjacent columns')
plt.show()
```

The following figure shows example outputs.



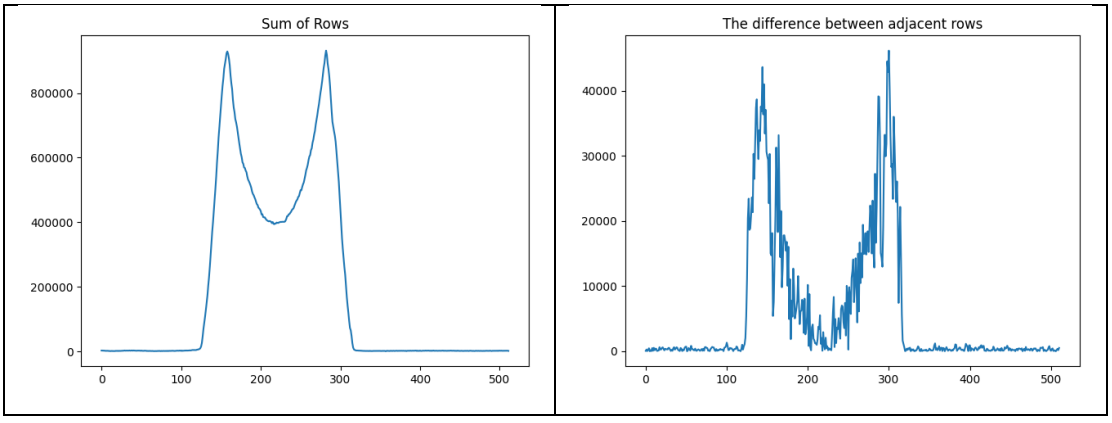

```
# calculate corresponding pixel number when kx = +-1.5
# 2. Find pixel number when the sum of columns increase dramatically

left_threshold = 5000
right_threshold = 5000

# ------------------------------------------------ #
# calculate the corresponding pixel number when kx = +-1.5

# start from the left
for N_kx15 in range(size-1):
    if difference[N_kx15] > left_threshold:
        break

# start from the right
for P_kx15 in reversed(range(size-1)):
    if difference[P_kx15] > right_threshold:
        break


kx0 = int((N_kx15+P_kx15)/2)
print('kx0 = ', kx0)


if ((P_kx15 - N_kx15) % 2) == 0:
   N_kx15 = N_kx15 + 1
else:
   N_kx15 = N_kx15
   P_kx15 = P_kx15

print('N_kx15 = ', N_kx15)
print('P_kx15 = ', P_kx15)

finalSize = P_kx15 - N_kx15 + 1
print('finalSize = ', finalSize)
```

Example output: kxo = 254, N_kx15 = 163, P_kx15 = 346, finalSize = 184

```
# calculate corresponding pixel number when ky = +-1.5
# 1. Find the sum of each row
```

```
import numpy as np
import pandas as pd

data_y = pd.read_csv('/route to open the data.csv', header=None)
data_y = np.array(data_y)

size = len(data_y)

# normalize
data_y_n = data_y / data_y[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]

# ------------------------------------------------------------------------------------------
------------------------------- #
# calculate the sum of each row
sumOfRow = data_y.sum(axis=1)

# plot the sum of each row
import matplotlib.pyplot as plt

a = np.array(np.arange(0, size, 1))
b = sumOfRow

plt.plot(a, b)
plt.title('Sum of Rows')
plt.show()

# ------------------------------------------------------------------------------------------
------------------------------- #
# calculate the difference between adjacent rows
difference_y = abs(np.diff(sumOfRow))

# plot the difference between adjacent rows
A = np.array(np.arange(0, size-1, 1))

plt.plot(A, difference_y)
plt.title('The difference between adjacent rows')
plt.show()
```

The following figure shows example outputs.



```
# calculate corresponding pixel number when ky = +-1.5
# 2. Find pixel number when the sum of rows increase dramatically

left_threshold_y = 10000
right_threshold_y = 10000

# ------------------------------------------------ #
# calculate the corresponding pixel number when ky = +-1.5

# start from the left
for N_ky15 in range(size-1):
    if difference_y[N_ky15] > left_threshold_y:
        break

# start from the right
for P_ky15 in reversed(range(size-1)):
    if difference_y[P_ky15] > right_threshold_y:
        break


ky0 = int((N_ky15+P_ky15)/2)
print('ky0 = ', ky0)

# calculate the pixel number of ky = +- 1.5 by kx = +- 1.5 since the emission pattern
is a circle, the radius should be consistent
P_ky15 = int(ky0 + (P_kx15 - kx0))
N_ky15 = int(ky0 - (kx0 - N_kx15))
finalSize_y = P_ky15 - N_ky15 + 1

print('P_ky15 = ', P_ky15)
print('N_ky15 = ', N_ky15)
print('finalSize_y', finalSize_y)
```

Example output: kyo = 220, P_kx15 = 312, N_kx15 = 129, finalSize_y = 184

A.1.5 Scripts of Background Subtraction

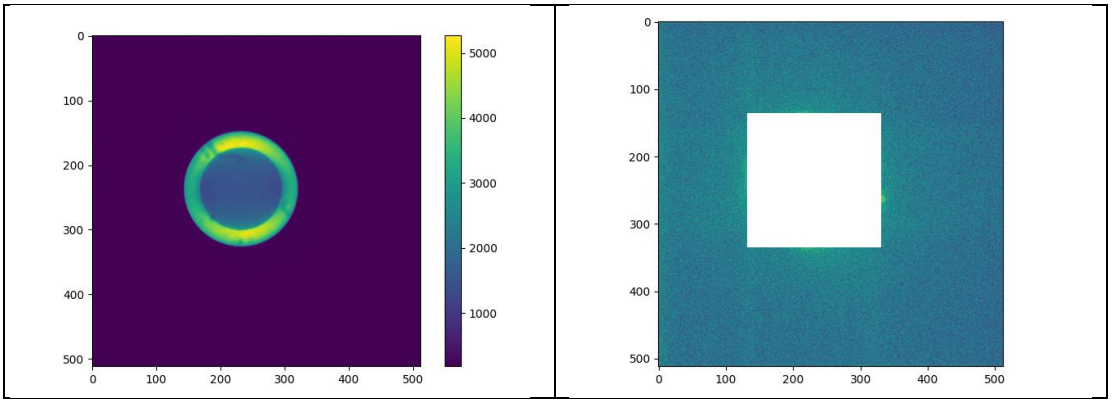
After knowing the edges of a TDM emission pattern, we can do the background subtraction.

1. Remove data points in the square where the TDM emission pattern at
2. Replace the missing data by methods discussed in Chapter 4.3 (linear of column, average of rows, average of columns, griddata linear, griddata nearest, and griddata cubic)
3. Remove the background noise: original data – artificial background noise

```
#%%
# Turn where the emission pattern is into a square with nan
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

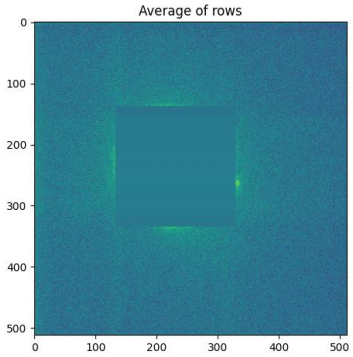
```
data = pd.read_csv('/route to open the data', header=None)
data = np.array(data, dtype=float)  # to use np.NaN, the data type must be float

# enter the edges of the emission pattern
N_ky15 = 148
P_ky15 = 325
N_kx15 = 143
P_kx15 = 320

# turn the range above into NaN
data[N_ky15:P_ky15, N_kx15:P_kx15] = np.NaN

data_nan = data

plt.imshow(data_nan)
plt.show()
```

The following figure shows example outputs.



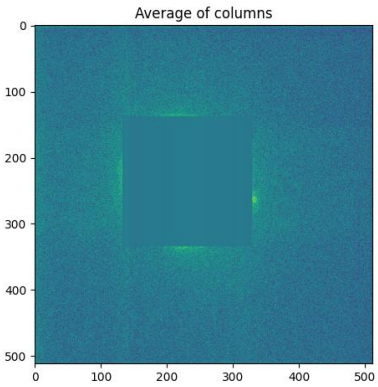

```
#%%
# Background Subtraction: replace nan values with "average of rows"
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


# Obtain mean of rows
row_mean = np.nanmean(data_nan, axis=1)

# Find indices that you need to replace
inds = np.where(np.isnan(data_nan))

# Place row means in the indices
data_nan[inds] = np.take(row_mean, inds[0])

plt.title('Average of rows')
plt.imshow(data_nan)
plt.show()
```

The following figure shows example output.




```
#%%
# Background Subtraction: replace nan values with "average of columns"
import numpy as np


# Obtain mean of columns as you need
col_mean = np.nanmean(data_nan, axis=0)

# Find indices that you need to replace
inds = np.where(np.isnan(data_nan))

# Place column means in the indices
data_nan[inds] = np.take(col_mean, inds[1])

plt.title('Average of columns')
plt.imshow(data_nan)
plt.show()
```

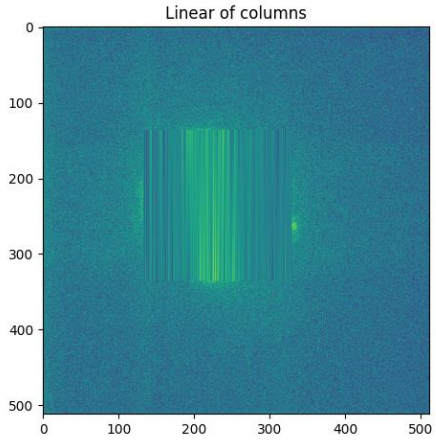
The following figure shows example output.




```
#%%
# Background Subtraction: replace nan values with "linear of columns"
import numpy as np
import matplotlib.pyplot as plt
nan = np.nan

def pad(data):
    bad_indexes = np.isnan(data)
```

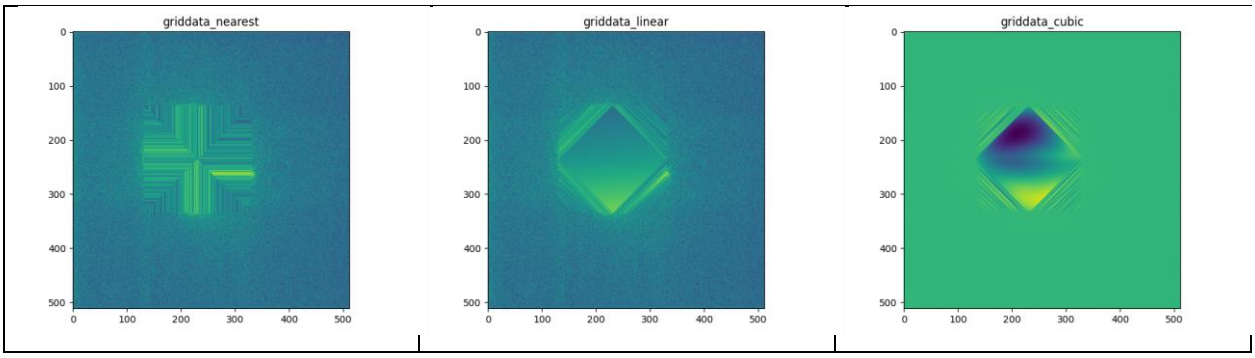
```
    good_indexes = np.logical_not(bad_indexes)
    good_data = data[good_indexes]
    interpolated = np.interp(bad_indexes.nonzero()[0], good_indexes.nonzero()[0],
good_data)
    data[bad_indexes] = interpolated
    return data

data_nan = np.apply_along_axis(pad, 0, data_nan)

plt.title('Linear of columns')
plt.imshow(data_nan)
plt.show()
```

The following figure shows example output.




```
#%%
# Background Subtraction: replace nan values with "griddata: cubic, linear, nearest"
import numpy as np
import matplotlib.pyplot as plt
from scipy import interpolate


x = np.arange(0, data_nan.shape[1])
y = np.arange(0, data_nan.shape[0])

# mask invalid values
data_nan = np.ma.masked_invalid(data_nan)
xx, yy = np.meshgrid(x, y)

# get only the valid values
x1 = xx[~data_nan.mask]
y1 = yy[~data_nan.mask]
newarr = data_nan[~data_nan.mask]

GD1 = interpolate.griddata((x1, y1), newarr.ravel(), (xx, yy), method='linear')  #
method: cubic, linear, nearest

plt.title('griddata_cubic')
plt.imshow(GD1)
plt.show()
```

The following figures show example output.



A.1.6 Scripts of Mimetic Experimental Transition Dipole Moment Emission Pattern

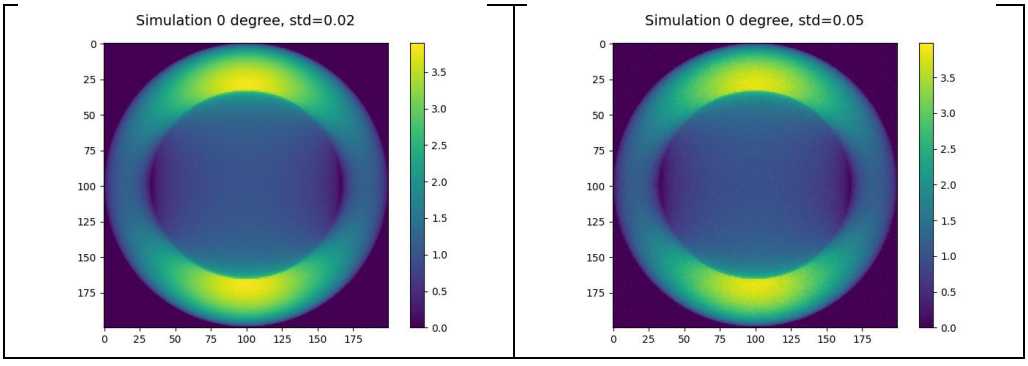
The following scripts show how to smooth and add noise to the theoretical TDM emission pattern. After smoothing and adding noise to the theoretical data, we can compare the artificial data with the experimental data to check how are they similar to each other.

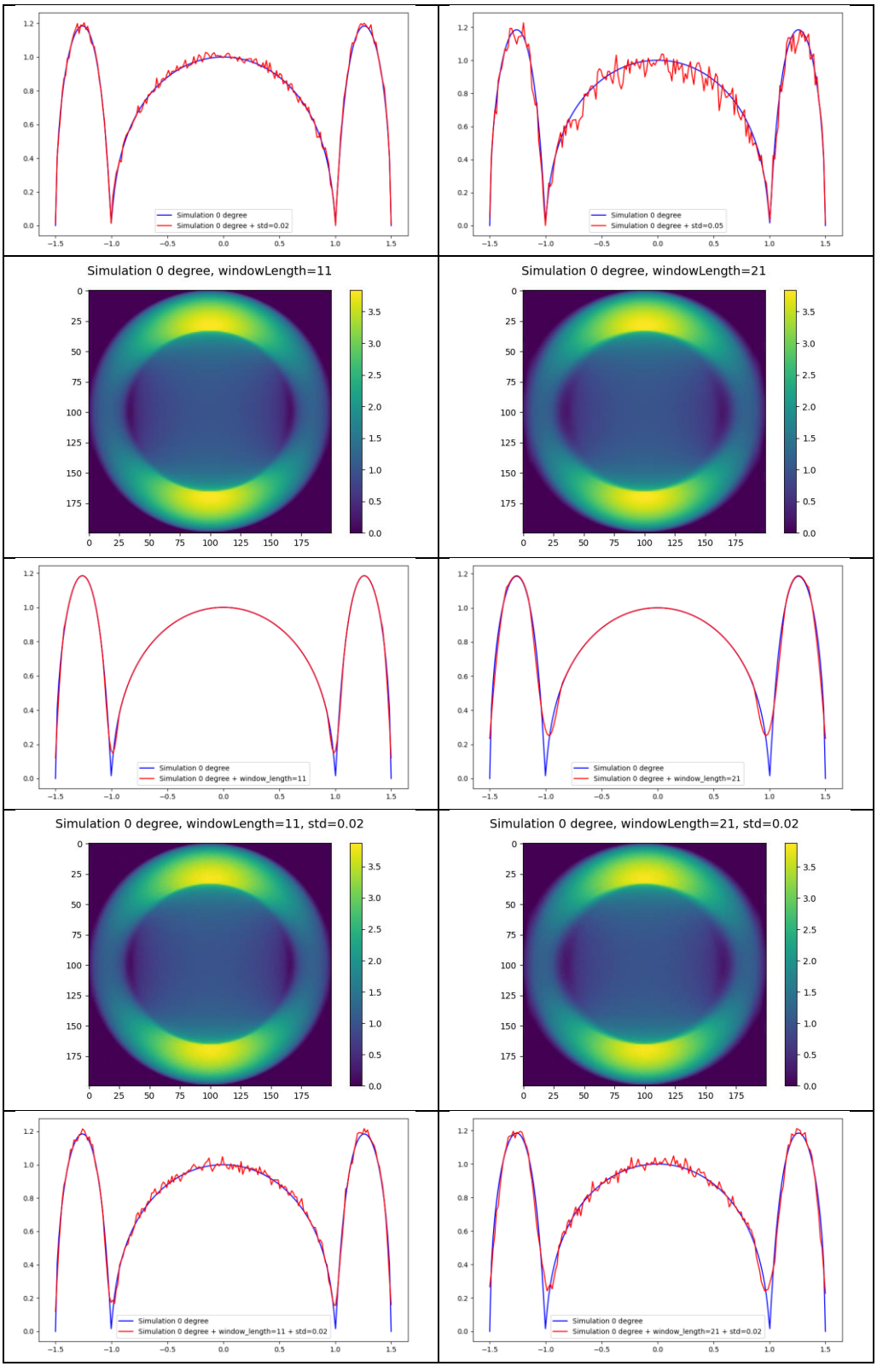
```
#%%
# theoretical data --> smooth data --> add noise

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# input the theoretical data
data = pd.read_csv('/route to open the data', header=None)
data = np.array(data)

size = len(data)

# ------------------------------------------------------------------------------------------
------------------------------ #
# smooth the data

smooth = []
for row_position in range(0, size, 1):

    # Getting data from rows
    y = data[[row_position], :]
    y = y.flatten()

    x = np.linspace(-1.5, 1.5, size)  # x axis
    x = np.array(x)

    # ------------------------------------------- #
    # smoothing: applies the Savitzky-Golay filter

    from scipy.signal import savgol_filter

    # tune the degree of smoothing by windowLength
```

```
    windowLength = 21

    y_new = savgol_filter(y, windowLength, 2)  # (y, window_length [need to be odd],
polyorder), window_length must be less than or equal to the size of y

    smooth.append(y_new)

smooth = np.asarray(smooth)

# ------------------------------------------------------------------------------------
------------------------------- #
# adding noise

# normalize data
smooth = smooth / smooth[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]

#  tune the degree of noise by mean and std values
mean = 0.0   # Mean ("centre") of the distribution.
std = 0.02    # Standard deviation (spread or "width") of the distribution.
noisy_smooth = smooth + np.random.normal(loc=mean, scale=std, size=smooth.shape)
noisy_smooth_clipped = np.clip(noisy_smooth, 0, 255)  # we might get out of bounds due
to noise, so we need to clip data

plt.imshow(noisy_smooth_clipped)
plt.show()

#%%
# calculate standard deviation

thData = pd.read_csv('/route to open the data', header=None)
thData = np.array(thData)

size = len(thData)
row_position = int(size/2)


# noramlize
thData = thData / thData[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]


# get data from row
# thData = thData[[row_position], 62:124]     # kx = +- 0.5
# thData = thData[[row_position], 31:155]     # kx = +- 1.0
thData = thData[[row_position], :]     # all regions
thData = np.transpose(thData)

np.std(thData, dtype=np.float64, axis=0)    # row: axis=0; column: axis=1

#%%
# calculate the standard deviation

import pandas as pd
import numpy as np

thData = pd.read_csv('/route to open the data', header=None)
thData = np.array(thData)

size = len(thData)

# enter the cross-section that you want to calculate the standard deviation
row_position = int(size/2)  # For example: calculate the cross-section of the emission
pattern in kx-direction (ky=0)
```

```
# noramlize
thData = thData / thData[int(size/2): int(size/2) + 1, int(size/2): int(size/2) + 1]


# get data from rows
thData = thData[[row_position], :]
thData = np.transpose(thData)

np.std(thData, dtype=np.float64, axis=0)     # row: axis=0; column: axis=1


#%%
# plot the cross-section of the emission pattern

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


rawData = pd.read_csv('/route to open the data', header=None)
rawData = np.array(rawData)

size = len(rawData)

# Get data from rows --> choose the row position
row_position = int(size/2)

# Get data from rows
rawData = rawData[[row_position], :]
rawData = np.transpose(rawData)


# Plot
fig, ax = plt.subplots(figsize=(8, 5))

x = np.linspace(-1.5, 1.5, size)
x = np.asarray(x)

plt.plot(x, rawData, 'b-')

plt.show()
```

The following figures show example output.



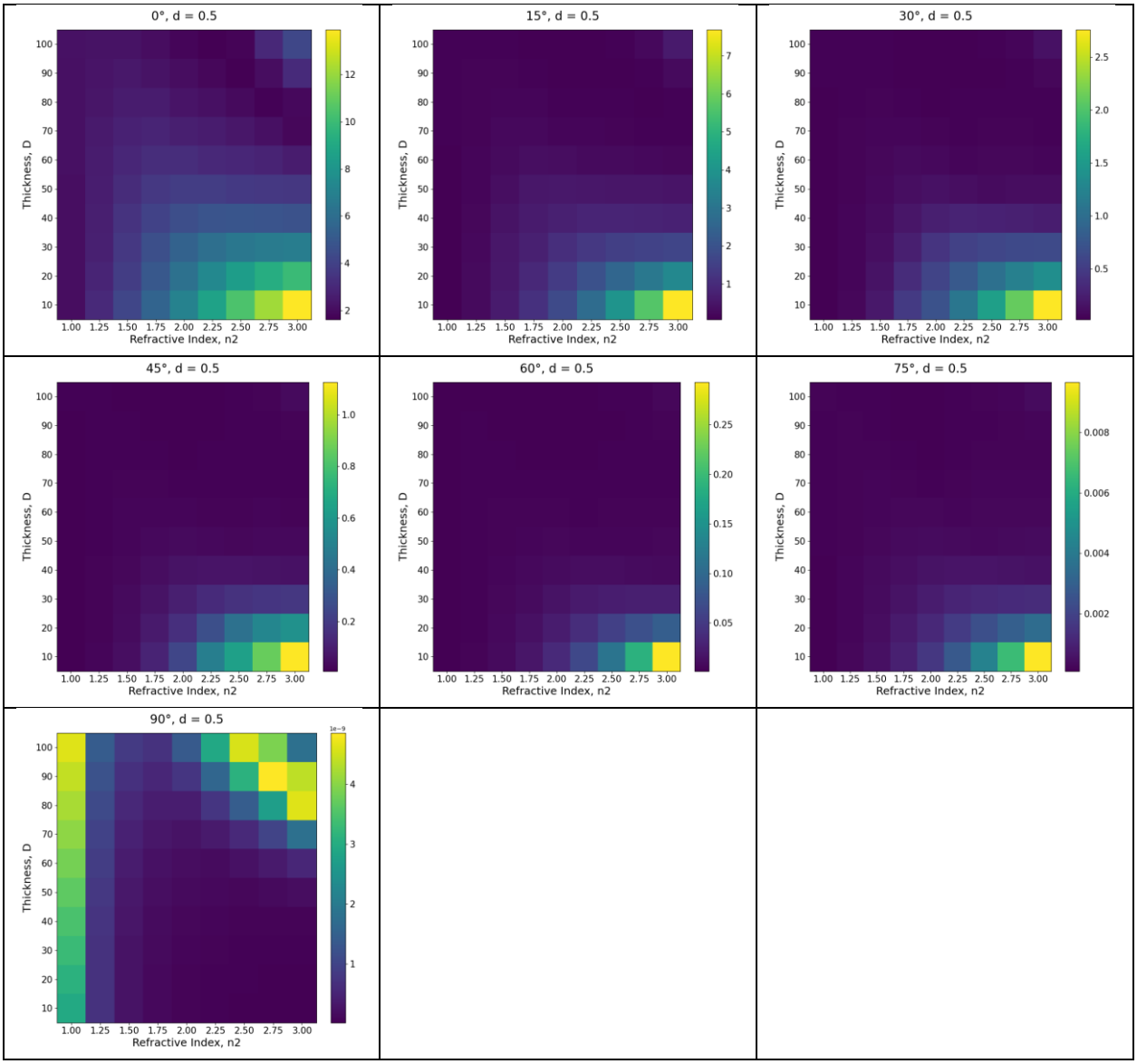

Simulation 0 degree
Simulation 0 degree + std=0.02
Simulation 0 degree
Simulation 0 degree + std=0.05
Simulation 0 degree, windowLength=11
Simulation 0 degree, windowLength=21
Simulation 0 degree
Simulation 0 degree + window_length=11
Simulation 0 degree
Simulation 0 degree + window_length=21
Simulation 0 degree, windowLength=11, std=0.02
Simulation 0 degree, windowLength=21, std=0.02
Simulation 0 degree
Simulation 0 degree + window_length=11 + std=0.02
Simulation 0 degree
Simulation 0 degree + window_length=21 + std=0.02

A.2 Maximum Uncertainty of Fitting TDM Angle Plots

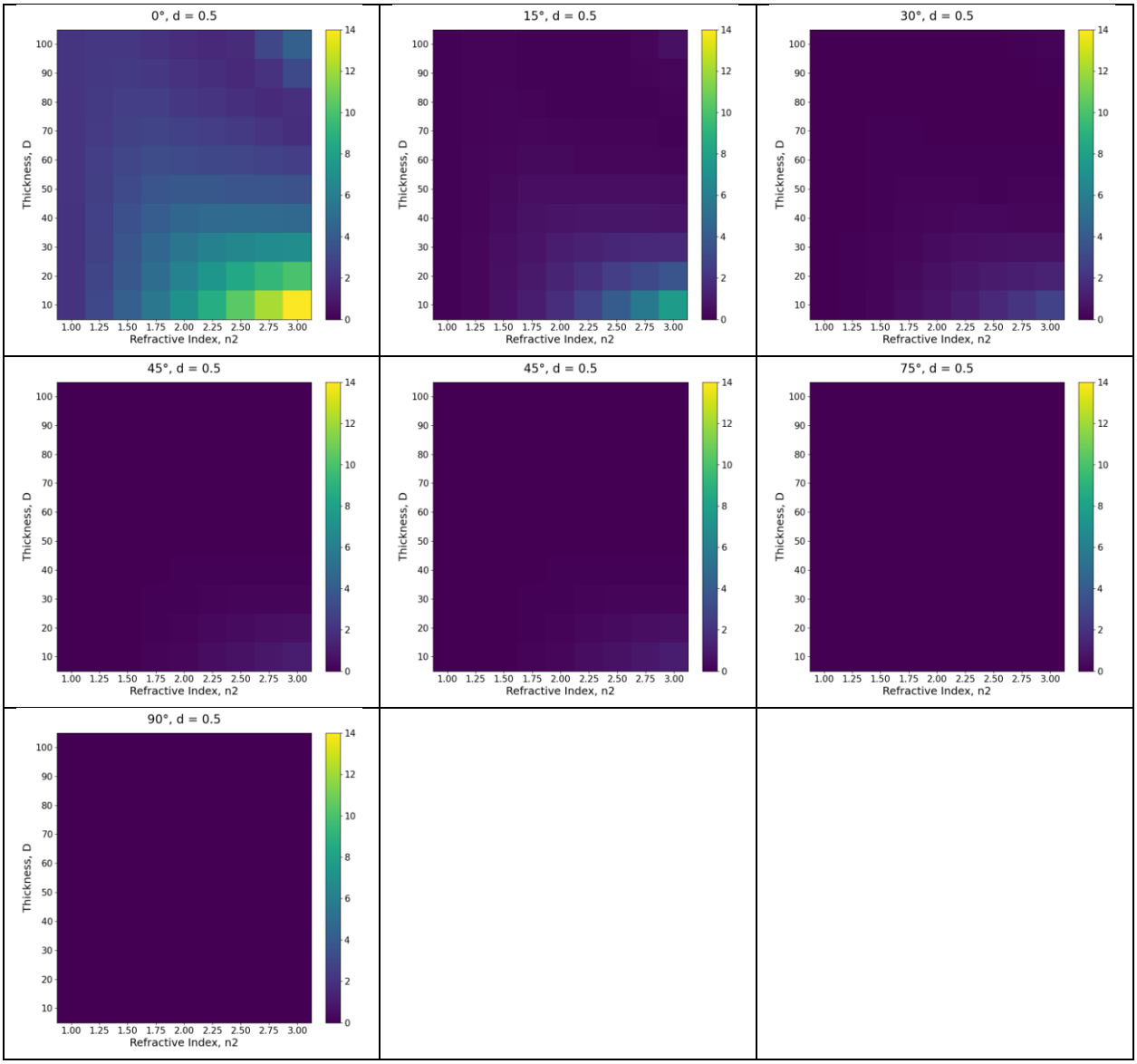
The following figures show the maximum uncertainty of angle-fitting outcomes at certain TDM angles. The parameter setting is n_1=1, n_3=1.5, wavelength=500nm, size=200X200.

A.2.1 Single Transition Dipole Moment at the Middle of the Emission Layer (d=0.5):

Findings: High refractive index and thin emission layer lead to high uncertainty of the TDM angle-fitting result.



Plots with the same color scale:



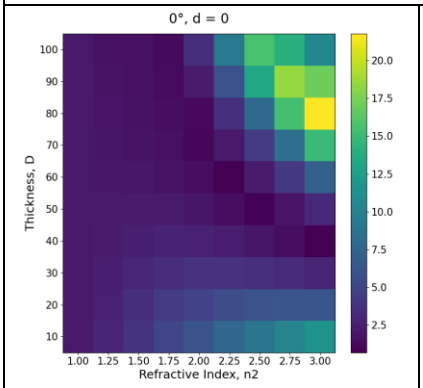
A.2.2 Single Transition Dipole Moment at Different Positions of the Emission Layer (various d):

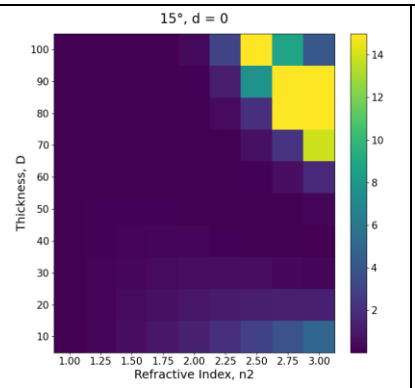
Findings:

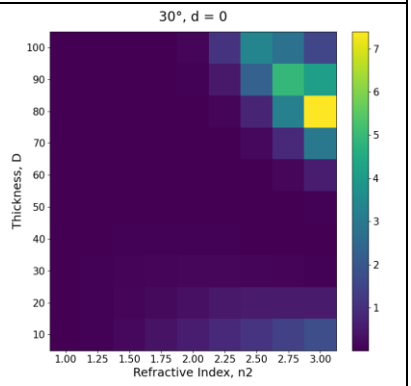
1. When d becomes larger (close to the substrate), the boundary is more blur.
2. The lowest uncertainty appears when d=0.5 (dipole is in the middle of the film).

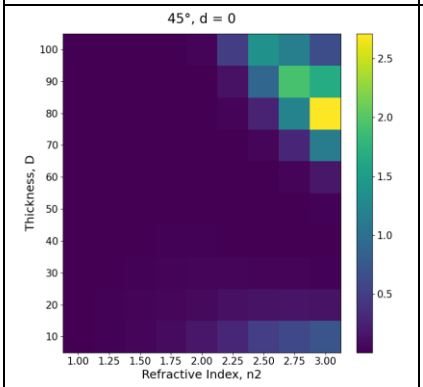
3. 0 degree + d=0 (dipole lies on the top of the film, between air and film) has the biggest uncertainty.

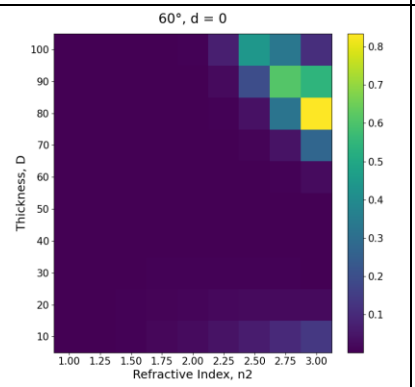
d = 0

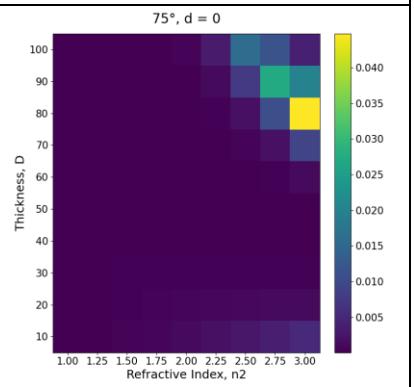


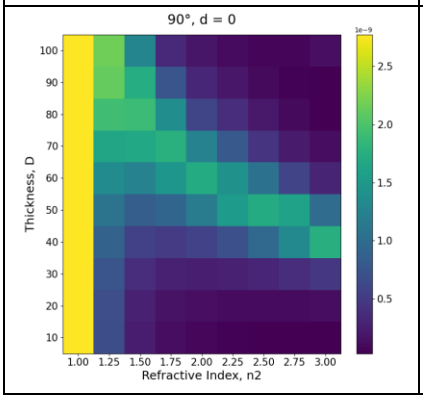




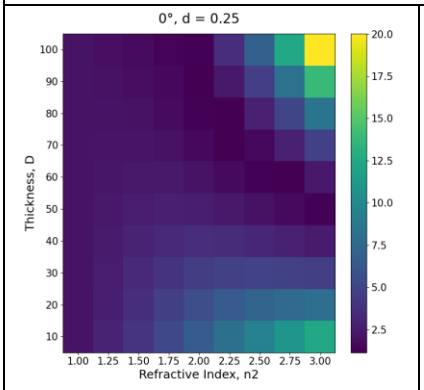


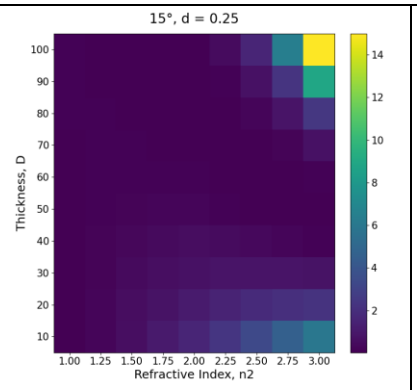


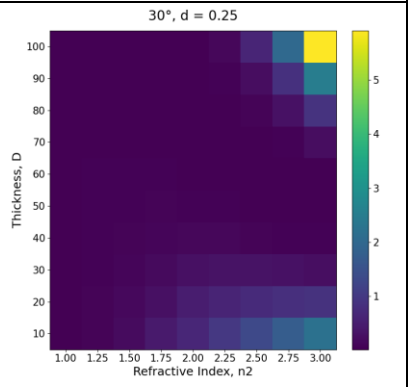


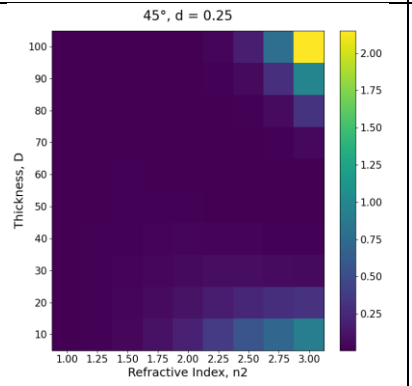


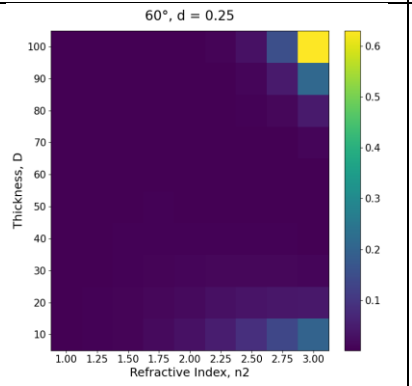
d = 0.25

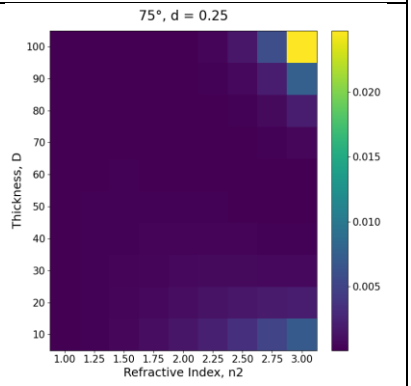

0°, d = 0.25
Thickness, D
Refractive Index, n2

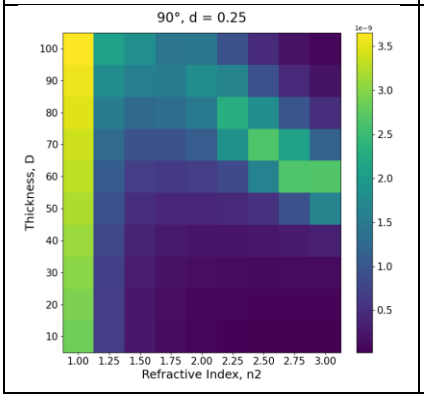

15°, d = 0.25
Thickness, D
Refractive Index, n2


30°, d = 0.25
Thickness, D
Refractive Index, n2

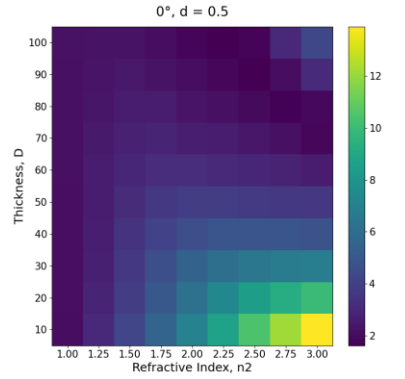

45°, d = 0.25
Thickness, D
Refractive Index, n2

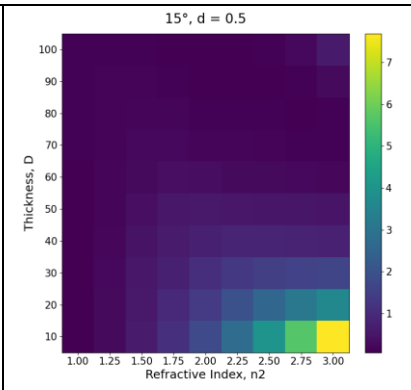

60°, d = 0.25
Thickness, D
Refractive Index, n2

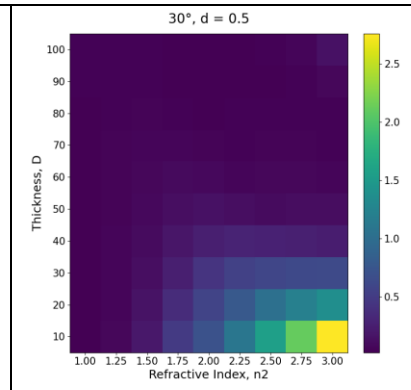

75°, d = 0.25
Thickness, D
Refractive Index, n2

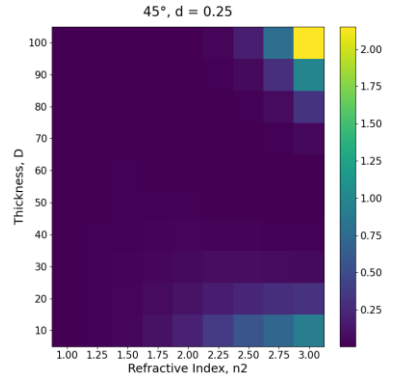

90°, d = 0.25
Thickness, D
Refractive Index, n2

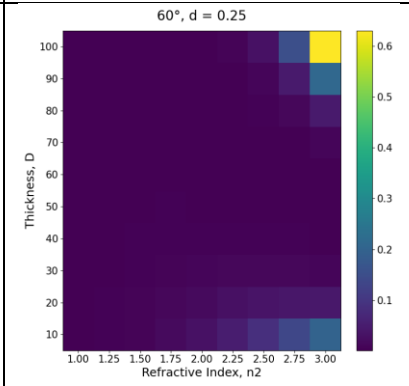
d = 0.5

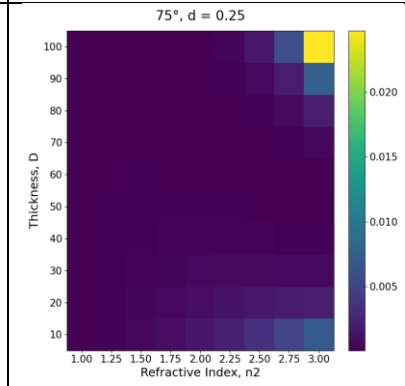

0°, d = 0.5
Thickness, D
Refractive Index, n2

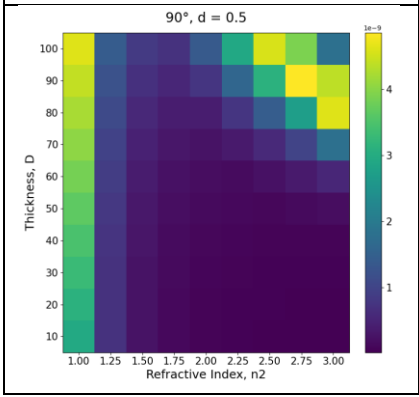

15°, d = 0.5
Thickness, D
Refractive Index, n2


30°, d = 0.5
Thickness, D
Refractive Index, n2

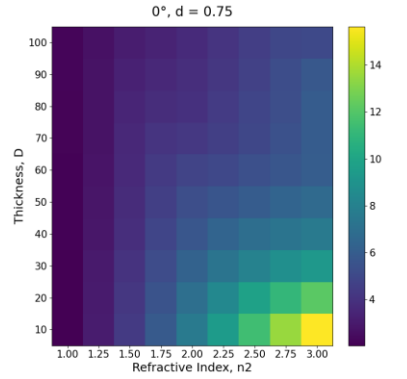

45°, d = 0.25
Thickness, D
Refractive Index, n2

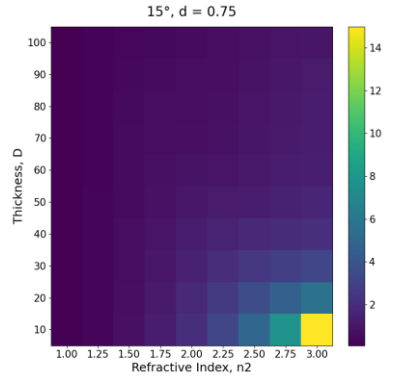

60°, d = 0.25
Thickness, D
Refractive Index, n2

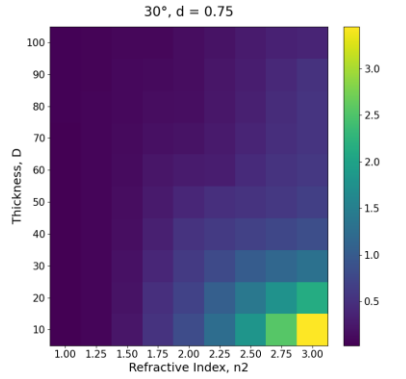

75°, d = 0.25
Thickness, D
Refractive Index, n2

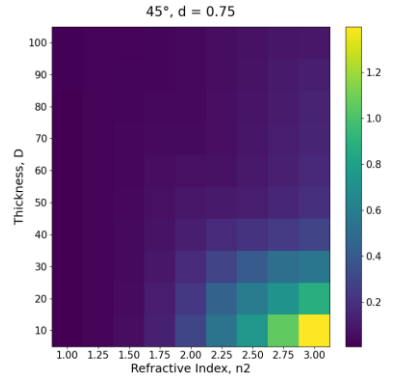

90°, d = 0.5
Thickness, D
Refractive Index, n2

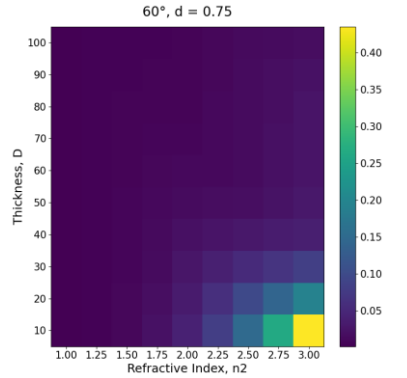
d = 0.75

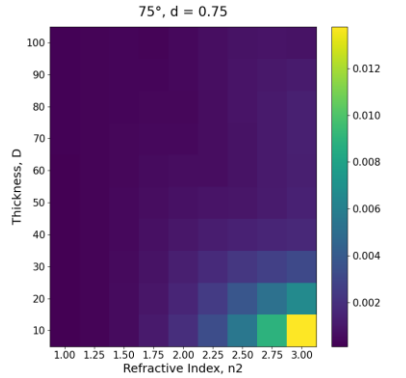

0°, d = 0.75
Thickness, D
Refractive Index, n2

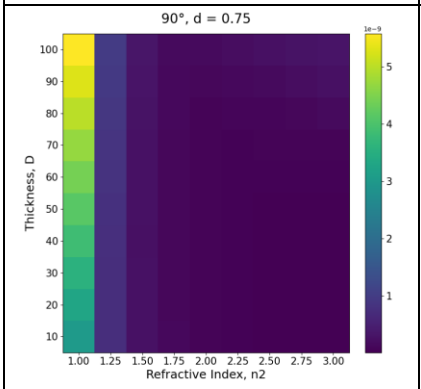

15°, d = 0.75
Thickness, D
Refractive Index, n2


30°, d = 0.75
Thickness, D
Refractive Index, n2

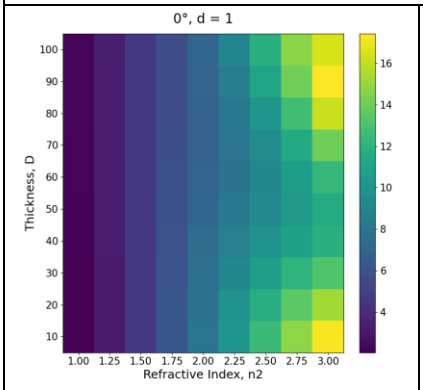

45°, d = 0.75
Thickness, D
Refractive Index, n2

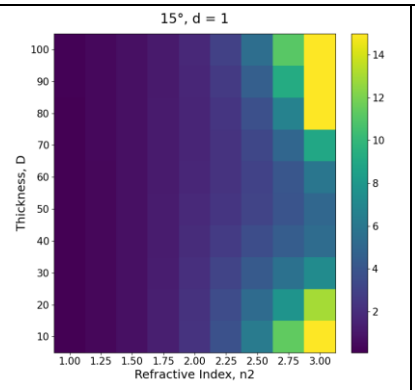

60°, d = 0.75
Thickness, D
Refractive Index, n2

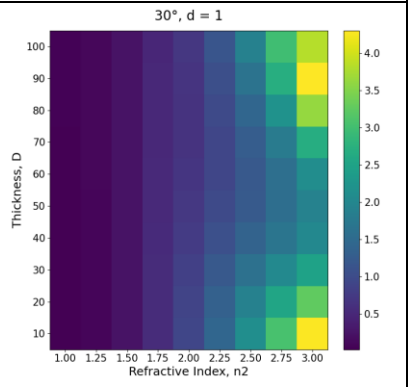

75°, d = 0.75
Thickness, D
Refractive Index, n2

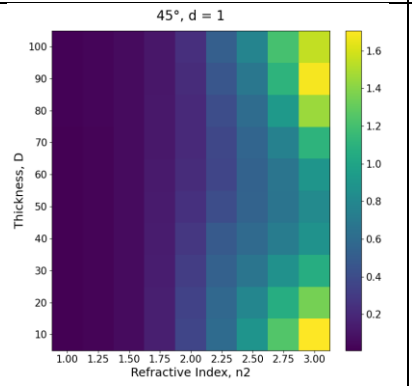

90°, d = 0.75
1e−9
Thickness, D
Refractive Index, n2

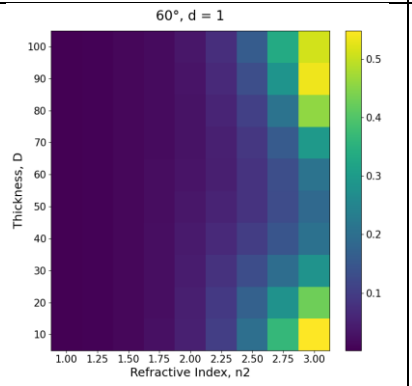
d = 1

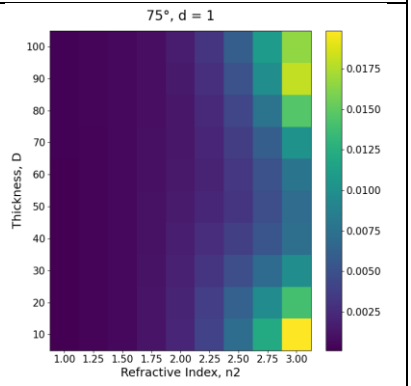

0°, d = 1
Thickness, D
Refractive Index, n2

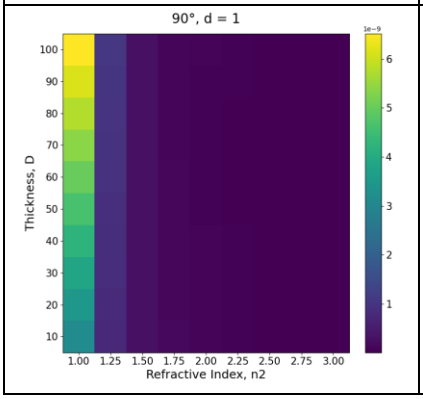

15°, d = 1
Thickness, D
Refractive Index, n2


30°, d = 1
Thickness, D
Refractive Index, n2


45°, d = 1
Thickness, D
Refractive Index, n2

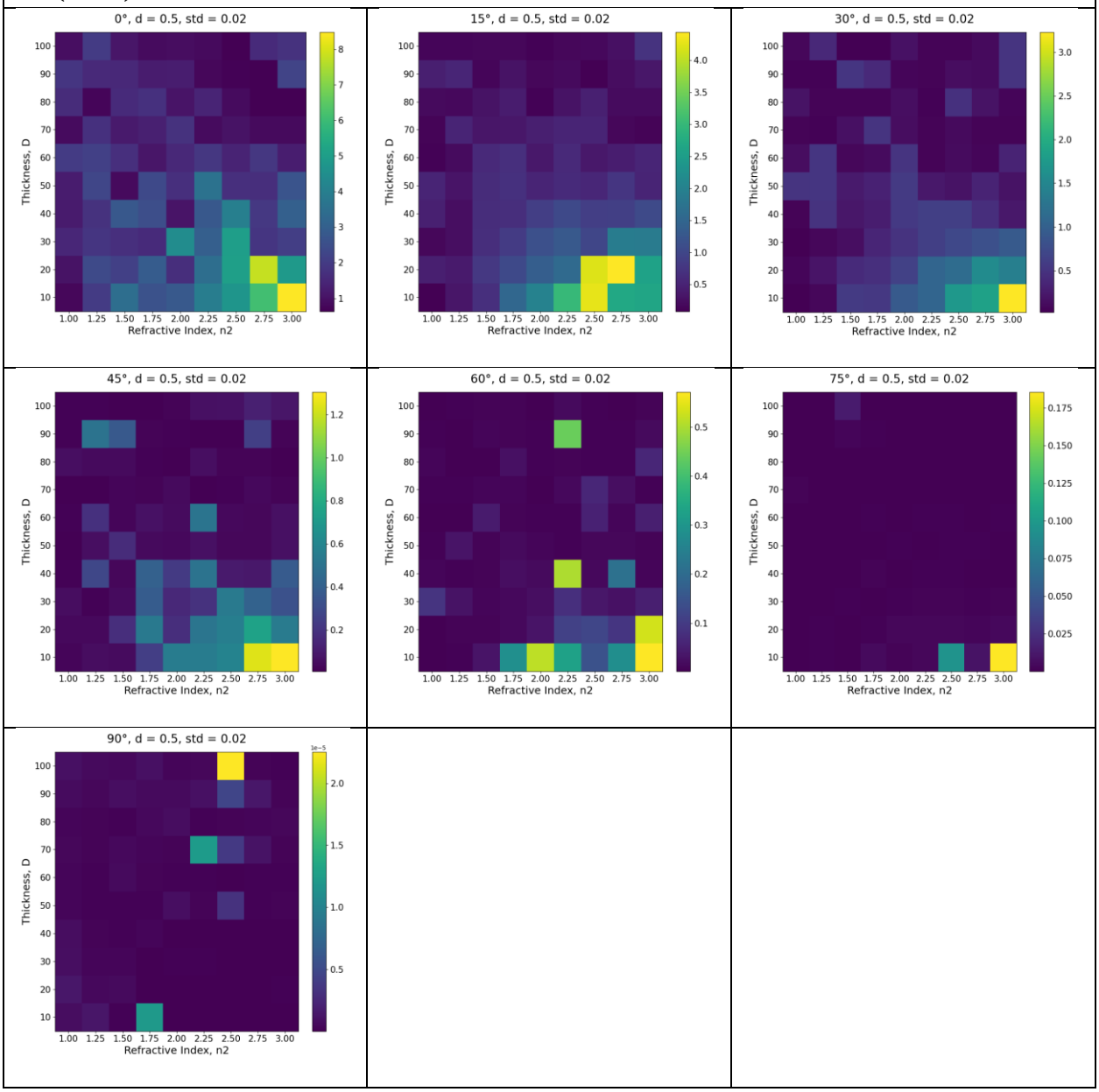

60°, d = 1
Thickness, D
Refractive Index, n2


75°, d = 1
Thickness, D
Refractive Index, n2

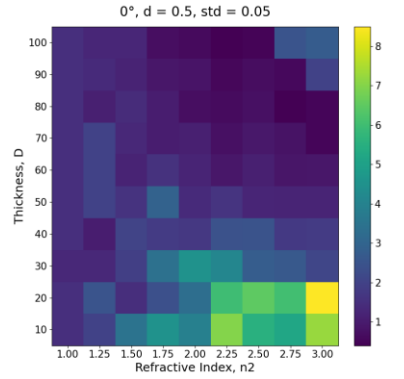

90°, d = 1
Thickness, D
Refractive Index, n2

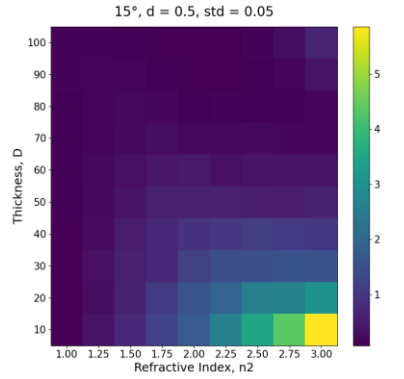
A.2.3 Single Transition Dipole Moment at the Middle of the Emission Layer with Background Noise:

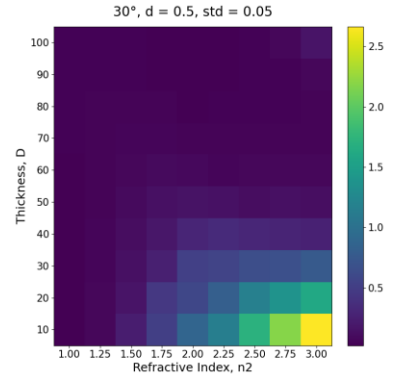
std (noise) = 0.02

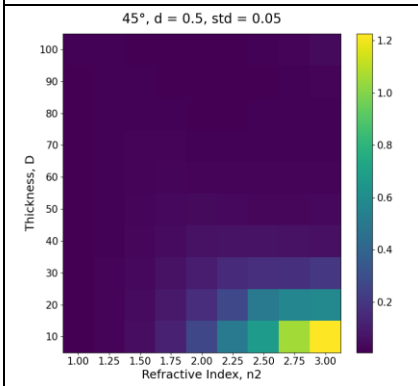


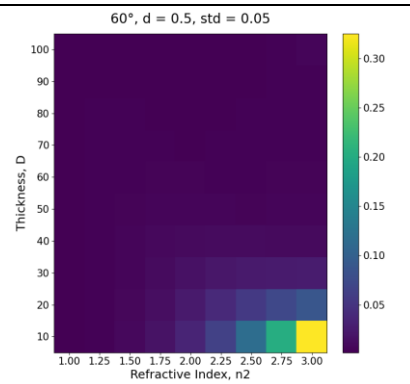
std (noise) = 0.05

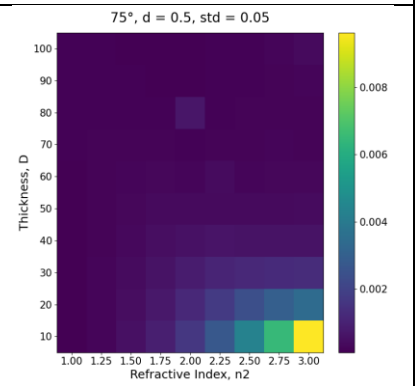


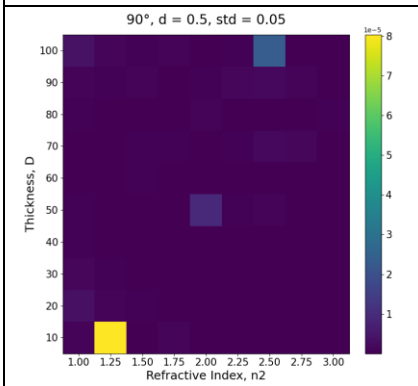




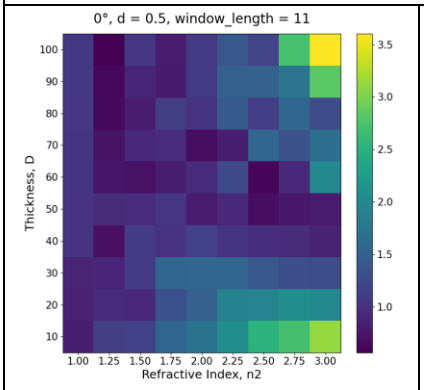


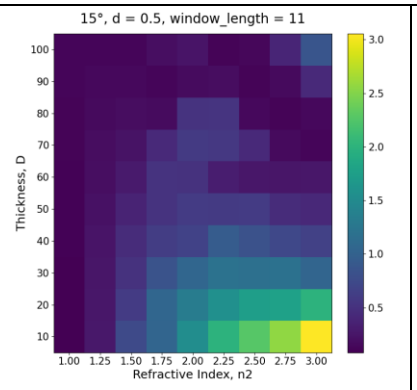


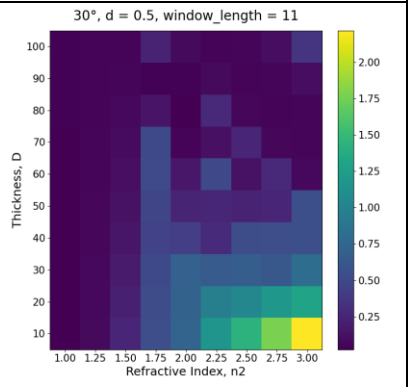


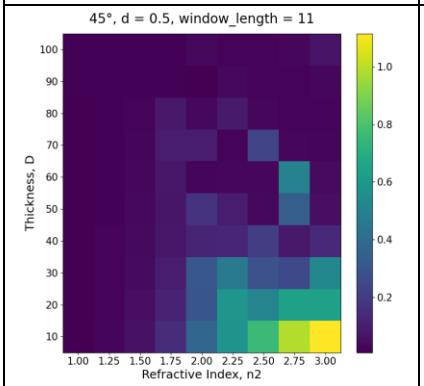


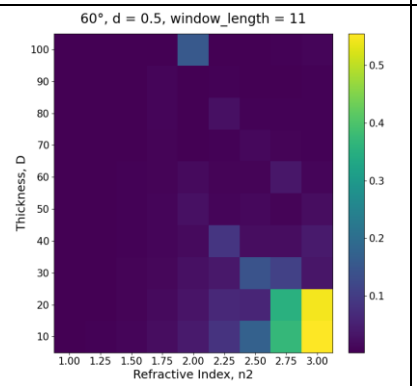
window length (smooth/blur) = 11

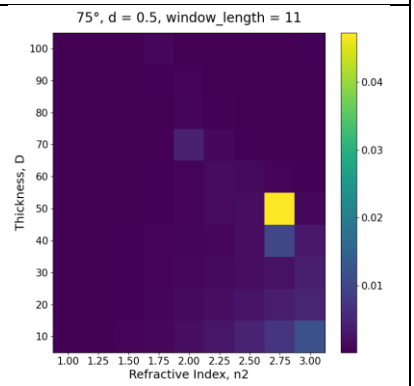


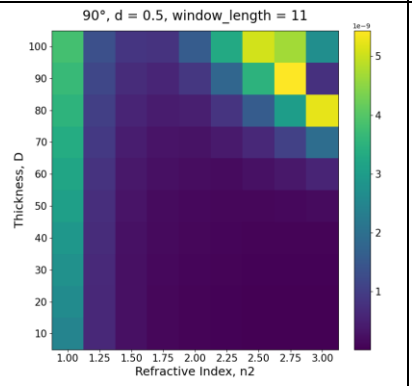




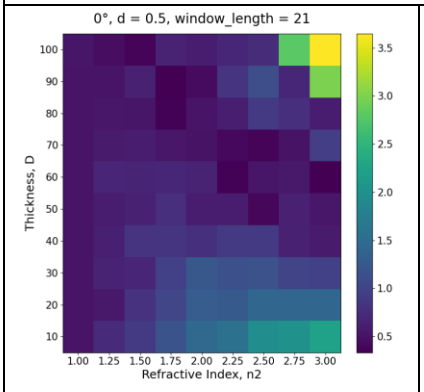


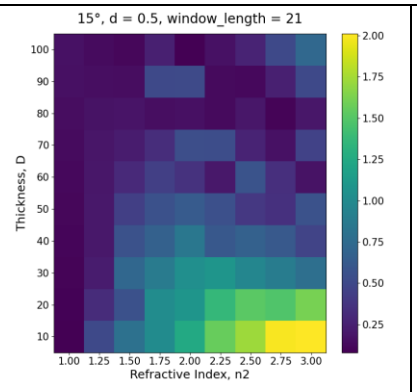


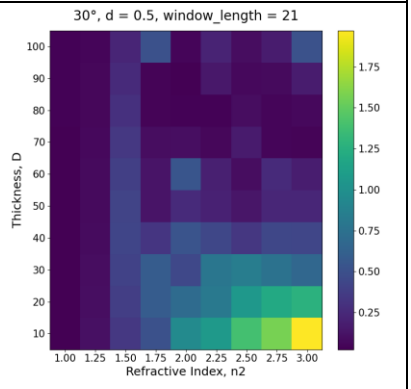


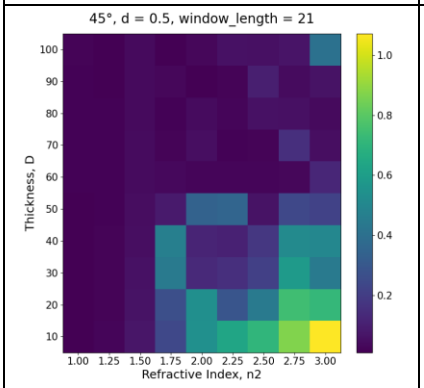


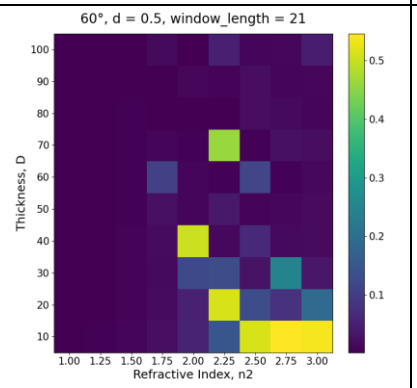
window length (smooth/blur) = 21

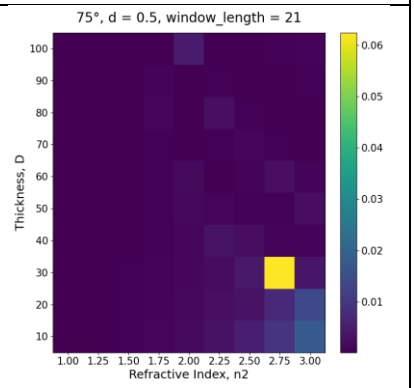


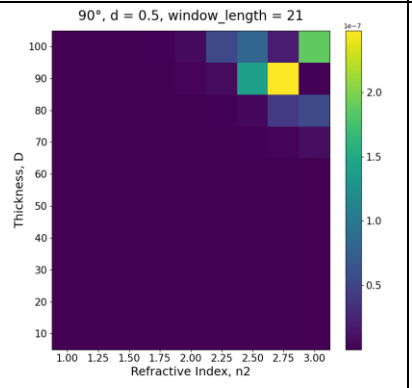




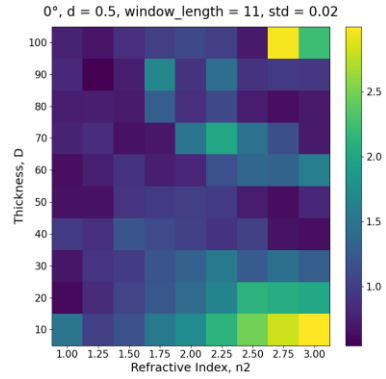


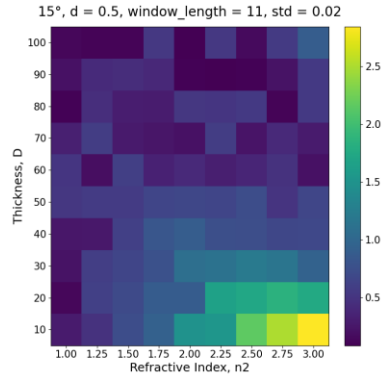


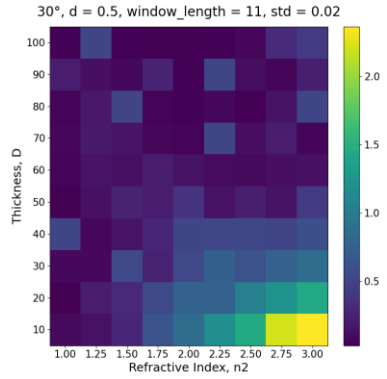


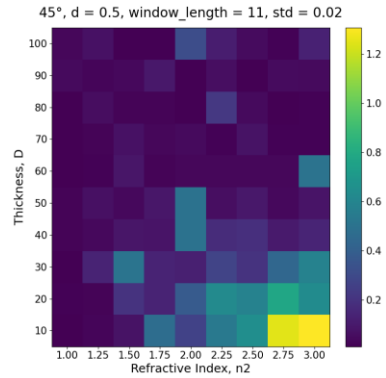


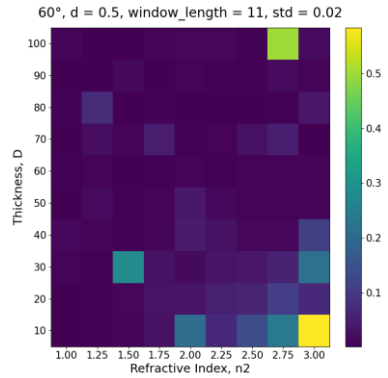
window length (smooth/blur) = 11, std (noise) = 0.02

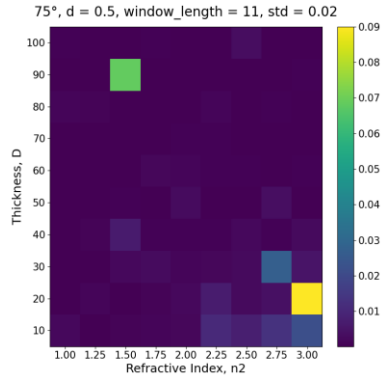


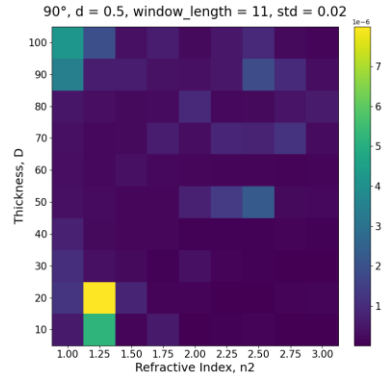




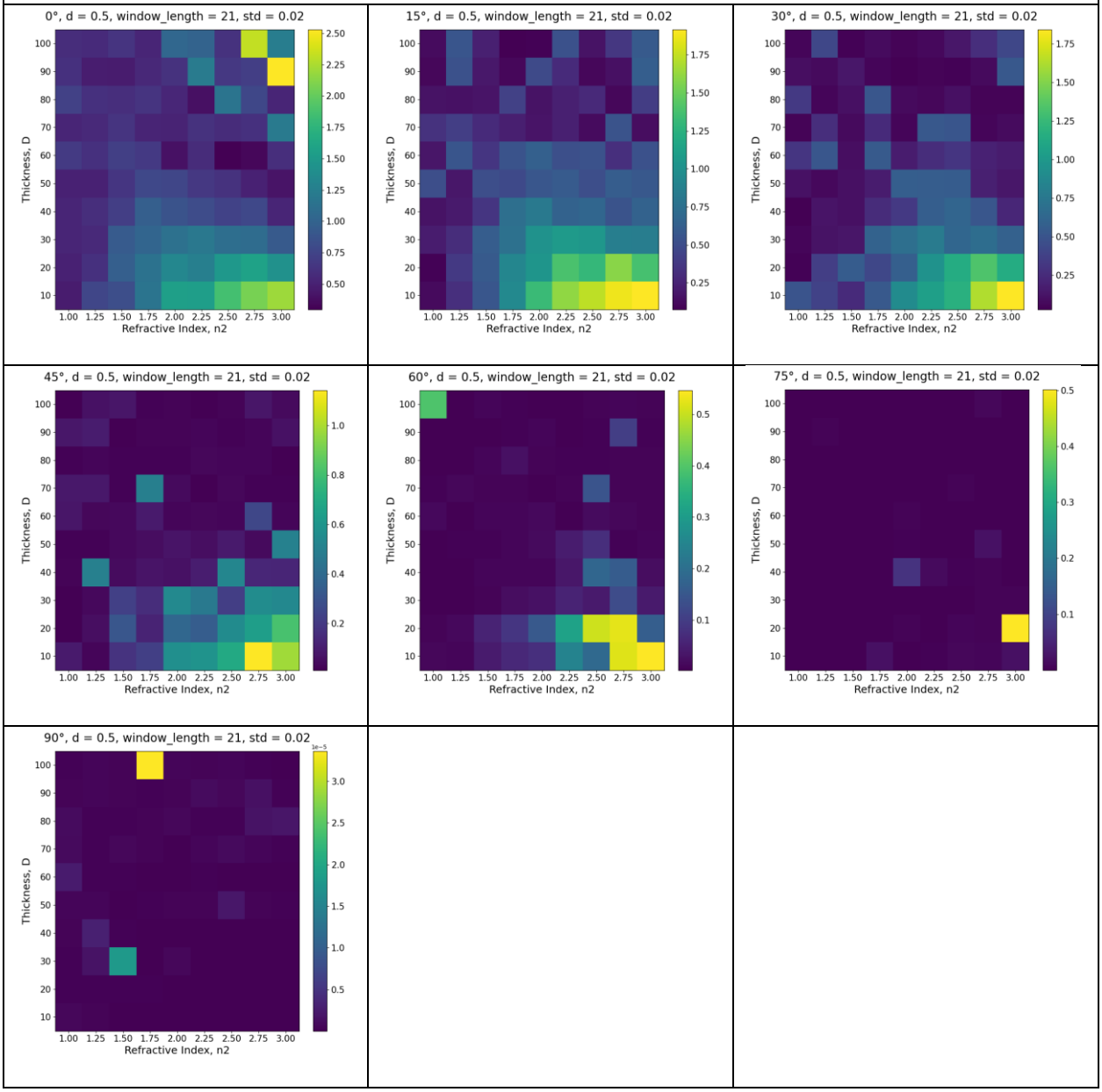








window length (smooth/blur) = 21, std (noise) = 0.02

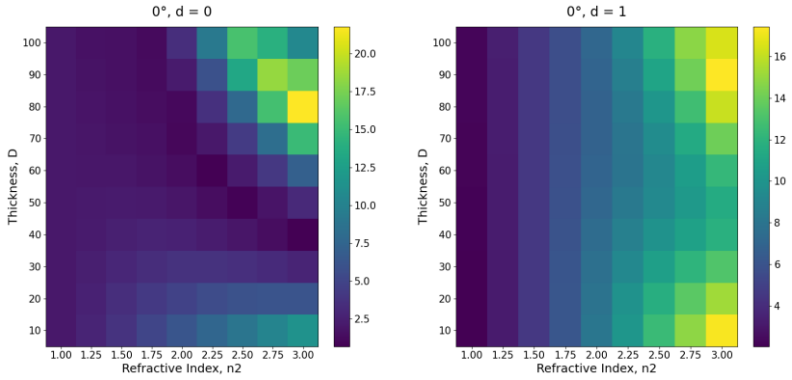


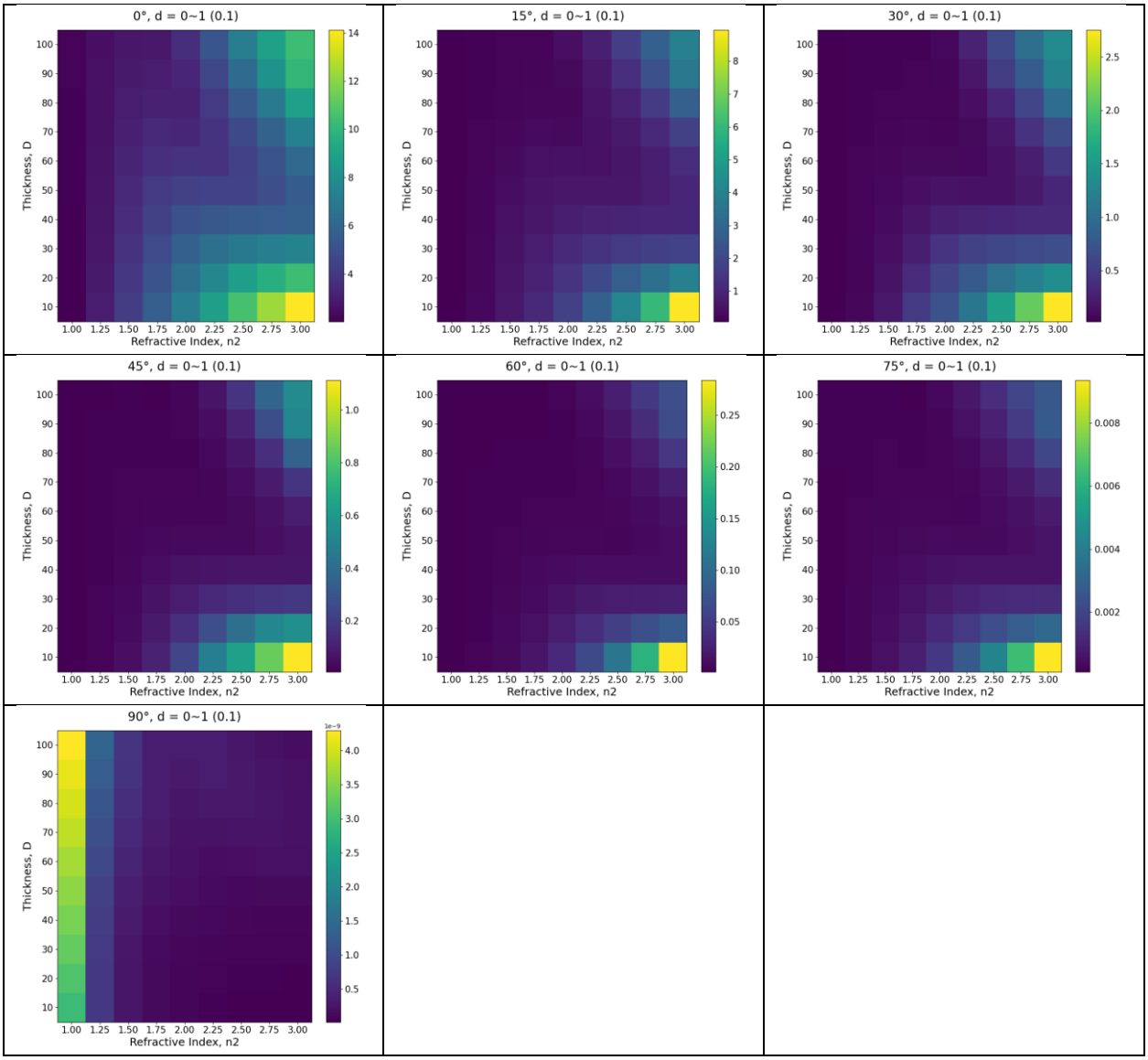
A.2.4 Multiple Transition Dipole Moments (# dipole = 11, distribution: d=0, 0.1, 0.2, …, 1):

Findings:

1. The signal is donated by the dipole which is closer to the bottom (d → 1)
2. d = 1 dominates the pattern of #dipole = 11 (because d = 1 is closer to the BFP, the signal is stronger)

3. d = 0 dominates the diagonal of the pattern





Reference

[01] Jurow, M. J., Lampe, T., Penzo, E., Kang, J., Koc, M. A., Zechel, T., ... & Liu, Y. (2017). Tunable anisotropic photon emission from self-organized CsPbBr3 perovskite nanocrystals. Nano letters, 17(7), 4534-4540.

[02] Eisler, C. (n.d.). EISLER LAB @UCLA - RESEARCH. UCLA. Retrieved May 4, 2021, from https://sites.google.com/g.ucla.edu/eisler-lab/research?authuser=0

[03] U.S. Energy Information Administration. (n.d.). Frequently Asked Questions (FAQs) - U.S. Energy Information Administration (EIA). Retrieved May 4, 2021, from https://www.eia.gov/tools/faqs/faq.php?id=99&t=3

[04] Budde, H., Coca-López, N., Shi, X., Ciesielski, R., Lombardo, A., Yoon, D., ... & Hartschuh, A. (2016). Raman radiation patterns of graphene. ACS nano, 10(2), 1756-1763.

[05] Mulder, C. L., Reusswig, P. D., Velázquez, A. M., Kim, H., Rotschild, C., & Baldo, M. A. (2010). Dye alignment in luminescent solar concentrators: I. Vertical alignment for improved waveguide coupling. Optics Express, 18(101), A79-A90.

[06] Jurow, M. J., Morgenstern, T., Eisler, C., Kang, J., Penzo, E., Do, M., ... & Liu, Y. (2019). Manipulating the transition dipole moment of CsPbBr3 perovskite nanocrystals for superior optical properties. Nano letters, 19(4), 2489-2496.

[07] Cançado, L. G., & Novotny, L. (2016). Observing the angular distribution of Raman scattered fields. ACS nano, 10(2), 1722-1723.

[08] Lieb, M. A., Zavislan, J. M., & Novotny, L. (2004). Single-molecule orientations determined by direct emission pattern imaging. JOSA B, 21(6), 1210-1215.

[09] Taminiau, T. H., Karaveli, S., Van Hulst, N. F., & Zia, R. (2012). Quantifying the magnetic nature of light emission. Nature communications, 3(1), 1-6.

[10] Schuller, J. A., Karaveli, S., Schiros, T., He, K., Yang, S., Kymissis, I., ... & Zia, R. (2013). Orientation of luminescent excitons in layered nanomaterials. Nature nanotechnology, 8(4), 271.

[11] Gao, Y., Weidman, M. C., & Tisdale, W. A. (2017). CdSe nanoplatelet films with controlled orientation of their transition dipole moment. Nano letters, 17(6), 3837-3843.

[12] https://www.edmundoptics.com/knowledge-center/application-notes/optics/introduction-to-polarization/

[13] Venkatanarayanan, A., & Spain, E. (2014). Review of recent developments in sensing materials.

[14] Fellers, T. J., Davidson, M. W. (n.d.). Hamamatsu Learning Center: CCD Noise Sources and Signal-to-Noise Ratio. HSMAMATSU. Retrieved May 4, 2021, from https://hamamatsu.magnet.fsu.edu/articles/ccdsnr.html

[15] Brokmann, X., Coolen, L., Hermier, J. P., & Dahan, M. (2005). Emission properties of single CdSe/ZnS quantum dots close to a dielectric interface. Chemical Physics, 318(1-2), 91-98.