In [25]:
```
import pandas as pd
import matplotlib.pyplot as plt
import glob
from pathlib import Path
from collections import defaultdict
import os
import numpy as np
import csv
```

In [26]:
```
parent_dir = '/home/sethshj/Programs/Materials/Wed1/'
```

In [27]:
```
transition_metals = [
    'Sc', 'Ti', 'V', 'Cr', 'Mn', 'Fe', 'Co', 'Ni', 'Cu', 'Zn'
]

matching_files = []

# Loop through each transition metal and search for files that match the pattern
for metal in transition_metals:
    # Use glob to search for files where the transition metal appears after the und
    pattern = f'{parent_dir}**/**_{metal}/Corvus.cfavg.xes.out'
    matching_files.extend(glob.glob(pattern, recursive=True))

print(matching_files)
```

```
['/home/sethshj/Programs/Materials/Wed1/Y2TiO5/Y2TiO5_Ti/Corvus.cfavg.xes.out', '/ho
me/sethshj/Programs/Materials/Wed1/V2O5/V2O5_V/Corvus.cfavg.xes.out', '/home/sethsh
j/Programs/Materials/Wed1/Cr2O3/Cr2O3_Cr/Corvus.cfavg.xes.out', '/home/sethshj/Progr
ams/Materials/Wed1/CrPbO4/CrPbO4_Cr/Corvus.cfavg.xes.out', '/home/sethshj/Programs/M
aterials/Wed1/LiMnP/LiMnP_Mn/Corvus.cfavg.xes.out', '/home/sethshj/Programs/Material
s/Wed1/MnO/MnO_Mn/Corvus.cfavg.xes.out', '/home/sethshj/Programs/Materials/Wed1/Fe2O
3/Fe2O3_Fe/Corvus.cfavg.xes.out', '/home/sethshj/Programs/Materials/Wed1/CoS2/CoS2_C
o/Corvus.cfavg.xes.out', '/home/sethshj/Programs/Materials/Wed1/NiO/NiO_Ni/Corvus.cf
avg.xes.out', '/home/sethshj/Programs/Materials/Wed1/TmNiC2/TmNiC2_Ni/Corvus.cfavg.x
es.out', '/home/sethshj/Programs/Materials/Wed1/Sr2CuO3/Sr2CuO3_Cu/Corvus.cfavg.xes.
out', '/home/sethshj/Programs/Materials/Wed1/ZnS/ZnS_Zn/Corvus.cfavg.xes.out']
```

In [28]:
```
def xes_integrated_abs_difference(data_1, data_2):
    '''
    Calculate the integrated absolute difference of Δμ(E) for data_1 and data_2.

    Data 1 and 2 have orthogonal polarizations.

    Parameters:
    data_1 (pandas.DataFrame): The DataFrame containing the X-ray absorption data f
    data_2 (pandas.DataFrame): The DataFrame containing the X-ray absorption data f

    Returns:
    difference (float): The integrated absolute difference of Δμ(E) between the two
    '''
    # Calculate the absolute difference of Delta mu(E)
    abs_difference = np.abs(data_1 - data_2)

    # Integrate the absolute difference over the energy range
```

```
    integrated_abs_difference = np.sum(abs_difference)

    return integrated_abs_difference


def xes_average(data_1, data_2, data_3):
    '''
    Calculates the integral of the average of 3 orthogonally polarized XES spectra.

    Parameters:
    data_1 (pandas.DataFrame): The DataFrame containing the X-ray absorption data f
    data_2 (pandas.DataFrame): The DataFrame containing the X-ray absorption data f
    data_3 (pandas.DataFrame): The DataFrame containing the X-ray absorption data f

    Return:
    integrated_average (float): The integral of the average of the 3 spectra.
    '''
    # Calculate the average of Delta mu(E) values
    average_delta_muE = (data_1 + data_2 + data_3) / 3

    # Integrate the average Delta mu(E) over the energy range
    integrated_average = np.sum(average_delta_muE)

    return integrated_average


def anisotropy_parameter(xes_difference, xes_average):
    '''
    Calculate the anisotropy parameter, which is the quotient of the XES difference

    Parameters:
    xes_difference (float): The integrated absolute difference of Δμ(E).
    xes_average (float): The integral of the average of the 3 spectra.

    Returns:
    float: The anisotropy parameter.
    '''
    if xes_average == 0:
        raise ValueError(
            "The xes_average must not be zero to avoid division by zero.")

    return xes_difference / xes_average


def anisotropy_matrix(data_x, data_y, data_z):
    '''
    Calculate a 3x3 anisotropy matrix where each entry represents the anisotropy pa
    for the difference between two datasets divided by the average of all three dat

    Parameters:
    data_x (pandas.DataFrame): The DataFrame containing the X-ray absorption data f
    data_y (pandas.DataFrame): The DataFrame containing the X-ray absorption data f
    data_z (pandas.DataFrame): The DataFrame containing the X-ray absorption data f

    Returns:
    numpy.ndarray: A 3x3 anisotropy matrix.
```

```
    '''
    # Calculate the XES average of all three datasets
    xes_avg = xes_average(data_x, data_y, data_z)

    # Initialize a 3x3 matrix
    anisotropy_mat = np.zeros((3, 3))

    # Define the pairs for which to calculate the differences
    pairs = [(data_x, data_y), (data_x, data_z), (data_y, data_z)]

    # Fill the anisotropy matrix with the anisotropy parameters
    for i, (data1, data2) in enumerate(pairs):
        diff = xes_integrated_abs_difference(data1, data2)
        anisotropy_mat[i][(i+1) % 3] = anisotropy_parameter(diff, xes_avg)
        anisotropy_mat[(i+1) % 3][i] = anisotropy_mat[i][(i+1) %
                                                         3]  # Symmetric entries

    return anisotropy_mat
```

In [29]:

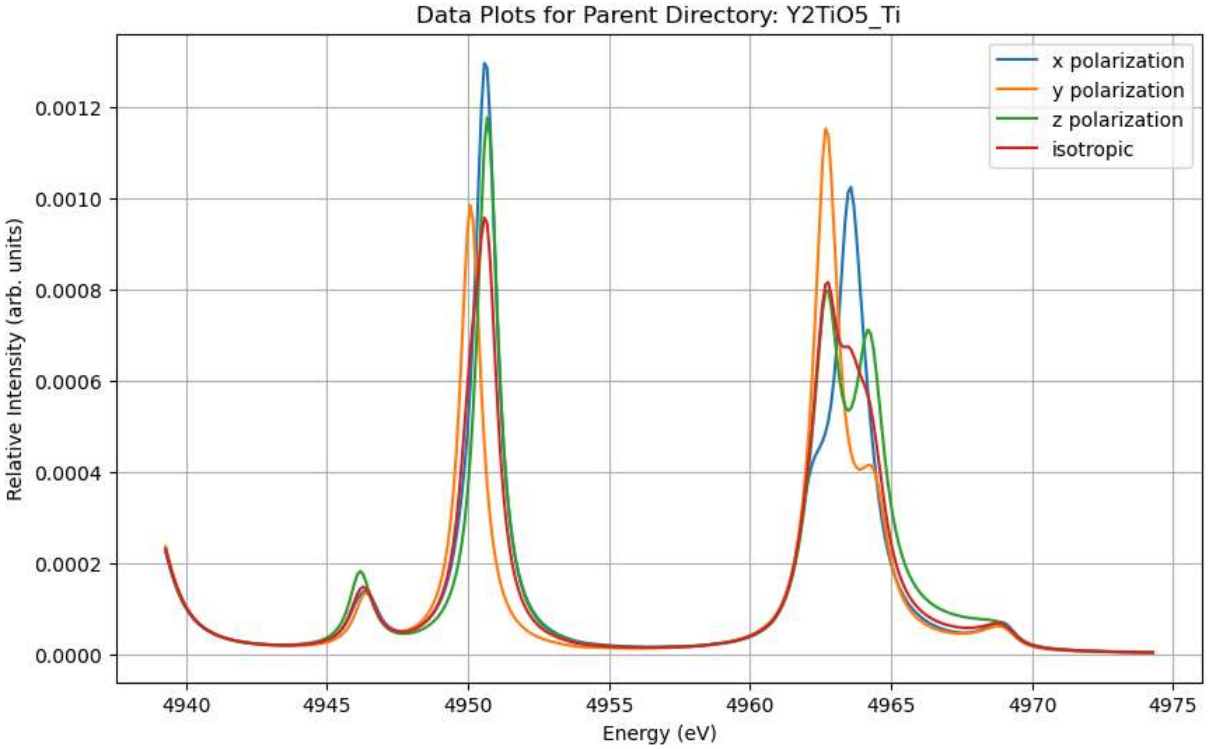
```
def read_data(file):
    return np.loadtxt(file)

output_csv = f'{parent_dir}anisotropy_data.csv'

# Open the CSV file for writing
with open(output_csv, mode='w', newline='') as file_out:
    csv_writer = csv.writer(file_out)

    # Write the header row
    csv_writer.writerow(['parent_dir', 'm00', 'm01', 'm02', 'm10', 'm11', 'm12', 'm

    # Loop through each matching file and process the data
    for file in matching_files:
        data = read_data(file)
        data_x = data[:, 1]
        data_y = data[:, 2]
        data_z = data[:, 3]

        # Calculate the anisotropy matrix (3x3)
        anisotropy_mat = anisotropy_matrix(data_x, data_y, data_z)

        # Get the parent directory name
        parent_dir = os.path.basename(os.path.dirname(file))

        # Flatten the 3x3 matrix into a single row (list)
        flattened_matrix = anisotropy_mat.flatten().tolist()

        # Prepend the parent_dir to the flattened matrix row
        row = [parent_dir] + flattened_matrix

        # Write the row to the CSV
        csv_writer.writerow(row)

print(f"Data has been written to {output_csv}")

# for file in matching_files:
```

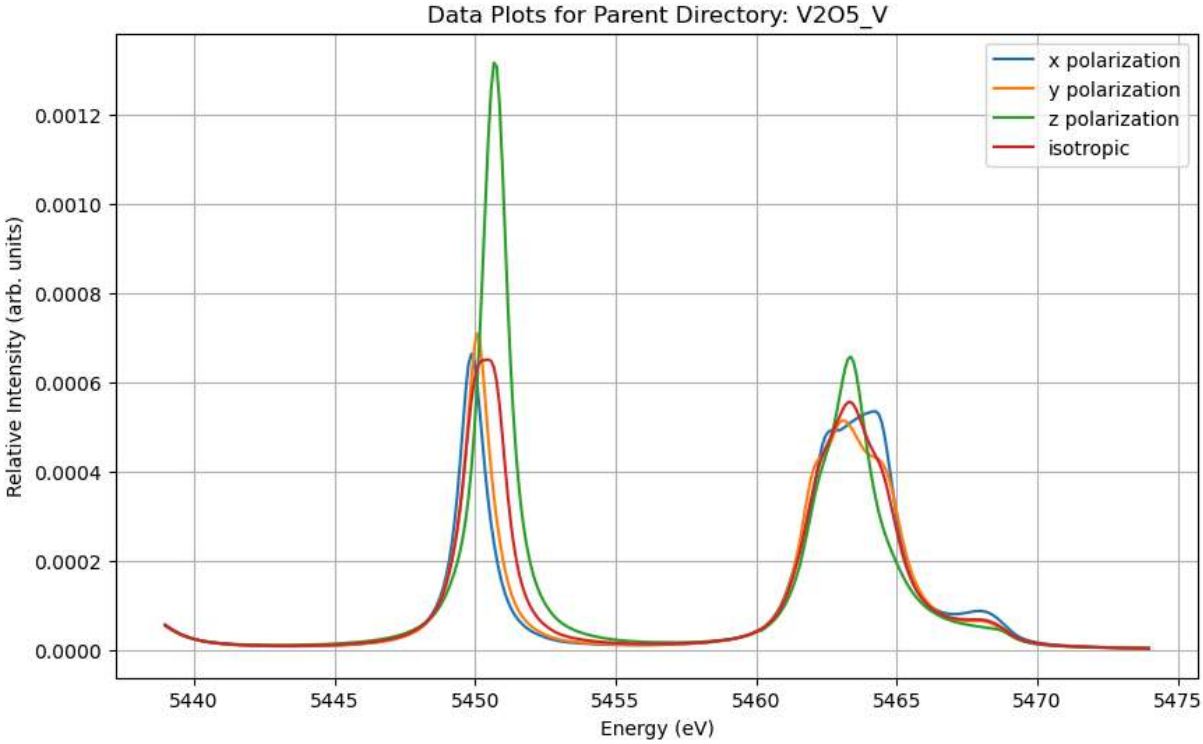
```
#     data = read_data(file)
#     data_x = data[:, 1]
#     data_y = data[:, 2]
#     data_z = data[:, 3]
#     anisotropy_mat = anisotropy_matrix(data_x, data_y, data_z)
#     parent_dir = os.path.basename(os.path.dirname(file))
#     print(anisotropy_mat)
```

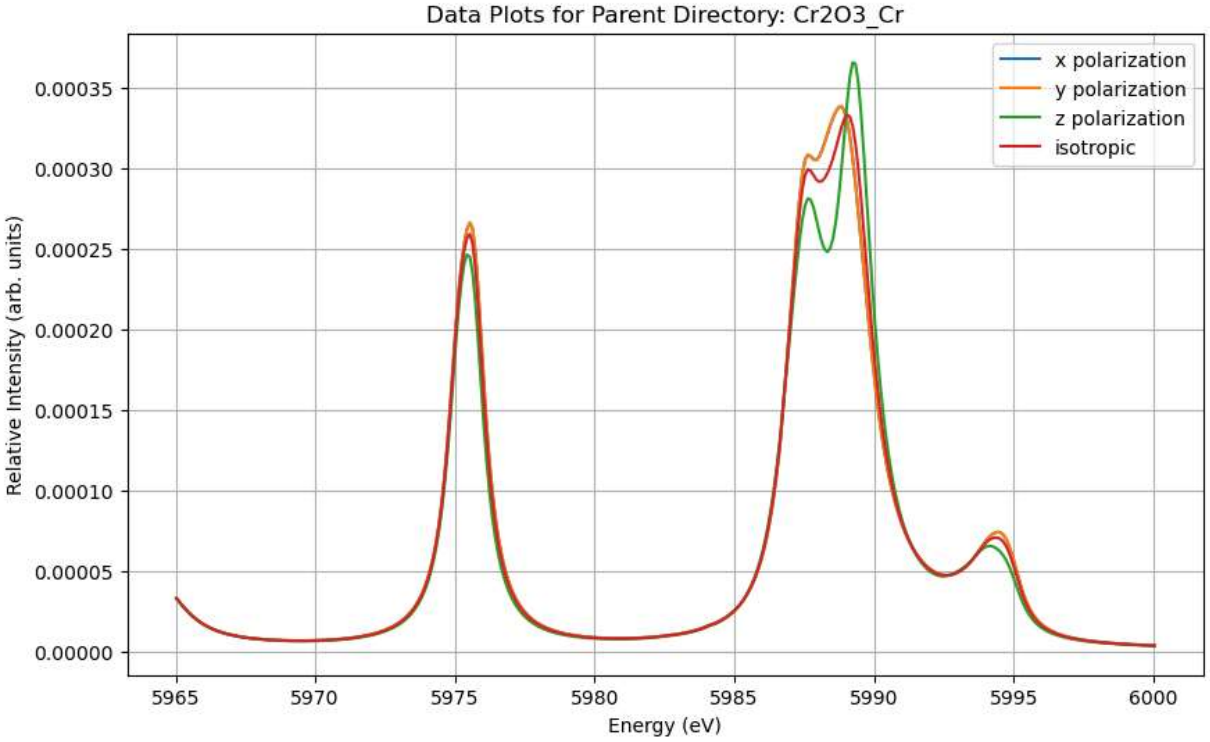
```
Data has been written to /home/sethshj/Programs/Materials/Wed1/anisotropy_data.csv
```

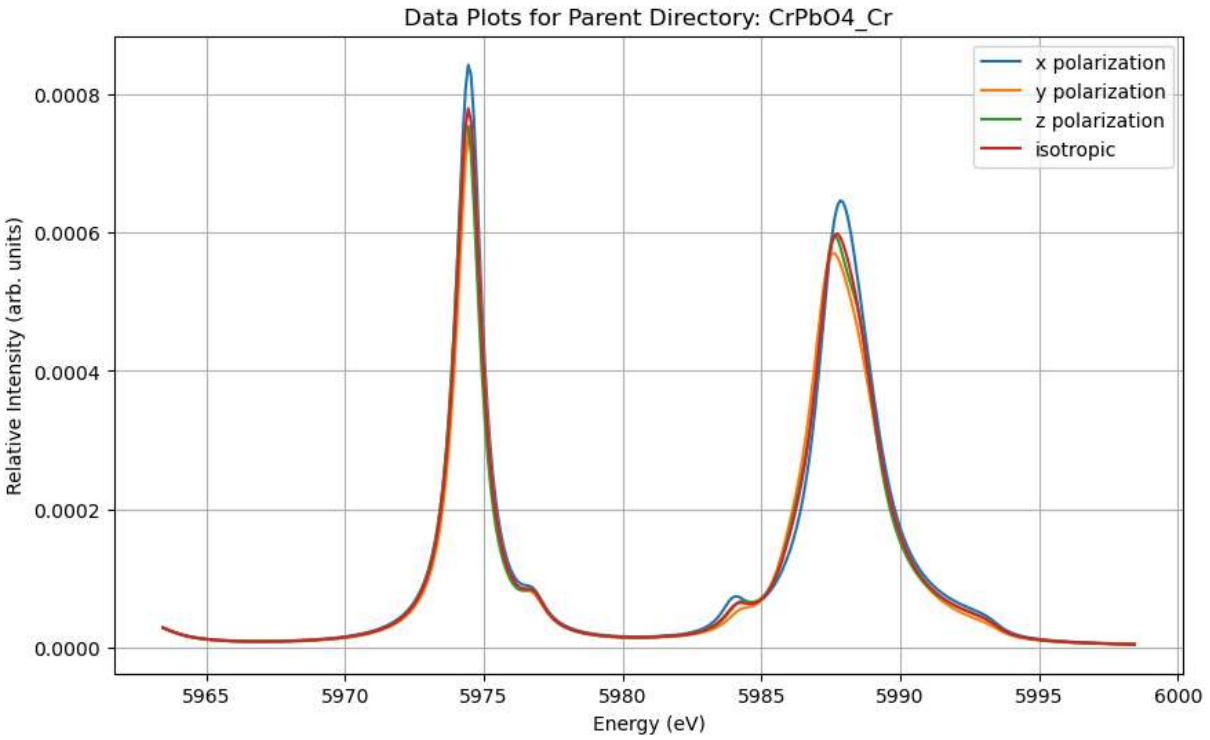
In [20]:

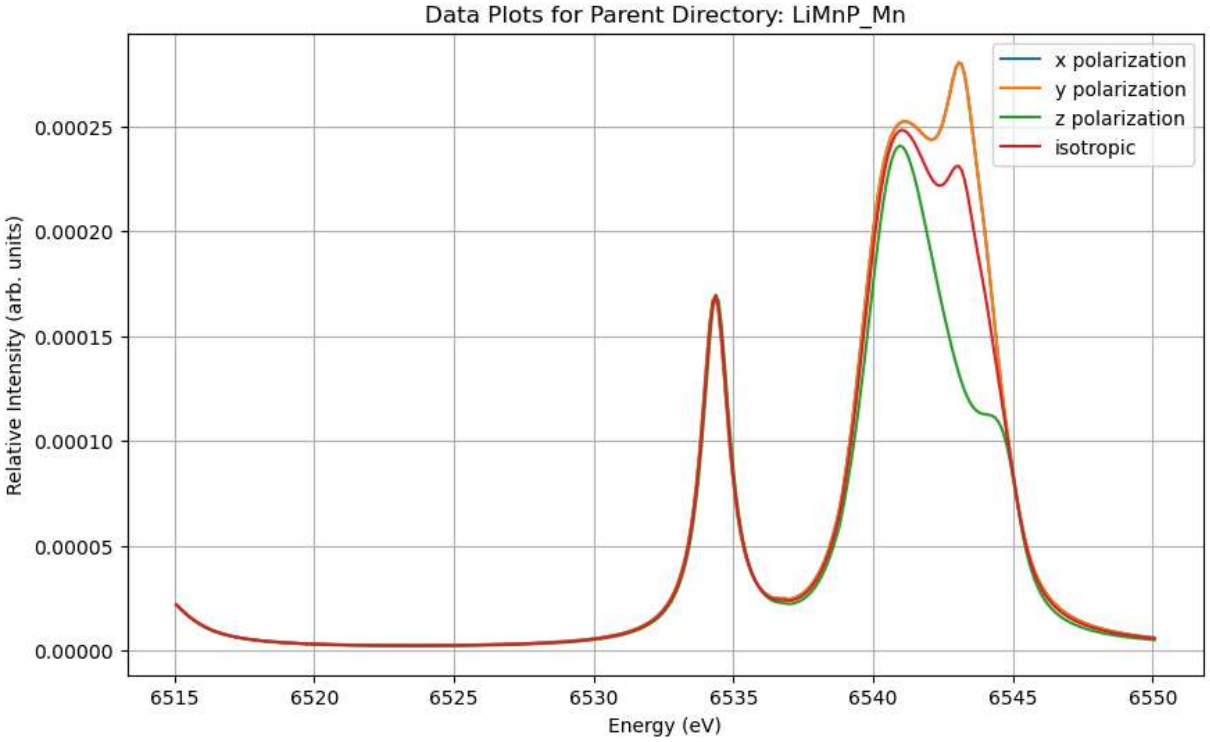
```
file_groups = defaultdict(list)
for file in matching_files:
    # Extract the second-to-last directory (parent directory)
    parent_dir = os.path.basename(os.path.dirname(file))
    file_groups[parent_dir].append(file)

# Plot each group of files on the same plot
for parent_dir, files in file_groups.items():
    plt.figure(figsize=(10, 6))

    # Plot data for each file in the group
    for file in files:
        data = read_data(file)

        if data.shape[1] >= 4:

            plt.plot(data[:, 0], data[:, 1], label='x polarization')
            plt.plot(data[:, 0], data[:, 2], label='y polarization')
            plt.plot(data[:, 0], data[:, 3], label='z polarization')
            plt.plot(data[:, 0], data[:, 4], label='isotropic')
        else:
            print(f'File {file} does not have enough columns to plot all combinatio

    plt.xlabel('Energy (eV)')
    plt.ylabel('Relative Intensity (arb. units)')
    plt.title(f'Data Plots for Parent Directory: {parent_dir}')
    plt.legend(loc='best')
    plt.grid(True)

    plt.show()
```

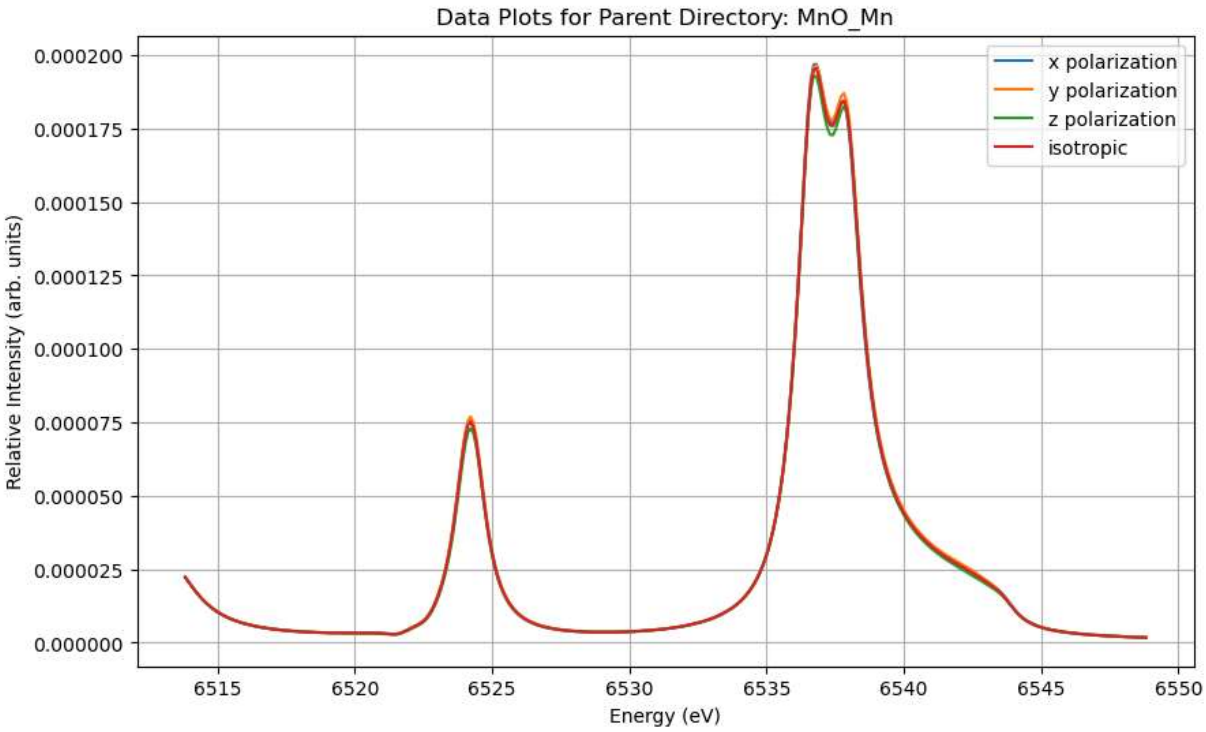

Data Plots for Parent Directory: Y2TiO5_Ti
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
4940
4945
4950
4955
4960
4965
4970
4975
Energy (eV)

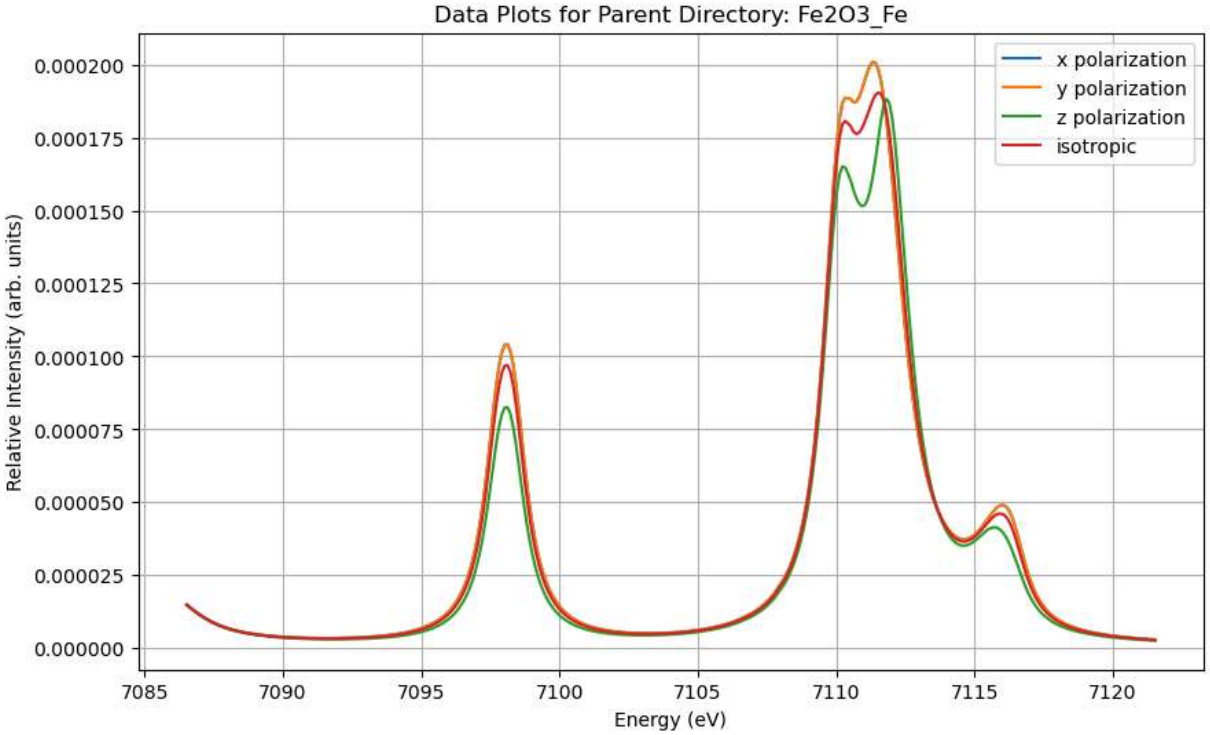

Data Plots for Parent Directory: V2O5_V
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
5440
5445
5450
5455
5460
5465
5470
5475
Energy (eV)

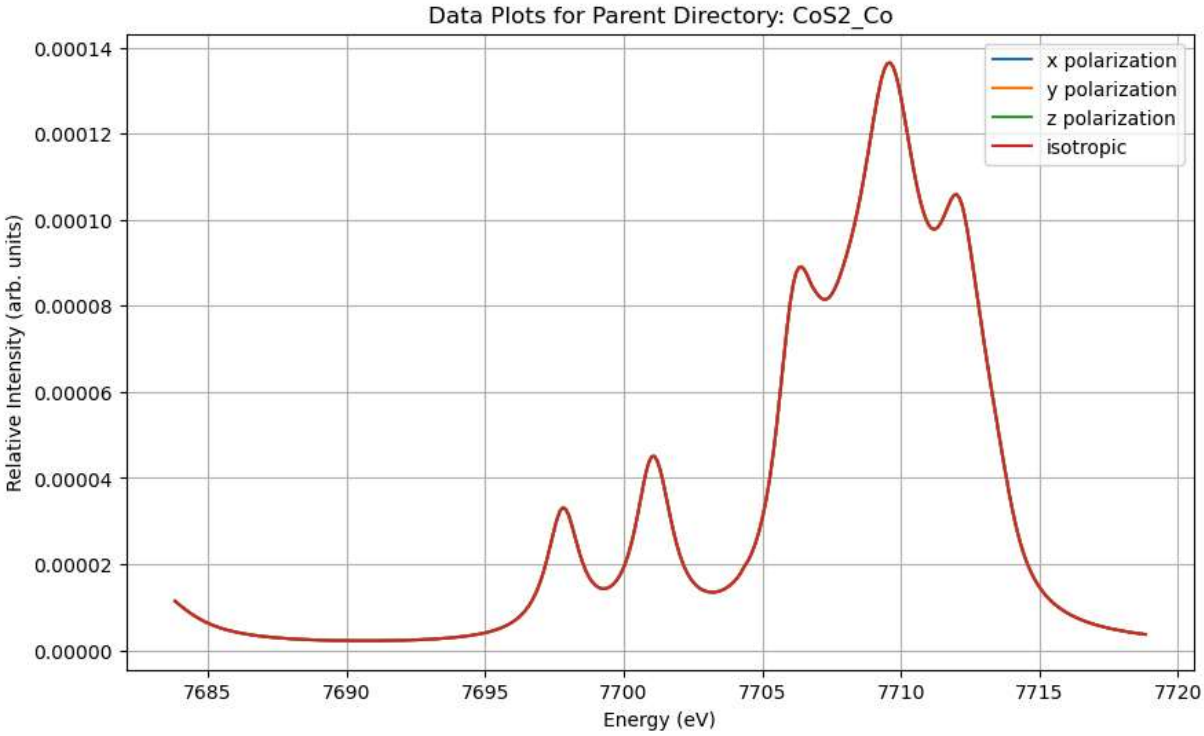

Data Plots for Parent Directory: Cr2O3_Cr
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.00035
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
5965
5970
5975
5980
5985
5990
5995
6000
Energy (eV)

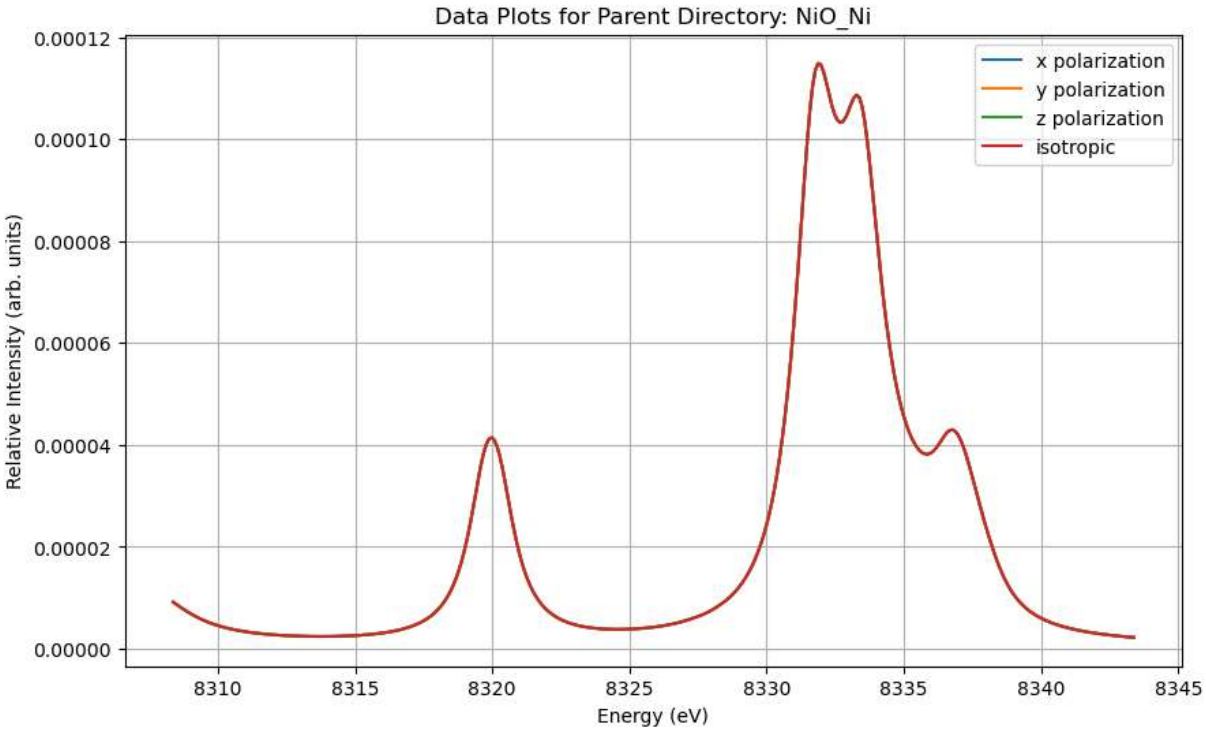

Data Plots for Parent Directory: CrPbO4_Cr
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.0008
0.0006
0.0004
0.0002
0.0000
5965
5970
5975
5980
5985
5990
5995
6000
Energy (eV)

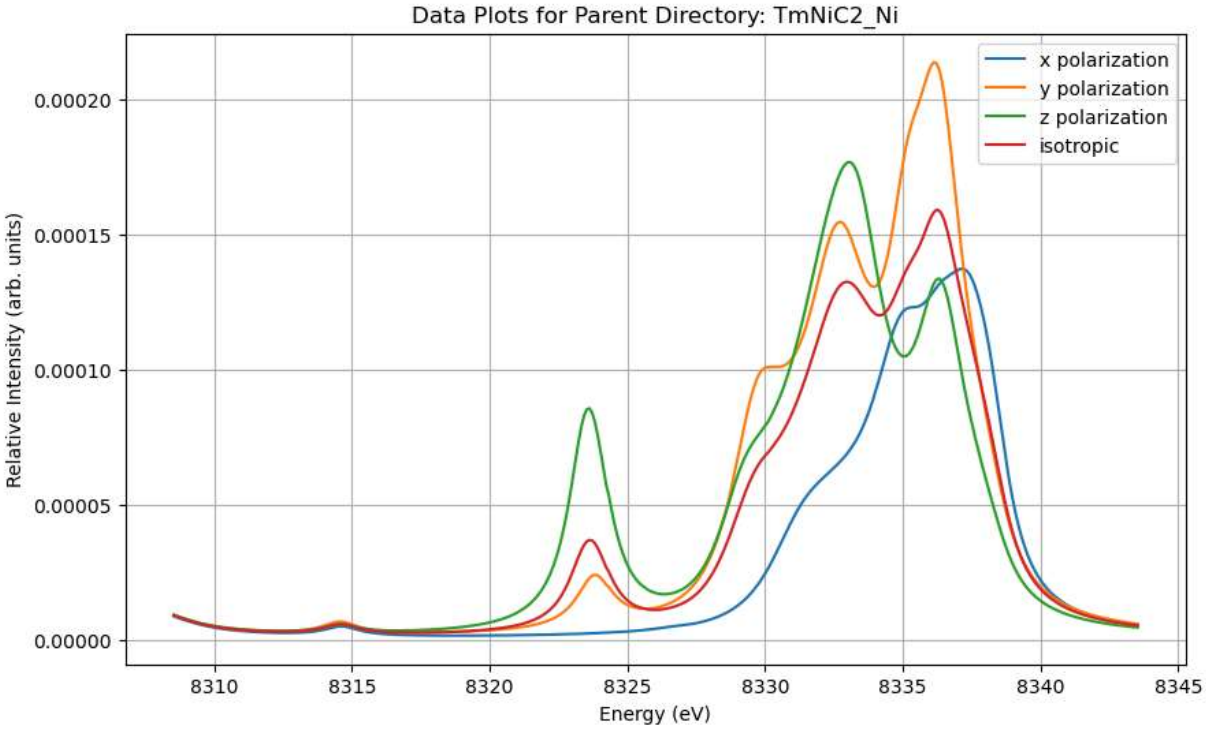

Data Plots for Parent Directory: LiMnP_Mn
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
6515
6520
6525
6530
6535
6540
6545
6550
Energy (eV)

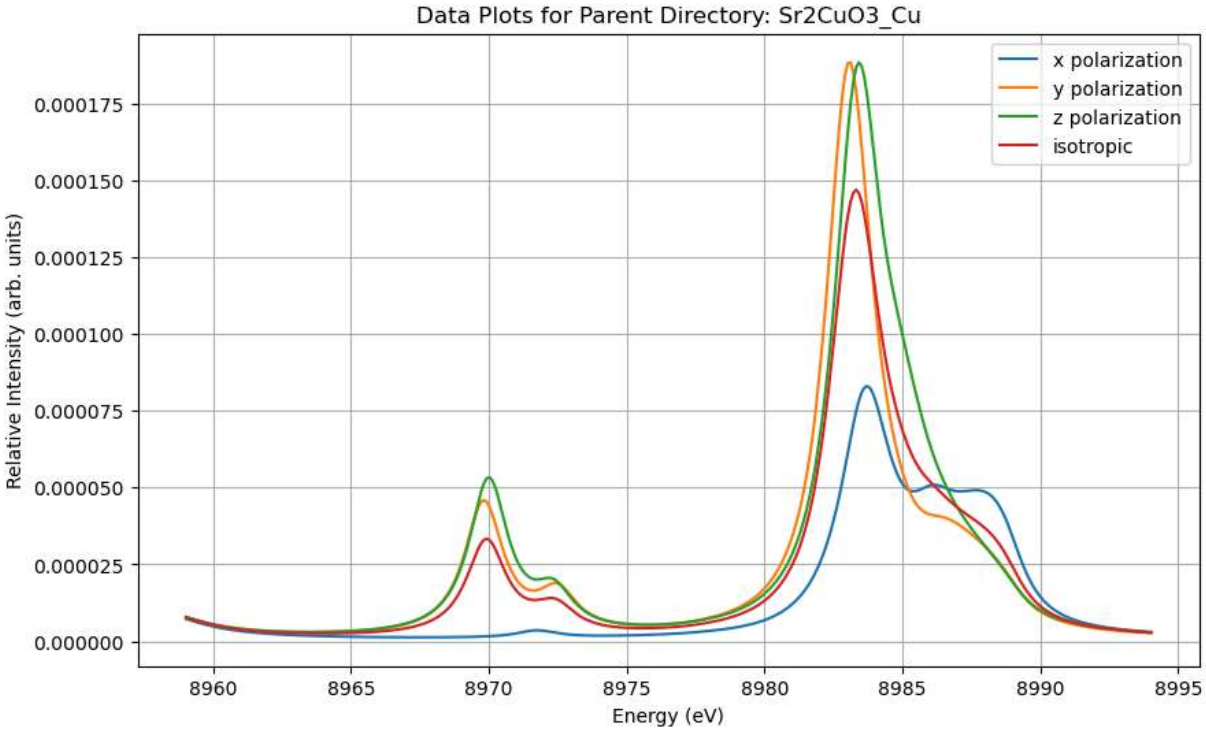

Data Plots for Parent Directory: MnO_Mn
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.000200
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
6515
6520
6525
6530
6535
6540
6545
6550
Energy (eV)

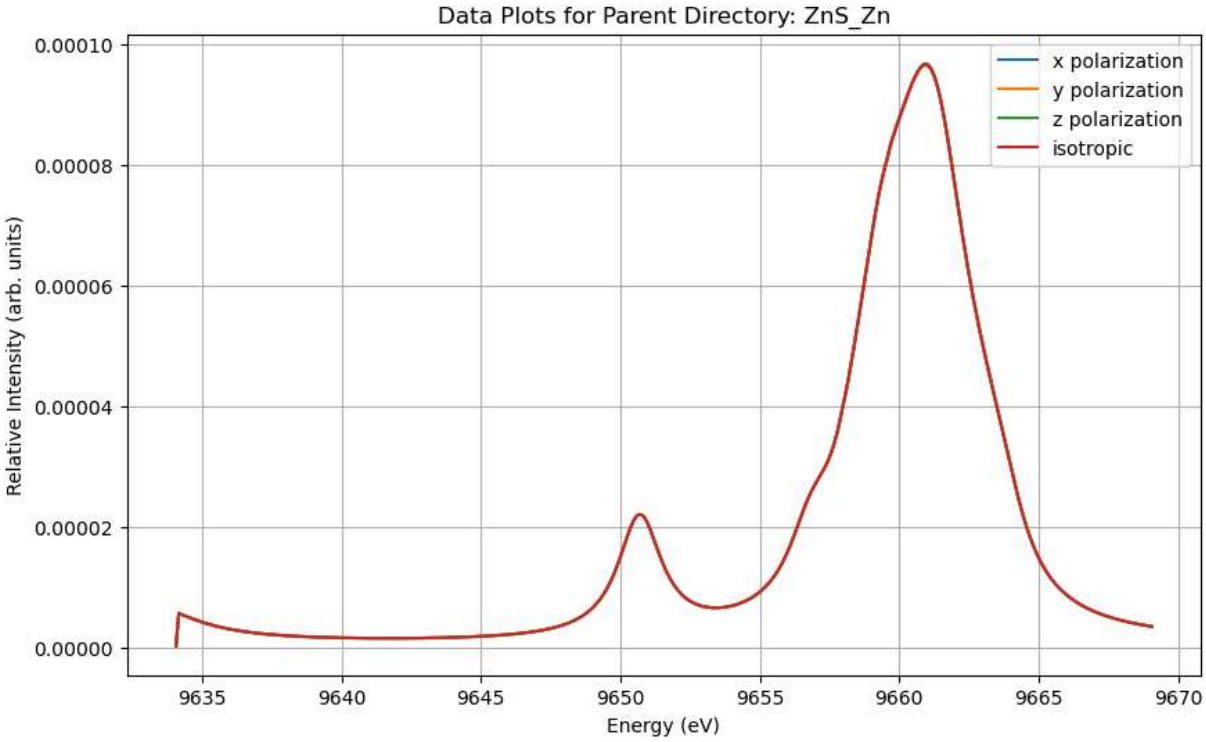

Data Plots for Parent Directory: Fe2O3_Fe
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.000200
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
7085
7090
7095
7100
7105
7110
7115
7120
Energy (eV)


Data Plots for Parent Directory: CoS2_Co
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.00014
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
7685
7690
7695
7700
7705
7710
7715
7720
Energy (eV)


Data Plots for Parent Directory: NiO_Ni
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
8310
8315
8320
8325
8330
8335
8340
8345
Energy (eV)


Data Plots for Parent Directory: TmNiC2_Ni
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.00020
0.00015
0.00010
0.00005
0.00000
8310
8315
8320
8325
8330
8335
8340
8345
Energy (eV)


Data Plots for Parent Directory: Sr2CuO3_Cu
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
8960
8965
8970
8975
8980
8985
8990
8995
Energy (eV)


Data Plots for Parent Directory: ZnS_Zn
x polarization
y polarization
z polarization
isotropic
Relative Intensity (arb. units)
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
9635
9640
9645
9650
9655
9660
9665
9670
Energy (eV)