PyWPEM : X-ray Spectroscopy Refinement Software (v1)

Whole Pattern fitting of powder X-ray diffraction by Expectation Maximum (WPEM) package.

Bin CAO[1] and TongYi ZHANG*[1]

[1]**Guangzhou Municipal Key Laboratory of Materials Informatics, Advanced Materials Thrust, Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China.**

Abstract The Whole Pattern Fitting of Powder X-ray Diffraction (WPEM) package is a Python library designed to support the refinement of various spectra, including XRD, XPS, and EXAFS. Developed by Mr. Cao Bin and Prof. Zhang Tong-yi, this paper provides detailed code examples and demonstrations for performing spectrum refinement. The GitHub repository can be found here: https://github.com/Bin-Cao/PyWPEM (note: the repository is currently private).

Citation : Cao, B., & Zhang, T. (2024). WPEM manual : Whole pattern fitting of powder X-ray diffraction using the Expectation-Maximization package. *Figshare*. https://doi.org/10.6084/m9.figshare.28078985

Keywords

Refinement, Code, XRD, XPS, EXAFS

Introduction

WPEM is a long-term project focused on X-ray-related refinement technologies. My development journey began in 2020, driven by a commitment to patience and precision in our work. Upon the publication of our final paper, we plan to share all of our code publicly. WPEM specializes in elucidating complex crystal structures and resolving heavily overlapping Bragg peaks in mixed X-rays and polycrystalline powder diffraction. It addresses key challenges in the refinement field, such as the precise detection of subtle structural differences (e.g., the and ’ phases of Ti-Ni alloy), distinguishing between amorphous and crystalline contributions in Polybutene, analyzing complex solid solution structures, and computing scattering information in organized processes.

This document introduces the architecture, execution code, and relevant theories in the refinement field, serving as supporting material for my doctoral thesis. While the package has not yet been published, I welcome collaborations and am happy to support those who require the functionality of WPEM. Please feel free to contact me at : bcao686@connect.hkust-gz.edu.cn (Mr. CAO Bin).

I have listed several publications related to this project below. Ultimately, I hope this comprehensive platform will be recognized as an intelligent system for AI-driven material characterization. I also deeply appreciate the help and guidance from my collaborators.

1. **2022**: We published the first experimental powder XRD refinement database, XRed, available at https://github.com/WPEM/XRED.
2. **2023**: We introduced a convolution-self-attention machine learning model for intelligent structure identification, published in *IUCrJ* [1]. We used the WPEM optimization model (pattern simulation) and achieved acceptable results. However, we recognized that the gap between simulation and experiment is a significant barrier to this technology. https://github.com/WPEM/CPICANN.
3. **2023**: My master’s thesis provides a comprehensive theory and case analysis of WPEM refinement technology. Due to restricted access, the paper will be made publicly available after the completion of our project.
4. **2024**: We further developed the simulation aspect, not just for refinement, but also to pursue more accurate experimental simulations. We introduced the first million-level database and a comprehensive benchmark to analyze data and model characteristics. Our paper is currently available as a preprint on ArXiv [2]: https://arxiv.org/pdf/2406.15469v1. The latest version contains more detailed representations and is currently under review.
5. **2024**: We have partnered with the Karlsruhe Institute of Technology (Germany) to establish a comprehensive powder XRD experimental database to standardize the process. https://xrd.aimat.science/. We welcome contributions to this project to accelerate progress in the field.
6. **2024–2025**: In an upcoming completed project, we will propose a holistic solution for structure identification. This work represents a milestone in our project, as we believe we have thoroughly addressed the challenge of intelligent structure identification.
7. **2025**: The structure identification model, as a crucial part of WPEM’s architecture, will enable integrated, full-process structure refinement at the millisecond level. Our paper is currently under preparation, and we hope to publish it soon. Once published, we will open-source all WPEM code and make the technology freely accessible.

General Code

Installation

To install the PyWPEM package, run the following command:

```
pip install PyWPEM
```

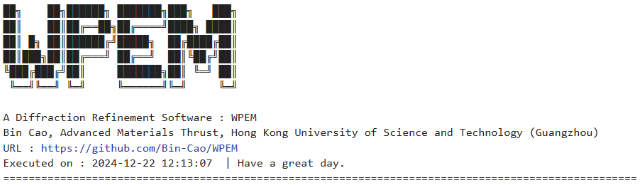
Execution

The following section provides a general code to execute the PyWPEM package.

1. Import the Package

First, import the package (Figure 1):

```
from PyWPEM import WPEM
```



```
A Diffraction Refinement Software : WPEM
Bin Cao, Advanced Materials Thrust, Hong Kong University of Science and Technology (Guangzhou)
URL : https://github.com/Bin-Cao/WPEM
Executed on : 2024-12-22 12:13:07  | Have a great day.
====================================================================================================
```

Figure 1

2. Background Stratification

To perform background fitting, load the intensity data and define the background parameters (Figure 2):

```
intensity_csv = pd.read_csv(r'intensity.csv', header=None)
var = WPEM.BackgroundFit(intensity_csv, lowAngleRange=17, poly_n=13, bac_split=16, bac_num=300)
```

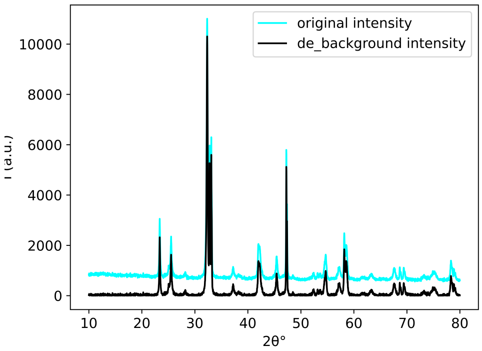
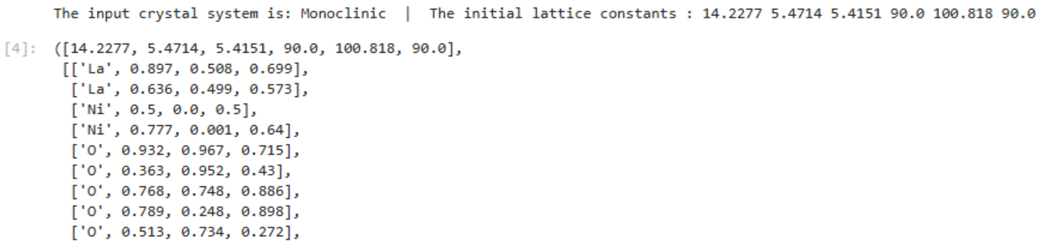


Figure 2

3. Parse CIF and Derive Initial Parameters

To preprocess the CIF file and calculate the initial parameters, use the following code (Figure 3):

```
WPEM.CIFpreprocess(filepath='ref_phase.cif', two_theta_range=(10, 80), show_unitcell=True,
    cal_extinction=True)
```



```
     The input crystal system is: Monoclinic  |  The initial lattice constants : 14.2277 5.4714 5.4151 90.0 100.818 90.0

[4]: ([14.2277, 5.4714, 5.4151, 90.0, 100.818, 90.0],
      [['La', 0.897, 0.508, 0.699],
       ['La', 0.636, 0.499, 0.573],
       ['Ni', 0.5, 0.0, 0.5],
       ['Ni', 0.777, 0.001, 0.64],
       ['O', 0.932, 0.967, 0.715],
       ['O', 0.363, 0.952, 0.43],
       ['O', 0.768, 0.748, 0.886],
       ['O', 0.789, 0.248, 0.898],
       ['O', 0.513, 0.734, 0.272],
```

Figure 3

4. Start Refinement Process

The results are shown in Figure 4

```
# The wavelength is set according to the actual light source
wavelength = [1.540593, 1.544414]


# Define the file names for the different data sets:
# The file name of non-background data (2theta-intensity data)
no_bac_intensity_file = "no_bac_intensity.csv"

# The file name of raw/original data (2theta-intensity data)
original_file = "intensity.csv"

# The file name of background data (2theta-intensity data)
bacground_file = "bac.csv"

# Input the initial lattice constants {a, b, c, alpha, beta, gamma}, whose values need to be assumed at
     initialization.
Lattice_constants = [[14.2277, 5.4714, 5.4151, 90.0, 100.818, 90.0]]

# Finally, execute the model:

WPEM.XRDfit(
    wavelength, var, Lattice_constants, no_bac_intensity_file, original_file, bacground_file,
    subset_number=11, low_bound=10, up_bound=50, bta=0.85, iter_max=100, asy_C=0, InitializationEpoch
        =0,
)
```

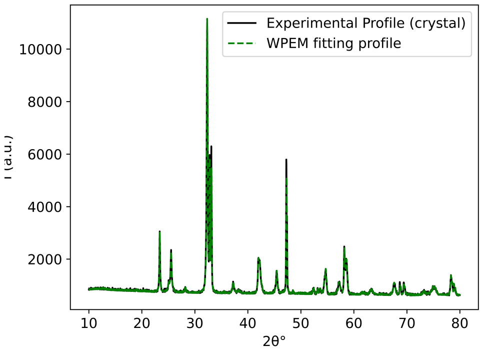


Figure 4

Architecture

Code logic

```
PyWPEM/
    __init__.py
    WPEM.py
    Extinction/
        __init__.py
        CifReader.py
        XRDpre.py
        Relaxer.py
        wyckoff/
        m3gnet/
          __init__.py
          wyckoff_dict.py
    Background/
        __init__.py
        BacDeduct.py
    EMBraggOpt/
        __init__.py
        EMBraggSolver.py
        BraggLawDerivation.py
        WPEMFuns/
            __init__.py
            SolverFuns.py
    Refinement/
        __init__.py
        VolumeFractionDertermination.py
    Amorphous/
        fitting/
            __init__.py
            AmorphousFitting.py
        QuantitativeCalculation/
            __init__.py
            AmorphousRDF.py
    DecomposePlot/
        __init__.py
        plot.py
    XRDSimulation/
        __init__.py
        Simulation.py
        DiffractionGrometry/
            __init__.py
            atom.py
    Raman/
        Decompose/
            __init__.py
            RamanFitting.py
    WPEMXAS/
            __init__.py
            EXAFS.py
            fftdemo.ipynb
    WPEMXPS/
            __init__.py
            XPSEM.py
    Plot/
            __init__.py
            UnitCell.py
    StructureOpt/
            __init__.py
            SiteOpt.py
    GraphStructure/
            __init__.py
            graph.py
```

Functions

WPEM.py

WPEM.py defines the main call interface of PyWPEM package.

```
XRDfit(wavelength, Var, Lattice_constants, no_bac_intensity_file, original_file, bacground_file,
    density_list=None, two_theta_range = None,structure_factor = None, ta=0.8, bta_threshold = 0.5,
    limit=0.0005, iter_limit=0.05, w_limit=1e-17, iter_max=40, lock_num = 2, asy_C=0.5, s_angle=50,
    subset_number=9, low_bound=65, up_bound=80, InitializationEpoch=2, MODEL = 'REFINEMENT',
    Macromolecule =False, cpu = 4, num =3, EXACT = False, Cu_tao = None, Ave_Waves = False,loadParams=
    False, ZeroShift=False,)

    :param wavelength: list type, The wavelength of diffraction waves
    :param Var: a constant or a array, Statistical variance of background
    :param Lattice_constants: 2-dimensional list, initial value of Lattice_constants
    :param no_bac_intensity_file: csv document, Diffraction intensity file with out bacground intensity
    :param original_file: csv document, Diffraction intensity file
    :param bacground_file: csv document, The fitted background intensity
    :param density_list : list default is None, the densities of crytal, can be calculated by fun. WPEM
        .CIFpreprocess()
        e.g.,
        _,_,d1 = WPEM.CIFpreprocess()
        _,_,d2 = WPEM.CIFpreprocess()
        density_list = [d1,d2]
    :param two_theta_range: The studied range of diffraction angels
    :param structure_factor: list, if EXACT = True, the structure factor is used calculating
            the mass fraction of mixed components
    :param bta: float default = 0.8, the ratio of Lorentzian components in PV function
    :param bta_threshold: float default = 0.5, a preset lower boundary of bta
    :param limit: float default = 0.0005, a preset lower boundary of sigma2
    :param iter_limit: float default = 0.05, a preset threshold iteration promotio (likelihood)
    :param w_limit: float default = 1e-17,  a preset lower boundary of peak weight
    :param iter_max: int default = 40, maximum number of iterations
    :param lock_num: int default = 3,  restriction  of loglikelihood iterations continously decrease
    :param asy_C, s_angle: Peak Correction Parameters
    :param subset_number (default = 9), low_bound (default = 65), up_bound (default = 80):
        subset_number peaks
            between low_bound and up_bound are used to calculate the new lattice constants by bragg law
    :param InitializationEpoch: int, default = 2, at initialization, frozen the peaks location for
        searching a satisified Model initial parameters.
    :param MODEL: str default = 'REFINEMENT' for lattice constants REFINEMENT; 'ANALYSIS' for
        components ANALYSIS
    :param Macromolecule : Boolean default = False, for profile fitting of crystals. True for
        macromolecules
    :param cpu : int default = 4, the number of processors
    :param num : int default = 3, the number of the strongest peaks used in calculating mass fraction
    :param EXACT : Boolean default = False, True for exact calculation of mass fraction by diffraction
        intensity theory
    :param Cu_tao: The restriction on the diffraction intensities of copper K 1 and K 2 rays.
    :param Ave_Waves: A boolean, default is False. Set to True to optimize using the average wavelength
         of K 1 and K 2.
    :param loadParams : Boolean default = False, for loading parameters
    :param ZeroShift : If ZeroShift == True and the standard sample is available, the instrument offset
         can be calibrated
    :return: An instantiated model
```

1. **wavelength**: Represents the wavelength of the incident X-ray.

2. **Var**: The variance of the background distribution, calculated by the `BackgroundFit()` function in step 2.

3. **Lattice_constants**: Represents the lattice constants of the components in the powder sample. As initial values, these can be derived from CIF files or obtained using our structure identification model. The lattice constant of each component can be frozen by [a,b,c,alpha,beta,gamma,'fixed']

4. **no_bac_intensity_file**: The file location of the diffraction data without background. This file is generated by the background processing module of WPEM and saved in the `ConvertedDocuments` folder as `no_bac_intensity.csv`.

5. **original_file**: The file location of the experimentally tested data.

6. **background_file**: The file location of the background intensity. This file is generated by the background processing module of WPEM and saved in the `ConvertedDocuments` folder as `bac.csv`.

7. **density_list**: The ideal densities of the mixture components in the powder sample, used for mass fraction calculation. These can be derived by the `CIFpreprocess()` function.

8. **two_theta_range**: The range of diffraction angles. If you wish to study the entire pattern, this parameter can be used to control the region of interest.

9. **structure_factor**: The structure factor of the crystals. By default, this is set to `None`. If you want to calculate the mass fraction of mixed components with the model set to `EXACT = True`, this must be provided.

10. **bta**: The ratio of Lorentzian components in the PV function.

11. **bta_threshold**: A preset lower boundary for `bta`, related to the algorithm's coverage.

12. **limit**: A preset lower boundary for σ^2 in the PV function, related to the algorithm's coverage.

13. **iter_limit**: A preset threshold for iteration promotion (likelihood), related to the algorithm's coverage.

14. **w_limit**: A preset lower boundary for peak weight, related to the algorithm's coverage.

15. **iter_max**: The maximum number of iterations allowed during execution.

16. **lock_num**: Represents the number of consecutive iterations where the log-likelihood continues to decrease, related to the algorithm's coverage.

17. **asy_C** and **s_angle**: Peak correction parameters that adjust the PV peak shape.

18. **subset_number**: The number of elements in the peak set constrained by Bragg's law.

19. **low_bound** and **up_bound**: Used to calculate the new lattice constants using Bragg's law.

20. **InitializationEpoch**: A tolerance parameter. During initialization, it freezes the peak locations to search for satisfactory model initial parameters.

21. **MODEL**: Default is `REFINEMENT` for lattice constant refinement; `ANALYSIS` for component analysis. If `MODEL = 'ANALYSIS'`, the statistics background will not be used. The reduced background can be analyzed further to confirm the second phase.

22. **Macromolecule**: A boolean, default is `False`. Set to `True` for profile fitting of macromolecules. See the macromolecule chapter for detailed calculations.

23. **cpu**: Represents the number of CPUs to be used during calculation.

24. **num**: The number of the strongest peaks used in calculating the mass fraction. This is only used when calculating the mass fraction.

25. **EXACT**: A boolean, default is `False`. Set to `True` for exact calculation of the mass fraction using diffraction intensity theory, considering structure factors.

26. **Cu_tao**: The restriction on the diffraction intensities of copper Kα_1 and Kα_2 rays. It can be a tolerance flat, where the Kα_1/Kα_2 ratio will vary between $(2 - \text{Cu\_tao})$ and $(2 + \text{Cu\_tao})$. It is suggested to use `None`.

27. **Ave_Waves**: A boolean, default is `False`. Set to `True` to optimize using the average wavelength of Kα_1 and Kα_2.

28. **loadParams**: A boolean, default is `False`. If set to `True`, it loads parameters from the last calculation.

29. **ZeroShift**: If `ZeroShift = True` and a standard sample is available, the instrument offset can be calibrated. The calculated pattern peaks will be matched with `peak0.csv`, and the offset will be computed. The user will need to minimize the offset and recalculate the pattern using WPEM.

Background stripping

The module contains the background strapping function in BacDeduct.py.
executed by WPEM main function as :

```
WPEM.BackgroundFit(intensity_csv, LFctg = 0.5, lowAngleRange=None, bac_num=None, bac_split=5,
    window_length=17, polyorder=3, poly_n=6, mode='nearest', bac_var_type='constant', Model='XRD',noise
    =None,segement=None)

    :param intensity_csv: the dir of experimental XRD data
    :param LFctg: low frequency filter Percentage, default  = 0.5
    :param lowAngleRange: low angle (2theta) with obvious background lift phenomenon
    :param bac_num: the number of background points in the background set
    :param bac_split: the background spectrum is divided into several segments
    :param window_length : int
        The length of the filter window (i.e., the number of coefficients).
        `window_length` must be a positive odd integer. If `mode` is 'interp',
        `window_length` must be less than or equal to the size of `x`.
    :param polyorder: int
        The order of the polynomial used to fit the samples.
        `polyorder` must be less than `window_length`.
    :param poly_n: background mean function fitting polynomial degree
    :param mode:  str, optional
        Must be 'mirror', 'constant', 'nearest', 'wrap' or 'interp'. This
        determines the type of extension to use for the padded signal to
        which the filter is applied.  When `mode` is 'constant', the padding
        value is given by `cval`.  See the Notes for more details on 'mirror',
        'constant', 'wrap', and 'nearest'.
        When the 'interp' mode is selected (the default), no extension
        is used.  Instead, a degree `polyorder` polynomial is fit to the
        last `window_length` values of the edges, and this polynomial is
        used to evaluate the last `window_length // 2` output values.
    :param bac_var_type:
        A pattern describing the background distribution
        one of constant, polynomial, multivariate gaussia
    :param  Model:
        Display the background curve of XRD diffraction spectrum (Model='XRD')
        or Raman spectrum (Model='Raman') or X-ray photoemission spectrography (Model='XPS') according
            to the type
    :param noise:
            float, default is None
            the noise level applied to gaussian processes model
    :param segement:
            A list containing the background point range. It can be easily defined by the user to
                manually adjust the background domains.
    :return:
        std of the background distribution
```

1. `intensity_csv`: The location of the experimental XRD data.
2. `LFctg`: The low-frequency filter percentage used in the FFT filter.
3. `lowAngleRange`: The low-angle (2θ) range where significant background lift is observed, for better background fitting.
4. `bac_num`: The number of background points in the background set.
5. `bac_split`: The number of segments into which the background is divided. Background data are selected from the background set.
6. `window_length`: The length of the filter window in the Savitzky-Golay filter.
7. `polyorder`: The order of the polynomial used to fit the samples. `polyorder` must be less than `window_length`.
8. `poly_n`: The degree of the polynomial used for background mean function fitting. A value of 6 is recommended.
9. `bac_var_type`: A pattern describing the background distribution. It can be one of the following: `constant`, `polynomial`, or `multivariate_gaussian`.
    - `constant`: Describes the variance of the background across the entire range using a constant value.
    - `polynomial`: Uses a polynomial to describe the background variance.
    - `multivariate_gaussian`: Models the background distribution as a Gaussian stochastic process.
10. `Model`: Displays the background curve for the XRD diffraction spectrum (Model='XRD'), Raman spectrum (Model='Raman'), or X-ray photoelectron spectroscopy (Model='XPS') depending on the type.
11. `noise`: The noise level applied to the Gaussian process model, used only when `bac_var_type` is set to `multivariate_gaussian`.
12. `segment`: A list containing the background point range. It can be easily defined by the user to manually adjust the background domains.

File Covert

```
WPEM.FileTypeCovert(file_name, file_type='dat')
    Convert XRD data from different file formats to the appropriate format.

    :param file_name: str
        The name of the original XRD data file (including file extension).

    :param file_type: str, optional, default='dat'
        The type of the original XRD data file. Can be 'dat' or 'xrdml'.
        - 'dat': Process the file as a .dat format.
        - 'xrdml': Process the file as a .xrdml format.

    :return: Processed data (depending on the file type)
```

Amorphous fitting

```
WPEM.Amorphous_fit(mix_component,amor_file = None, ang_range = None, sigma2_coef = 0.5, max_iter =
    5000, peak_location = None,Wavelength = 1.54184)

    :param mix_component : the number of amorphous peaks
    :param amor_file : the amorphous file location
    :param ang_range : default is None
        two theta range of study, e.g., ang_range = (20,80)
    :param sigma2_coef : default is 0.5
        sigma2 of gaussian peak
    :param max_iter : default is 5000
        the maximum number of iterations of solver
    : param peak_location : default is None
        the initial peak position of the amorphous peaks
        can input as a list, e.g.,
        peak_location = [20,30,40]
        the peak position can be frozen by the assigned input,
        peak_location = [20,30,40,'fixed']
    : param Wavelength : Wavelength of ray, default is 1.54184 (Cu)
```

This module fits the amorphous signals. Amorphous peaks can be identified from the crystal-singular stripping residual pattern.

1. `mix_component`: The number of amorphous peaks (a hyperparameter).
2. `amor_file`: The file location to be processed.
3. `ang_range`: The two-theta range to be studied.
4. `sigma2_coef`: Default value is 0.5 for each amorphous peak. It can be adjusted to represent the sigma^2 of all Gaussian peaks.
5. `max_iter`: The maximum number of iterations, related to the convergence.
6. `peak_location`: The initial positions of the amorphous peaks. This can be provided as a list, e.g.,

   $$\texttt{peak\_location} = [20, 30, 40]$$

   The peak positions can also be fixed by using the keyword `'fixed'`, e.g.,

   $$\texttt{peak\_location} = [20, 30, 40, \text{'fixed'}]$$

7. `Wavelength`: The wavelength of the incident rays. For amorphous materials, the value of 1.54184 (Cu Kα radiation) is commonly used.

RDF of amorphous

```
WPEM.AmorphousRDFun(wavelength=1.54184, amor_file=None,r_max = 5,density_zero=None,Nf2=None,highlight=
    4,value=0.6)
    Function to compute the radial distribution function (RDF) for amorphous materials based on X-ray
        diffraction data.

    Reference:
        J. Chem. Phys. 2, 551 (1934); https://doi.org/10.1063/1.1749528

    :param wavelength: float, optional, default=1.54184
        The wavelength of the X-ray used for diffraction. Typically set to 1.54184   for Cu  K
            radiation.

    :param amor_file: str, optional, default=None
        The file path to the amorphous phase intensity data. If not provided, the function will search
            for the default file
        located at '/DecomposedComponents/Amorphous.csv'.

    :param r_max: float, optional, default=5
        The maximum radius (in   ) from the center in the RDF plot. Defines the range of the radial
            distribution function.

    :param density_zero: float, optional, default=None
        The average density of the sample in atoms per cubic centimeter (atoms/cc). This is needed to
            calculate the
        atomic distribution in the material.

    :param Nf2: float, optional, default=None
        The effective number of atoms in the sample (N) multiplied by the atom scattering intensity (Aa
            ). This is used
        to compute the intensity contributions of each atomic species.

    :param highlight: int, optional, default=4
        The number of peaks to highlight in the RDF plot. Peaks are marked in the graphical output for
            easier visualization.

    :param value: float, optional, default=0.6
        Assumes that scattering can be treated as independent at sin(  /  ) = 0.6. This parameter
            influences how scattering
        is modeled for the RDF calculation.

    :return: tuple
        Returns the RDF plot and specified peaks.
```

The RDF reflects the local atomic arrangement, as in reference [3].

Show decomposed peaks

```
WPEM.Plot_Components(lowboundary, upboundary, wavelength, density_list=None, name = None, Macromolecule
     = False,phase = 1,Pic_Title = False,lifting=None)

    :param lowboundary : float, the smallest diffraction angle studied
    :param upboundary : float, the largest diffraction angle studied
    :param wavelength : list, the wavelength of the X-ray
    :param density_list : list default is None, the densities of crytal, can be calculated by fun. WPEM
        .CIFpreprocess()
        e.g.,
        _,_,d1 = WPEM.CIFpreprocess()
        _,_,d2 = WPEM.CIFpreprocess()
        density_list = [d1,d2]
    :param name : list, assign the name of each crystal through this parameter
    :param Macromolecule: whether it contains amorphous, used in amorphous fitting
    :param phase: the number of compounds contained in diffraction signals
    :param Pic_Title: Whether to display the title of the pictures, some title is very long
    :param lifting : list, whether to lift the base of each components,
```

XRD simulation

```
WPEM.XRDSimulation(filepath,wavelength=’CuKa’,two_theta_range=(10, 90, 0.01),SuperCell=False,
    PeriodicArr=[3,3,3],ReSolidSolution = None, RSSratio=0.1, Vacancy=False, Vacancy_atom = None,
    Vacancy_ratio = None,GrainSize = None,LatticCs = None,PeakWidth=True, CSWPEMout = None,orientation=
    None,thermo_vib=None,zero_shift = None, bacI=False,seed=42)

    :param filepath (str): file path of the cif file to be calculated
    :param wavelength: The wavelength can be specified as either a
                float or a string. If it is a string, it must be one of the
                supported definitions in the dict of WAVELENGTHS.
                Defaults to "CuKa", i.e, Cu K_alpha radiation.
    :param two_theta_range ([float of length 2]): Tuple for range of
        two_thetas to calculate in degrees. Defaults to (0, 90). Set to
        None if you want all diffracted beams within the limiting
        sphere of radius 2 / wavelength.
    :param SuperCell : : bool, default False
        If True, a supercell will be established
    :param PeriodicArr : list, default [3, 3, 3]
        Periodic translation the lattice 3 times along x, y, z direction
    :param ReSolidSolution : list, default None
        If not None, should contain the original atom type and replace atom type
        e.g., ReSolidSolution = [’Ru4+’, ’Na2+’], means ’Na2+’ replaces the ’Ru4+’ atom locations
    :param RSSratio :float, default 0.1
        In the supercell, the percentage of ’Ru4+’ atoms to be replaced randomly by ’Na2+’
    :param Vacancy : bool, default False
        If True, consider the charge balance, otherwise do not consider
    :param Vacancy_atom : str, default None
        If Vacancy is True, the atom to be considered for charge balancing
        e.g., Vacancy_atom = ’O2+’
    :param Vacancy_ratio : int, default None
        If Vacancy is True, the ratio of the number of vacancy atoms to the number of replaced atoms
        e.g., 1 means for each ’Ru4+’ atom replaced by ’Na2+’atom, a vacancy ’O2+’ is created for
            balancing the charge
    :param GrainSize
        The default value is ’none,’ or you can input a float representing
        the grain size within a range of 5-30 nanometers.
    :param LatticCs: The lattice constants after WPEM refinement. The default is None.
        If set to None, WPEM reads lattice constants from an input CIF file. Read parameters from CIF
            by using ..Extinction.XRDpre.
    :param PeakWidth
        PeakWidth=False, The peak width of the simulated peak is 0
        PeakWidth=True, The peak width of the simulated peak is set to the peak obtained by WPEM
    :param CSWPEMout : location of corresponding Crystal System WPEMout file
        if None, PEM simulates the peaks as the default Voigt function
        else WPEM simulates the peaks by the decomposed peak shapes
    :param orientation: The default value is ’none,’ or you can input a list such as [-0.2, 0.3],
        adjusting intensity within the range of (1-20%)I to (1+30%)I.
    :param thermo_vib: The default is ’none,’ or you can input a float, for example, thermo_vib=0.05,
        representing the variability in the average atom position. It is recommended to use values
            between 0.05 and 0.5 angstrom.
    :param zero_shift: The default is ’none,’ or you can input a float, like zero_shift=1.5,
        which represents the instrument zero shift. It is recommended to use values between 2    = -3
            and 3 degrees.
    :param bacI: The default is False. If bacI = True, a three-degree polynomial function is applied
        to simulate the background intensity.
    :param seed : default seed = 42
    return : Structure factors
```

Parse CIF

```
WPEM.CIFpreprocess(filepath, wavelength=’CuKa’,two_theta_range=(10, 90),latt = None, AtomCoordinates =
    None,show_unitcell=False,cal_extinction=True,relaxation=False)
    For a single crystal:
    Computes the XRD pattern and saves it to a CSV file.
    :param filepath: str
        The file path to the CIF file for which the XRD pattern will be calculated.
    :param wavelength: float or str, optional, default="CuKa"
        The wavelength of the X-ray. If provided as a string, it must be one of the keys in the
            WAVELENGTHS dictionary.
        By default, this is set to "CuKa", corresponding to Cu K_alpha radiation.
    :param two_theta_range: list of float, length 2, optional, default=(0, 90)
        A tuple specifying the range of 2   (in degrees) to calculate. Defaults to (0, 90).
        Set to None to include all diffracted beams within the limiting sphere of radius 2 / wavelength
            .
    :param latt: list
        The lattice constants, formatted as [a, b, c,   ,   ,   ], where ‘a‘, ‘b‘, and ‘c‘ are the edge
             lengths, and
        ‘  ‘, ‘  ‘, and ‘  ‘ are the angles between them (in degrees).
    :param AtomCoordinates: list of lists
        A list of atomic species and their coordinates in the unit cell, formatted as:
        [['Cu2+', 0, 0, 0], ['O-2', 0.5, 1, 1], ...].
        Note: '22' is the space group code.
        This input interface is designed to handle non-standard CIF files by allowing manual input for
            structure reading
        and method definition.
    :param relaxation: bool, optional, default=False
        Whether to relax the structure using the M3Gnet relaxation potential field.
    :return: tuple
        A tuple containing:
        - ‘latt‘: The lattice constants [a, b, c,   ,   ,   ].
        - ‘AtomCoordinates‘: The atomic species and coordinates in the unit cell, e.g., [['Cu2+', 0, 0,
             0], ['O-2', 0.5, 1, 1], ...].
        - ‘lattice_density‘: The calculated lattice density (  ).
```

Optimize substitution structures

```
WPEM.SubstitutionalSearch(xrd_pattern, cif_file,random_num=8, wavelength='CuKa',search_cap=50,
    SolventAtom = None, SoluteAtom= None,max_iter = 100,cal_extinction=True)

    :param xrd_pattern: str
        The path to the experimental XRD diffraction pattern of a single crystal, containing 2theta and
             intensity values.
    :param cif_file: str
        The path to the CIF (Crystallographic Information File) associated with the crystal structure.
    :param random_num: int
        The number of times the structure will be randomly initialized to establish the training
            dataset for BGO.
    :param wavelength: float or str, optional, default="CuKa"
        The wavelength of the X-ray used. If provided as a string, it must be one of the keys in the
            WAVELENGTHS dictionary.
        By default, this is set to "CuKa", which corresponds to Cu K_alpha radiation.
    :param search_cap: int
        This parameter limits the number of combinations considered during the search to avoid
            excessive memory usage. It helps to truncate the search process and prevent memory overflow
            .
    :param solvent_atoms: str, optional, default=None
        The solvent atoms in the system. If not provided, it defaults to None. For example, SolventAtom
             = 'Cu'.
    :param solute_atoms: str, optional, default=None
        The solute atoms in the system. If not provided, it defaults to None. For example, SoluteAtom =
             'Ti'.
    :param max_iter: int
        The maximum number of iterations to run during the computation.
    :param cal_extinction: bool, optional, default=False
        Whether to consider the extinction effect in the calculation. Set to ‘True‘ if the extinction
            effect should be included, ‘False‘ otherwise.
```

In this function, the substituted ratio is iteratively explored, with all possible scenarios considered as combinations. The parameter search_cap is an integer used to limit the search space, preventing memory overflow. The process involves replacing one atom with another, generating a new PXRD pattern, and calculating the residuals with respect to the experimental data. The results are then encoded as training data and used in Bgolearn to search for the next promising substitution structure that may match the observed pattern.

X-ray Photoelectron Spectroscopy

```
intensity_csv = pd.read_csv(r'int.csv',header=None )
var = WPEM.BackgroundFit(intensity_csv,segement=[[910,931],[948,952],[958,959],[966,970]],bac_num=120,
    Model='XPS',noise = 0.05,bac_var_type='multivariate gaussian')
```

```
import sys
import pandas as pd
Yourdir = '/Users/jacob/Documents/GitHub/'
```

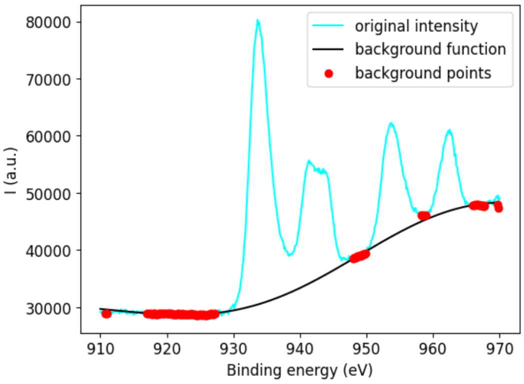


Figure 5

```
sys.path.append(Yourdir)
from PyWPEM import WPEM

AtomIdentifier = [[’CuII’,’2p3/2’,933.7,],[’CuII’,’2p1/2’,954,],]
satellitePeaks = [[’CuII’, ’2p3/2’,941.6,],[’CuII’,’2p3/2’,943.4],[’CuII’,’2p1/2’,962.5,],]
# The file name of non-background data
no_bac_intensity_file = "no_bac_intensity.csv"
# The file name of raw/original data
original_file = "int.csv"
# The file name of background data
bacground_file = "bac.csv"

# Execute the model
WPEM.XPSfit(
var, AtomIdentifier, satellitePeaks,no_bac_intensity_file, original_file, bacground_file,
    bta = 0.80,iter_max = 500, InitializationEpoch=1,)
```

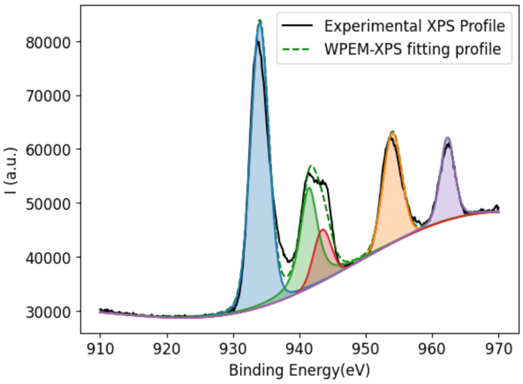


Figure 6

```
WPEM.XPSfit(Var, atomIdentifier, satellitePeaks,no_bac_df, original_df, bacground_df, energy_range =
    None, bta=0.8, bta_threshold = 0.5,limit=0.0005, iter_limit=0.05, w_limit=1e-17, iter_max=40,
    lock_num = 2, asy_C=0., s_energy=[100,1000], tao=0.5, ratio=0.8, InitializationEpoch=2,loadParams=
    False,)
    :param Var: Variance of the background intensity.
    :param atomIdentifier: List of atom identifiers.
        - Each element is a list describing the electron state and binding energy.
        - Example: [[’CuII’,’2p3/2’,933.7,],[’CuII’,’2p1/2’,954,],]
    :param satellitePeaks: List of satellite peaks.
        - Each element is a list describing the electron state and satellite peak energy.
        - Example: [[’CuII’, ’2p3/2’,941.6,],[’CuII’,’2p3/2’,943.4],[’CuII’,’2p1/2’,962.5,],]
    :param no_bac_df: DataFrame containing the direct electron binding energy pattern.
    :param original_df: DataFrame containing the experimentally observed XPS data.
    :param bacground_df: DataFrame containing the fitted background pattern.
    :param energy_range: Energy range studied in the spectrum. Default is None.
    :param bta: Ratio of Lorentzian components in the Pearson VII (PV) function. Default is 0.8.
    :param bta_threshold: Preset lower boundary of ‘bta‘, related to algorithm convergence. Default is
        0.5.
    :param limit: Preset lower boundary of sigma , related to algorithm convergence. Default is
        0.0005.
    :param iter_limit: Minimum threshold for the likelihood improvement during iteration. Default is
        0.05.
    :param w_limit: Preset lower boundary of peak weight. Default is 1e-17.
    :param iter_max: Maximum number of iterations allowed. Default is 40.
    :param lock_num: Number of consecutive iterations with decreasing log-likelihood before termination
        . Default is 2.
    :param asy_C: Asymmetry parameter used to describe asymmetric peaks. Default is 0.
    :param s_energy: Energy range for asymmetric peak modeling. Energies lower than ‘s_energy‘ will be
        treated as asymmetric peaks. Default is [100, 1000].
    :param tao: Fine-tuning parameter for binding energy. Ensures smaller changes between iterations,
        especially when focusing on a few peaks in XPS fitting. Default is 0.5.
    :param ratio: Adjustment factor for peak location during overfitting.
        - If the change suggested by the EM algorithm exceeds ‘tao‘, the new peak location is updated
            as:
          ‘new_mu_list[peak] = ratio * ori_mu_list[peak] + (1 - ratio) * new_mu_list[peak]‘.
        - Default is 0.8.
    :param InitializationEpoch: Number of epochs during initialization where peak locations are frozen
        to find satisfactory model parameters. Default is 2.
    :param loadParams: Boolean flag to determine whether to load existing parameters. Default is False.
```

X-ray Absorption Fine Structure

```
WPEM.EXAFS('absorb.csv',de_bac = True).fit(first_cutoff_energy=22100,second_cutoff_energy=22400)
```

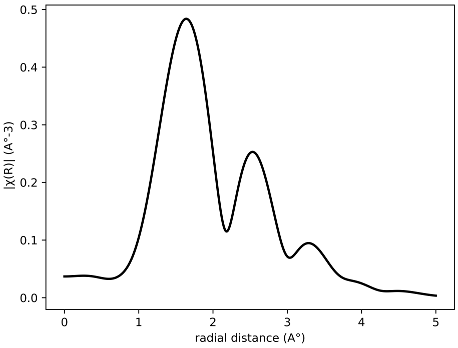


Figure 7

```
WPEM.EXAFSfit(XAFSdata,  power = 2, distance = 5, k_point = 8,k = 3,s= None,window_size=30,hop_size=
    None,Extend=50,name = 'unknown',transform ='fourier',de_bac = False,Ezero = None,
    first_cutoff_energy=None,second_cutoff_energy=None)

    Parameters:
    -----------
    XAFSdata : str
        The name of the input data.
    power : int, optional, default=2
        Scales the y-axis for larger k-ranges, compensating for weaker signals at high k values.
    distance : float, optional, default=5
        The maximum radial distance in real space (R-space).
    k_point : int, optional, default=8
        The cutoff range for k points.
    k : int, optional, default=3
        Degree of the smoothing spline (1     k     5).
        A cubic spline is used when k=3.
    s : float or None, optional, default=None
        Positive smoothing factor for selecting the number of knots.
        The smoothing condition is defined as:
            sum((w[i] * (y[i] - spl(x[i])))**2, axis=0) <= s
        If ‘s‘ is None, it defaults to ‘len(w)‘. If ‘s=0‘, the spline interpolates through all data
            points.
    window_size : int, optional, default=30
        The length of each segment used in FFT analysis.
    hop_size : int or None, optional, default=None
        The number of points to overlap between segments. If ‘None‘, defaults to ‘window_size / 8‘.
    Extend : int, optional, default=50
        Extends the observed energy range by this value for EXAFS signal analysis.
    name : str, optional, default='unknown'
        The chemical formula or name associated with the data.
    transform : str, optional, default='fourier'
        The inverse transform method to use. Options are 'wavelet' or 'fourier'.
    de_bac : bool, optional, default=False
        Whether to fit and remove the absorption background. This step is typically performed during
            data collection at a synchrotron light source.
    Ezero : float or None, optional, default=None
        The absorption edge energy ( E  ). If ‘None‘, the function will estimate  E   as the energy
            with the maximum slope in the absorption edge.
    first_cutoff_energy : float or None, optional, default=None
        The energy value before the observation range, used to fit the absorption background.
    second_cutoff_energy : float or None, optional, default=None
        The energy value after the observation range, used to fit the mean observation after
            photoelectron ejection.

    Notes:
    ------
    1. The parameter ‘de_bac‘ controls whether the absorption background is removed. This is typically
        unnecessary if data has been preprocessed at the source.
    2. Parameters related to smoothing splines (‘k‘ and ‘s‘) allow for fine control over the
        interpolation and smoothing process.
    3. FFT-related parameters (‘window_size‘, ‘hop_size‘) influence the frequency-domain analysis of
        the EXAFS data.
    Returns:
    --------
    Processed EXAFS data ready for further analysis.
```

References

[1] Shouyang Zhang, Bin Cao, Tianhao Su, Yue Wu, Zhenjie Feng, Jie Xiong, and Tong-Yi Zhang. Crystallographic phase identifier of a convolutional self-attention neural network (cpicann) on powder diffraction patterns. *IUCrJ*, 11(Pt 4):634, 2024.

[2] Bin Cao, Yang Liu, Zinan Zheng, Ruifeng Tan, Jia Li, and Tong-yi Zhang. Simxrd-4m: Big simulated x-ray diffraction data accelerate the crystalline symmetry classification. *arXiv preprint arXiv:2406.15469*, 2024.

[3] B Eo Warren. X-ray diffraction study of carbon black. *The journal of chemical physics*, 2(9):551–555, 1934.

Tables

.1 crystal systems

Table 1. Unit cell of seven crystal systems.

Code	Crystal system	Unit cell characteristics
1	Cubic	$a = b = c,\ \alpha = \beta = \gamma = 90^\circ$
2	Hexagonal	$a = b \neq c,\ \alpha = \beta = 90^\circ,\ \gamma = 120^\circ$
3	Tetragonal	$a = b \neq c,\ \alpha = \beta = \gamma = 90^\circ$
4	Orthorhombic	$a \neq b \neq c,\ \alpha = \beta = \gamma = 90^\circ$
5	Trigonal	$a = b = c,\ \alpha = \beta = \gamma \neq 90^\circ$
6	Monoclinic	$a \neq b \neq c,\ \alpha = \gamma = 90^\circ \neq \beta$
7	Triclinic	$a \neq b \neq c,\ \alpha \neq \beta \neq \gamma \neq 90^\circ$

.2 diffraction index

Table 2. The interplanar spacing of seven crystal systems.

Crystal system	Interplanar spacing/d_hkl
Cubic	$\frac{a}{\sqrt{H^2+K^2+L^2}}$
Hexagonal	$\frac{\sqrt{3}/2}{\sqrt{\frac{H^2+K^2+HK}{a^2}+\frac{3L^2}{4c^2}}}$
Tetragonal	$\frac{1}{\sqrt{\frac{H^2+K^2}{a^2}+\frac{L^2}{c^2}}}$
Orthorhombic	$\frac{1}{\sqrt{\frac{H^2}{a^2}+\frac{K^2}{b^2}+\frac{L^2}{c^2}}}$
Trigonal	$a \cdot \sqrt{\frac{1-3\cos^2\alpha+2\cos^3\alpha}{(H^2+K^2+L^2)\sin^2\alpha+2(HK+KL+HL)\cos^2\alpha-\cos\alpha}}$
Monoclinic	$\frac{\sin\beta}{\sqrt{\frac{H^2}{a^2}+\frac{K^2}{b^2}\sin^2\beta+\frac{L^2}{c^2}-\frac{2HL}{ac}\cos\beta}}$
Triclinic	$d = \frac{m}{\sqrt{\left(\frac{H}{a}\right)^2\sin^2\alpha+\left(\frac{K}{b}\right)^2\sin^2\beta+\left(\frac{L}{c}\right)^2\sin^2\gamma-2\frac{hk}{ab}q-2\frac{hl}{ac}n-2\frac{kl}{bc}p}}$

Note: where $m = \sqrt{\sin^2\gamma - \cos^2\alpha + \cos^2\beta + 2\cos\alpha\cos\beta\cos\gamma}$,
$p = \cos\alpha - \cos\beta\cos\gamma$, $n = \cos\beta - \cos\alpha\cos\gamma$, $q = \cos\gamma - \cos\alpha\cos\beta$.