ohy)

/ for **near-field** and **far-field imaging** in **TEM/STEM**.

stribution Deconvolution)

C)

)

/AG SEM, Physics Department, Humboldt-Universität

ronment for PyPty, you can use **conda**, **mamba**, or

ıtion instructions using `conda`.

```
nton-Gladyshev/pypty.git

.gpu.yml
```

```
nton-Gladyshev/pypty.git

cpu.yml
```

gradient-based phase reconstruction from far-field
`ypty.iterative.run()`. Most other functions are
is core function.

ments for `pypty.iterative.run()` is available in

single input dictionary called `pypty_params`, which
ntal and reconstruction settings. These
en defined separately in a dictionary called
eriment description).

`ty_params` for one experiment, you can reuse it for
experimental parameters.

ct methods and tools for initialization.

files and provides tools to convert from NumPy.
ataset named `"data"` with a 3D array: one scan

e NumPy arrays (with accompanying JSON

ut as `.npy` files inside an `output_folder`. The
is:

```
                **overwritable checkpoint for s
                **overwritable checkpoint for p
                **overwritable checkpoint for o
l.npy           **checkpoint for object at epoc
_N.npy          **checkpoint for probe at epoch
poch_N.npy      **checkpoint for scan grid at e
                **parameter file of your recons
                **.csv log-file**
                **folder  with tcBF results**
ing_5.npy       **upsampled tcBF image_**
                **intermediate tcBF image at it
l.npy           **Fitted shifts at iteration N
                **Fitted aberrations (in Angstr
                **array constaining fitted rota
                **2d array constaining fitted a
                **folder  with DPC results**
                **fft-based dpc phase**
                **DPC phase (iterative reconstr
                **folder  with WDD results**
                **complex object (WDD ptychogra
```

folder into single Nexus `.nxs` file via
ction.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

e in the `examples` folder on GitHub.

ollowing:

— A complete pipeline example

ains key functions and logic

stribution and other direct approaches

vanced setups

eading this guide, the following papers, books and
principle of the code:

the NN-style approach

mputing in Electron Microscopy

n. General framework for quantitative three-
n arbitrary detection geometries in TEM

n. Method for Retrieval of the Three-Dimensional
Dynamical Electron Scattering

nformation reduced data and experimentally
graphy with regularized optimization

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

latform generic X-ray image reconstruction
differentiation

airos. Maximum-likelihood refinement for coherent

mpression of Electron Diffraction Patterns for
sis

models for low counting rate coherent diffraction

structing state mixtures from diffraction
g formalism (for near-field imaging)

n-linear inline holography reconstruction algorithm

omputing in Electron Microscopy

p sub-angstrom resolution imaging by electron
n correction

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

information reduced data and experimentally
graphy with regularized optimization

mpression of Electron Diffraction Patterns for
sis

on of functions having Lipschitz continuous first
1969). Convergence Conditions for Ascent Methods

onvergence of a class of double-rank minimization

pproach to Variable Metric Algorithms

of Variable Metric Updates Derived by Variational

oning of quasi-Newton methods for function

nalen theorie der funktionen von mehr komplexen

ent - PyPty (python + ptychography)

...with the `initialize`

`ze` module (see the API Reference), which is
d ptychography.

cratch

ves additional complexity or non-standard
ustom Preset Guide for more flexible configuration.

pical **far-field ptychography** case, you can
parameters in a single Python dictionary. For clarity,
`rimental_params`.

e following keys:

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Description
Path to a **3D PyPty** `.h5` **dataset** (`[N_measurements, ky, kx]`) or a **4D-STEM** `.npy` **dataset**.
If the data is compressed, provide the virtual detectors (`[N_masks, ky, kx]`).
Directory where results will be stored.
Path to a Nion-style `.json` file with metadata (optional).

erties

Description
Accelerating voltage (in **kV**).
Reciprocal pixel size (in **Å⁻¹**).
Reciprocal pixel size (in **mrad**).
Beam convergence semi-angle (in **mrad**).
Binary **2D mask** representing the aperture.
Threshold for estimating an aperture. Everything above `threshold * max(PACBED)` is considered bright field.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

g

	Description
	Number of scan points along **slow (y)** and **fast (x)** axes.
	Scan step (STEM pixel size) in **Å**.
	Field of view (**FOV**) along the **fast axis** in **nm**.
	If data was acquired on a non-rectangular grid, specify positions as `[y_0, x_0], ..., [y_n, x_n]` (in **Å**).
	Transformation matrix for position correction.
	Rotation angle between scan and detector axes. If `"auto"`, an iDPC measurement estimates this angle.

ngs

	Description
	Number of slices used for multislice propagation (default: `1`).
	Total thickness of the sample (in **Å**).
	Reciprocal space padding. Default: `1/4` of pattern width.
	Upsampling factor for diffraction patterns.
	Flip the y-axis of diffraction patterns.
	Extra probe defocus (besides aberrations).
	Beam aberrations (stored in Krivanek notation).

hography - PyPty (python + ptychography)

amples for Iterative

	Description
	If `True`, generates plots of key experimental parameters.
	Controls verbosity (`0` = silent, `1` = summary, `2+` = detailed logs).

breakdown of an iterative ptychography

jet started, create a Python script and import the

```
preprocessing, while all major operations are
ypty._version_)
```

small preprocessing outside of pypty, but the major
nctions.

Description
You can optionally provide a formula for your sample. PyPty will store this information.
You can optionally provide a name of your sample. PyPty will store this information.

rative ptychography

he steps required to do a ptychographic

STEM data (far field)

and center a 4D-STEM dataset (in h5 format). PyPty
ys, but it's better to do this preprocessing step.

```
t
,"


pty_data(path_raw, path_h5, swap_axes=False,
y=0,flip_kx=0, comcalc_len=200*200,
, comy=0, bin=1, crop_left=None,
ight=None, crop_top=None, crop_bottom=None, n
```

om Nion microscopes, but you can also specify all
dictionary `experimental_params` with relevant
description for further details.

```
ve this string like this and specify everythi
folder


t_folder,
,
```

```
e of the following three parameters
ne,
 None,
 33.8,


, ## this is number of scan points along slow

ne of the following two parameters
 scan step in Angstrom


kV

0],## Angstrom  [C10, C12a, C12b], can be lon

, # deg


2, ## this parameter will be used to estimate

## Angstrom, thickness of the sample
umber of slices in your reconstruction

 will plot a few things
x verbosity
lone will add 1/4 of pixels to each side. This

us tags you can specify. This will help to id
UR_Formula,
_YOUR_Sample,
```

reset

graphy are stored in a dictionary typically named

red as .pkl data and you can load them via a

```
.oad_preset(path_to_your_pkl_preset)
```

em from a nexus data of another reconstruction

```
.oad_nexus_params(path_to_your_nxs_preset)
```

your custom preset as a dictionary from scratch

```
ıg,
```

`None`, then you will have a standard preset

```
```

rimental parameters

add experimental data to it.

```
.ize.append_exp_params(experimental_params, py
```

on and single argument, `pypty_params`:

```
.params)
```

ıs one Nexus file

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

e, you will find a bunch of files in the
them into one big nexus file via a single function.
xs file will be saved via `path_to_your_nexus_file`

```
(output_folder, path_to_your_nexus_file)
```

ut an extra h5 file

xtra .h5 file (**not recommended**), you can specify
experimental parameters (before you join them

```
et"]=dataset
```

`data_path` will be ignored and you can leave this
n be either 4D or 3D (with 2 scan axes merged into

y from Virtual Detectors

ruction from data compressed by virtual detectors,
array of detectors has shape `[N_detectors, k_Y,`
`[Scan_Y, Scan_X, N_detectors]`. Then you attach
l parameters.

```
data,
tor_array,


must specify one of the following two paramet
ne,
 1,
```

API Reference

pypty.initialize
pypty.dpc
pypty.tcbf
pypty.direct
pypty.iterative
pypty.objective
pypty.multislice_core
pypty.fft
pypty.utils
pypty.se
pypty.vaa

```
ımeters.
```

our reconstruction settings. For Compressed data
nction types:

on
ıared fit
noise model
ın noise model

hm as one of these three options in `pypty_params`:

```
s_compressed,
parameters
```

phy with the same pypty.iterative.run function.

ds - PyPty (python + ptychography)

ect methods

age of PyPty's direct reconstruction methods:
C), Tilt-Corrected Bright Field (tcBF), and Wigner
D). These methods are used when a direct (non-
rred or as preprocessing for iterative methods.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ntrast (DPC)

ions for differential phase contrast. Both of them
, i.e. to reconstruct a phase from its laplacian,
. The first method, based on the Fast Fourier
e phase efficiently and can be executed as follows
atplotlib)

```
ypty.dpc.fft_based_dpc(pypty_params, hpass=1e

="gray")
based)")
```

both phase and parameter dictionary as it stores
(and the angle between real and reciprocal spaces).
`pypty_params`, it attempts to automatically

accounts for non-periodic boundary conditions:

```
ive_dpc(pypty_params, num_iterations=100, bet
="gray")
ative)")
```

on for detailed description of the parameters.

ield (tcBF)

in two steps. First one aims to fit the aberrations of
ween real and reciprocal spaces by creating an
`un_tcbf_alignment`:

```
ın_tcbf_alignment(

.tile([5], 30),
ıe,

ıbs",


:s=0.9,
:",
```

binning values and cross-correlation parameters to
l rotation angle typically has a +- pi ambiguity, so if
vith a sign different from the experiment, I suggest

s that you already did an alignment and performs an

```
pypty.tcbf.upsampled_tcbf(pypty_params, upsam
          pad=10, default_float=32,round_shif

ıap="gray")
upsampling)")
```

econvolution (WDD)

ethod is Wigner distribution deconvolution (WDD). It
ucts the object's complex wavefunction by
nsity distribution with the known probe. WDD
`ams` dictionary containing rotation angle, scan steps
f aberrations). A simple usage example is

```
ypty_params,  eps_wiener=1e-3)

ld), cmap="gray")
```

ves as a regularization term (high-pass filter) to
enting division by zero.

s for Initial Guesses

d with a function `get_ptycho_obj_from_scan` that
ransmission function based on a measurement
can be a DPC or WDD phase. Usage example is:

```
ialize.get_ptycho_obj_from_scan(pypty_params,
```

h finer sampling than the actual scan grid of your
ate an upsampled grid in the image coordinates
s like this:

```
image
l size of your image
how many pixels of the image are from left o
how many pixels of the image are from top of
ize.get_grid_for_upsampled_image(pypty_params

ialize.get_ptycho_obj_from_scan(pypty_params,
```

he initial guess by providing keys like `"obj"`,
ore see Guide for creating custom presets.

tial tools in PyPty for rapid insight and as a starting
tructions. You can combine them with iterative
o the initialization phase.

rs for Creating Custom

sing a dictionary that defines your reconstruction
y named `pypty_params`.

rative ptychographic reconstruction is
accepts `pypty_params` as its single argument.

nd using the `pypty.initialize` module, which
eters and arrays automatically. However, for more
s, this guide explains how to construct your own
ally.

eters in PyPty

ails, it's important to understand a special type of
pe parameters.

e parameters can be adjusted depending on the
lambda function that takes the current epoch
alue dynamically.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

only every 10 epochs, set it like this:

```
% 10 == 0;
```

ten in this way are marked as `pypty_lambda` **type** in hey can also be specified as a sting containing the

```
 % 10 == 0";
```

Jsage

constraints or critical parameters every few epochs. y's BFGS optimizer, which builds a Hessian matrix Sudden changes can disrupt convergence and lead

s so that once a parameter is activated, it **remains** t of the reconstruction. This ensures smooth patibility with the underlying solver.

eter starting at epoch 20 and keep it active

```
-3*(x >= 20)
```

on advanced features or constraints after an abilizing the optimization.

s for pypty.iterative.run()

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Default Data Type	Description
`NumPy-like` `python` `module`	Currently not used, but will be a feature in the future. Right now, whenever CuPy is available, it is used as the GPU backend. If no CUDA is detected, NumPy is used as a CPU replacement. We plan to add support for Apple Silicon, but we are waiting for an optimal library to appear.
`str`	Default data type for computations. Another option is `"single"`.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

	Default Data Type	Description
	`str`	Path to the dataset. It can be an `.h5` file with a dataset named `"data"` containing a **3D measurement array** `(N_measurements, y, x)`. Another option is a **4D** `.npy` **array** or a **3D** `.npy` **array**.
	`numpy-array` or None	If you don't want to store data on disk, you can attach a numpy-array with your data to the parameters. If it's provided, `data_path` is ignored.
	`numpy.ndarray` or `None`	Masks (virtual detectors) used for data compression. For uncompressed data, leave it as `None`.
	`float`	Multiplier for data values. Used to rescale patterns on the fly without modifying the stored dataset. All patterns will be multiplied by this number.
	`int`	Padding applied to data. Use it to pad patterns on the fly without modifying the stored dataset. We recommend setting it to **1/4 of the pattern width** for optimal sampling conditions in far-field mode.
	`int`	Binning factor for data. Used to bin patterns on the fly without modifying the stored dataset. All patterns will be binned by this number.
	`bool`	Flag indicating that one has to flip `ky`. Useful if patterns are flipped and you don't want to modify the stored dataset. Another option is to create a PyPty-style `.h5` dataset.

`list`	Shift vector (list with two-values) applied to measurements. Used to shift patterns on the fly without modifying the stored dataset. All patterns will be shifted by the specified number of pixels.
`int`	Upsampling factor. If the beam footprint is larger than the extent (in far-field mode), this allows to artificially upsample the beam in reciprocal space. **Experimental feature!** Windowing constraints may be required.
`list` or `None` or `pypty_lambda`	Sequence used in data processing. This is a list indicating the measurements that will be used for iterative refinement. If `None`, all measurements contribute. This parameter is useful for reconstructions on subscans without creating additional data files.
`bool`	Boolean flag. Only useful if a sequence is provided. If `True`, the object can accommodate all measurements. If `False`, the object accommodates only selected measurements.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Default Data Type	Description
`str`	Path to the folder where output files will be saved.
`bool`	Boolean flag. If `True`, the loss log will be saved as `loss.csv`.
`int`	Previous epoch count. Useful for restarting a reconstruction.
`bool` or `int`	Save checkpoints every epoch. If `True`, checkpoints will be always saved, if it is provided as an integer, checkpoints will be saved every n'th epoch.
`bool` or `int`	Save intermediate **overwritable** checkpoints. This will create `.npy` arrays: `co.npy` for the object, `cp.npy` for the probe, `cg.npy` for the scan grid, `ct.npy` for the tilts, `cs.npy` for the static background, and `cb.npy` for the beam current. If `True`, checkpoints will be always saved, if it is provided as an integer, checkpoints will be saved every n'th epoch.
`int`	Print verbosity level: `0` for no printing, `1` for one overwritable line, `2` and `3` for moderate output. `4` gives the most detailed output.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Default Data Type	Description
`float`	Acceleration voltage in kV.
`numpy.ndarray` or `None`	Mask for the aperture. Can be used for reciprocal probe constaint later (see section constraints).
`str`	Type of reconstruction. Options: `"far_field"` or `"near_field"`.
`float`	Alpha parameter for near-field reconstruction & flux preservation.
`numpy.ndarray`	Array of defocus values for near-field measurement. Irrelevant for far-field. It can contain either a single common defocus value for all measurements or individual values for each measurement. **Units: Angstroms.**
`float`	Spherical aberration coefficient. **Units: Angstroms.**

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Default Data Type	Description
`numpy.ndarray`	Distances between object slices. **Units: Angstroms.** You can specify a single value common for all slices or provide individual values.
`float`	Pixel size in the **x-direction** (**Angstroms**).
`float`	Pixel size in the **y-direction** (**Angstroms**).
`tuple` or `None`	Tuple describing the number of scan points in **y- and x- directions**. Required for constraining positions and tilts.
`int`	Number of slices in the object.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Default Data Type	Description
`numpy.ndarray`	Initial guess for the transmission function to be retrieved. Shape: `(y, x, z, modes)`. If the `y` and `x` dimensions are insufficient for the scan grid, the object will be padded with ones.
`numpy.ndarray` or `None`	Real-space probe. Shape: `(y, x, modes)`. For advanced experiments, the probe can be 4D `(y, x, modes, subscans)`. If `None`, PyPty will automatically initialize the beam from the dataset.
`numpy.ndarray`	Scan positions in **pixels** of the reconstruction. Shape: `[N_measurements, 2]`, formatted as `[[y0, x0], [y1, x1], ..., [yn, xn]]`. Single-shot experiments can define one common scan point, e.g., `[[0,0]]`.
`numpy.ndarray`	Tilt angles in **real and reciprocal spaces**. There are 3 types of tilts in PyPty framework: before, inside and after. First one is a beam tilt before the specimen, i.e. a shift in aperture plane. Second type is a tilt inside of a specimen, i.e. after each slice the beam is shifted in real space. Third type is a post-specimen tilt i.e. a shift in a detector plane. All three types of shifts are contained in this tilt array. Shape: `[N_measurements, 6]`. Format: `[[y0_before, x0_before, y0_inside, x0_inside, y0_after, x0_after], ..., [yN, xN]]`. Single-shot experiments can define one common tilt (with shape `[1, 6]`).
`int`	Mode for applying tilts: `0, 3, 4` for **inside**, `2, 4` for **before**, and `1, 3, 4` for **after** the

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

	specimen.
`numpy.ndarray` or `float`	Static background intensity. Shape should match initial patterns but padded by `data_pad//upsample_pattern`. Use `0` for no static offset. If provided as postive float, the algorithm will initialize the backgrund with a proper shape on its own.
`numpy.ndarray` or `None`	Accounts for different currents (or exposure times) during measurements. If provided, must be a **1D array** with length matching `N_measurements`.

Resizing

Default Data Type	Description
`str`	Wave propagation method. Options: `"multislice"`, `"better_multislice"`, and `"yoshida"`. The last two are **higher precision but slower**.
`bool`	Allow **subpixel shifts**. If `False`, positions will be rounded to integers until refined.
`bool` or `int` or `pypty_lambda`	If position updates become too large, the object will be padded to accommodate the new scan grid. If set to a **positive integer**, resizing occurs when position updates exceed this value.
`int`	Extra space added around the object in **pixels**.

Default Data Type	Description
`float`	Frequency cutoff for **multislice beam propagation**. Values larger than `2/3` can cause aliasing artifacts. Recommended **≤ 2/3**.
`float`	Rolloff parameter for smooth frequency cutoffs.
`float`	Extra frequency cutoff for the **full object**. Ensures bandwidth limitation beyond the **cropped ROIs** of the multislice object.
`bool`	If `True`, the full transmission function **will not** be bandwidth-limited (only cropped ROIs will be). **Recommended:** `False`.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Default Data Type	Description
`str`	Error metric for reconstruction comparison. Options: `"lsq_sqrt"` (Gaussian), `"ml"` (Poisson), `"lsq"` (classic summed squared error), and `"lsq_sqrt_2"` (modified Gaussian). If data is compressed via virtual detectors, the only option is `"lsq_compressed"` (summed squared error between signals).
`int`	Maximum number of **epochs (iterations)**.
`float` or `pypty_lambda`	**Wolfe condition parameter (C1)**. Prevents update steps from being too large. Must be **> 0** and **< C2**. Larger values enforce shorter step size.
`float` or `pypty_lambda`	**Wolfe condition parameter (C2)**. Prevents update steps from being too small. Must be **> C1** but **< 1**. Larger values allow larger steps.
`float` or `pypty_lambda`	Weight applied to the **loss function**.
`int` or `None`	Maximum number of forward-backward propagations per **line search iteration**. If exceeded, the update is **rejected** and history is reset. Use `None` or `np.inf` to disable.
`float`	Factor for **reducing step size** when the first Wolfe condition **is not met**.
`float`	Factor for **increasing step size** when the second Wolfe condition **is not met**. To prevenet algorithm from going back and

	forth during linesearch, multiplication of optimism and reduce_factor (or of any powers of them) should not be equal to 1.
`float`	**Minimum step size**. If the step falls below this value, the algorithm **resets history**. Use `0` to disable.
`int` or `np.inf` or `pypty_lambda`	**BFGS optimization history length**. Values: `0` (Gradient Descent), `1` (Conjugate Gradient), `N>1` (Limited-memory BFGS), `np.inf` (Full BFGS).
`float` or `pypty_lambda`	Common step applied to all refinable quantities. By default, after the first iteration, a Barzilai-Borwein method is used to inistialize the inverse Hessian, so most of the time, an update step of 1 should be accepted. Only during the very first iteration the linesearch might take some time to find an appropriate step.
`bool` or `pypty_lambda`	Whether to consider the object as phase-only.
`bool` or `pypty_lambda`	Optimize only the reciprocal-space phase (CTF) of the probe.
`bool` or `pypty_lambda`	Optimize only the reciprocal-space amplitude (aperture) of the probe.
`bool` or `pypty_lambda`	Flag to reset optimization history. See section "lambda-types" in this document. If provided, history will be manually resetted.

Default Data Type	Description
`bool` or `pypty_lambda`	Whether to update the **probe** (`1` for yes, `0` for no).
`bool` or `pypty_lambda`	Whether to update the **object** (`1` for yes, `0` for no).
`bool` or `pypty_lambda`	Whether to update **probe positions** (`1` for yes, `0` for no).
`bool` or `pypty_lambda`	Whether to update **tilt angles** (`1` for yes, `0` for no).
`bool` or `pypty_lambda`	Whether to update **beam current** (`1` for yes, `0` for no).
`bool` or `pypty_lambda`	Whether to update **aberration array** (`1` for yes, `0` for no).
`bool` or `pypty_lambda`	Whether to update **static background** (`1` for yes, `0` for no).

tions

Default Data Type	Description
`numpy.ndarray`	Array of aberration values for multiple beams. Useful for large fields of view where the beam changes. Shape: `[N_subscans, N_aberrations]`.
`str` or `None`	Path to an **HDF5 file** containing phase plates for different measurements. Dataset name should be `"configs"`. Shape: `[N_measurements, Y_probe, X_probe]`.
`numpy.ndarray` or `None`	**Marker for multiple CTFs**. Should be a **1D array** of length `N_measurements`, where each entry corresponds to a CTF index in `aberrations_array`.
`numpy.ndarray` or `None`	**Marker for probe variations**. If provided, the probe should have shape `[y, x, modes, N_subscans]`, and this array should contain indices specifying which probe to use for each measurement.

Default Data Type	Description
`bool`	If `True`, data is **loaded dynamically** to save GPU memory. If `False`, all data is loaded at once (faster but memory-intensive).
`bool` or `pypty_lambda`	If `True`, **memory is managed intelligently**, clearing cache when necessary to prevent memory fragmentation.
`bool`	If `True`, **FFT cache is removed** periodically to save memory. **(Experimental feature)**
`int` or `str`	**Batch size for multislice computation.** Default value "auto" will automatically estimate a value that would fit into your memory, but ideally one should balance it by hand for a particular GPU. Increasing this can speed up reconstruction but requires more GPU memory. Super large values are also useless as at some point you will reash the limit the compute capability.
`numpy.dtype`	Forces the dataset to be stored in a specified **data type**. Can help reduce memory usage at the cost of precision.
`bool`	If `True`, **preloads data to CPU** before transferring it to GPU, improving transfer speeds for `.h5` datasets.
`bool`	If `True`, pads data **at the start** of reconstruction (uses more memory but speeds up computation). If `False`, padding is applied **on the fly** to save memory.
`float`	If compute batch is set to `"auto"`, this ratio will be used to estimate a compute

batch that would requiere `memory_saturation` of the avalible GPU memory. If `compute_batch` is integer, `memory_saturation` is ignored.

ɔ the Loss

ıe - PyPty (python + ptychography)

mple pipeline

Default Data Description

```
 np
ib.pyplot as plt


.patches import Rectangle
.patches import Circle
.colors import LogNorm
.patches import Ellipse
 import rcParams
dark_background')
ersion__)
```

`numpy.ndarr ay` or `None`

Mask for **probe constraint in reciprocal space**. Masked pixels are **regularized using L2 norm**.

d output path

`float`

L2 regularization weight for the **probe in reciprocal space**.

```
on/data/test_folder/"

ame+".h5"

ame+".npy"
/home/anton/data/test_folder/test_mops_2/"
```

`numpy.ndarr ay` or `None` or or `pypty_lambd a`

Window function used to constrain the probe in real space. Masked pixels are damped using L2 regularization. It can be either a 2d- real valued array with the same shape as upsampled and padded beam **or** a list containing two values: inner

```
e.create_pypty_data(path_raw, path_h5, swap_a
    flip_ky=0,flip_kx=0, comcalc_len=200*200
    comx=0, comy=0, bin=1, crop_left=None,
    crop_right=None, crop_top=None, crop_bot
```

of window will be kept intact, everything outside will be zeroed and intermediate values will be slighly damped.

```
********************************************
****** Creating an .h5 File ****************
**********
```

`float`

L1 regularization weight applied to the **absorptive potential** (negative log of the transmission function's absolute value).

`float` or `pypty_lambd`

L1 regularization weight applied to the **phase of the object**.

e Object and Probe 'By Hand'

`float` or

t the BFGS history when applied. Weight for **Adaptive Total Variation (ATV)**

```
rams={
der': output_folder,
: path_json,
: path_h5,
ngle_mrad': 4,
: [28,28],
A': 20,
*27,
e': 200, ## kV
_deg': 0, # deg
eshold': 0.2,
e,
': 1,
 None,
e': 'nano_mops'


':  1,
be': 1,

h': 10,
: "bfgs",
p_bfgs': lambda x: 1e-2 if x==0 else 1,

_meas': 1,
 1e-30,
 3,
tor': 1/10,
onstant': 1e-1,
y_resize_yx_object': 5,
s': [-2e4,0,0], ## Angstrom # 200A

onstant': 0.9,


ra_cut': 0.02,

_checkpoints': True,
points_every_epoch': False,
```

```
:  'lsq_sqrt',
OV': True,

: 100,

y_one': False,
_cpu': False,
ry': False,
: True,
ype': 'double',
toff_multislice': 0.66,


pty.initialize.append_exp_params(experimental
```

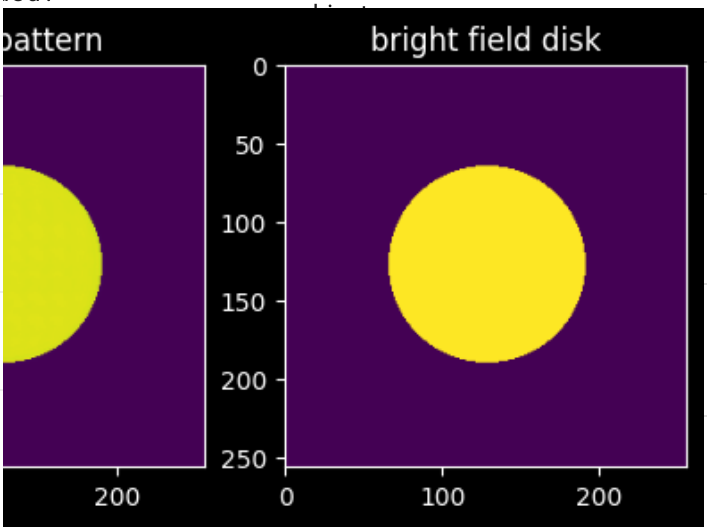
`pypty_lambda`	low kx and ky in **3D object FFTs**.

```
**********************************************
```

`bool` or `pypty_lambda`	Enforces **orthogonality of probe modes**.

```
g the experimental parameters to your PyPty p
**********************************************
```

`bool`	Performs **charge flipping** on the



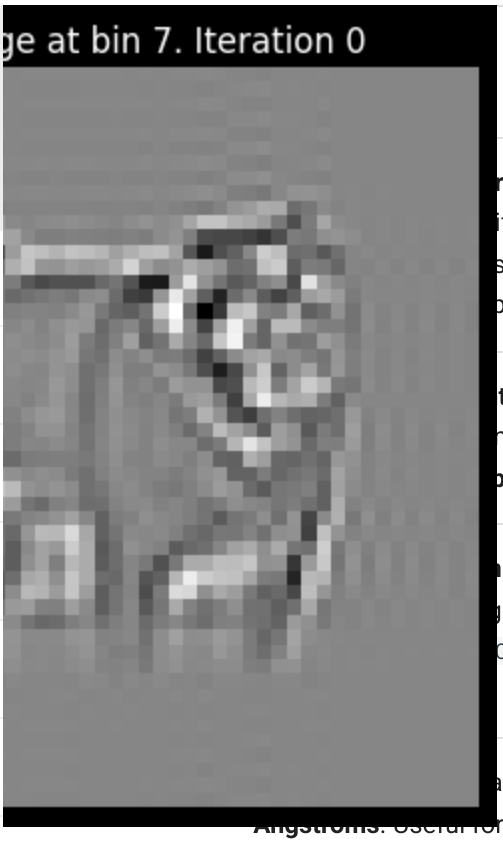

```
ded!
field is 62.52 px
1.53fe+00
adding 1.02e+00 Å
```

`float` or `pypty_lambda`	Delta amplitude for charge flipping.
`float` or `pypty_lambda`	Beta phase parameter for charge flipping.
`float` or `pypty_lambda`	Beta amplitude parameter for charge flipping.

```
pty.tcbf.run_tcbf_alignment(pypty_params,
     aberrations=[-1.5e4,0,0],
     binning_for_fit=np.tile([7],30),

    refine_box_dim=10, upsample=3,
    reference_type="bf", optimize_angle=False
    cancel_large_shifts=0.9,
    subscan_region=None, interpolate_scan_fac
    cross_corr_type="phase"
   )
```

`bool` or `pypty_lambda`	**Resets the positions to initial guess** while keeping other parameters unchanged. See section "lambda-types" in this document.

```
********************************************
****** Running the tcBF alignment **********
********************************************

ns: C10 -1.50e+04 Å, C12a 0.00e+00 Å, C12b 0.
data:  (784, 256, 256)  scan size:  [28 28]
abberation fit!
ration 1/30 of the CTF fit, this binning is 7
by 7
a is done!
/pypty/tcbf.py:304: FutureWarning: cupy.fft.c
ache is experimental. The interface can chang

lear_plan_cache() ## free the memory
```



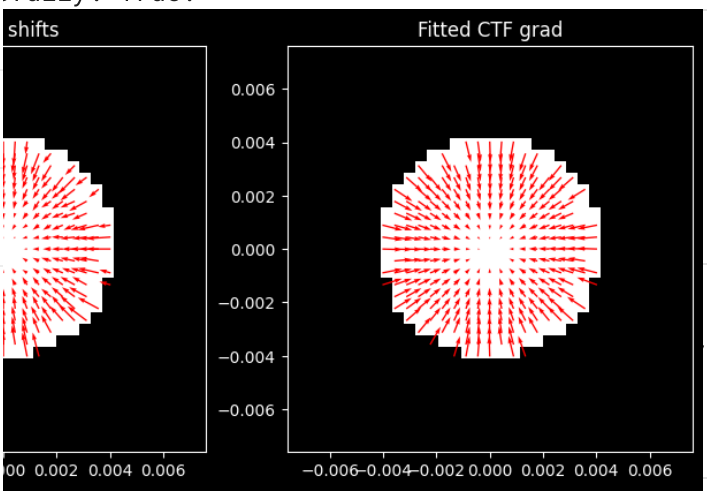
robe modes. A
ifying mode orders
s. If `None`, no
pplied.

t defocus values
nodes. Useful for
pread in the beam.

coefficients in
g., `C10, C12a,`
`C23a, C23b, C30`.

pplied in
Angstroms. Useful for adjusting initial
beam focus in **multislice reconstructions**.

```
ifts for 258/287 pixels.
sfully: True.
```



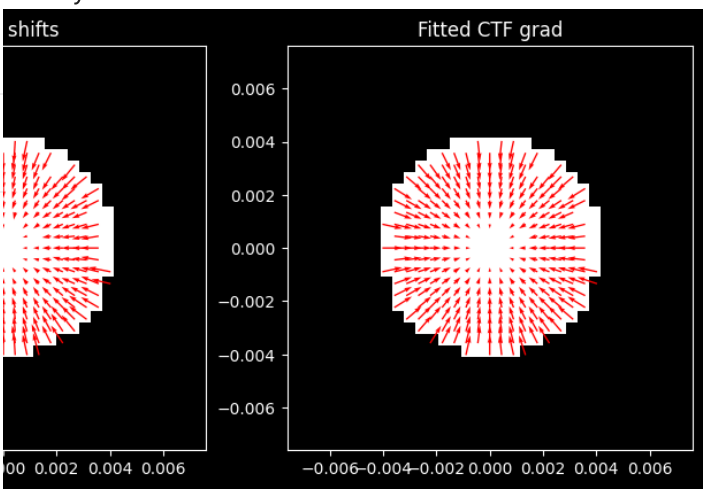

```
s: C10 -2.09e+04 A, C12a 2.04e+02 A, C12b 6.5e
   ay or None
ration 2/30 of the CTF fit, this binning is 7
previous binning
/pypty/tcbf.py:588: FutureWarning: cupy.fft.c
ache is experimental. The interface can chang
```

Mean diffraction pattern. Used for probe initialization. If provided, the probe is created using an **inverse Fourier transform** of this pattern.

lear_plan_cache()

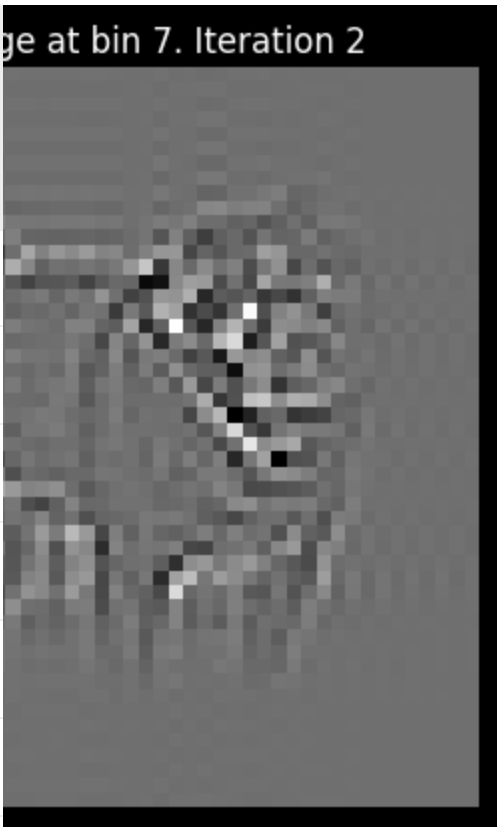
ifts for 258/287 pixels.

sfully: True.



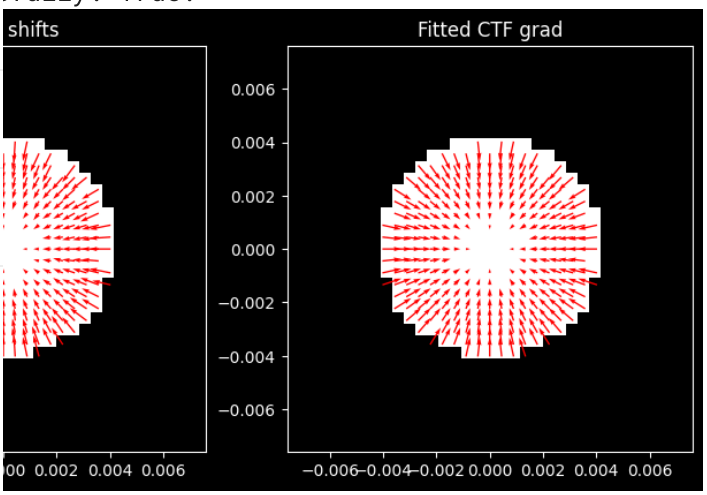
s: C10 -2.23e+04 A, C12a 2.79e+00 A, C12b 1.2

ration 3/30 of the CTF fit, this binning is 7
previous binning



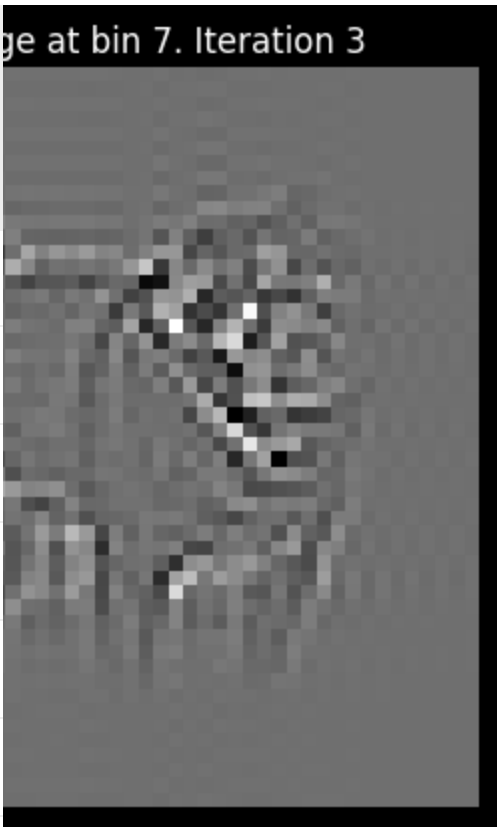
ifts for 258/287 pixels.

sfully: True.



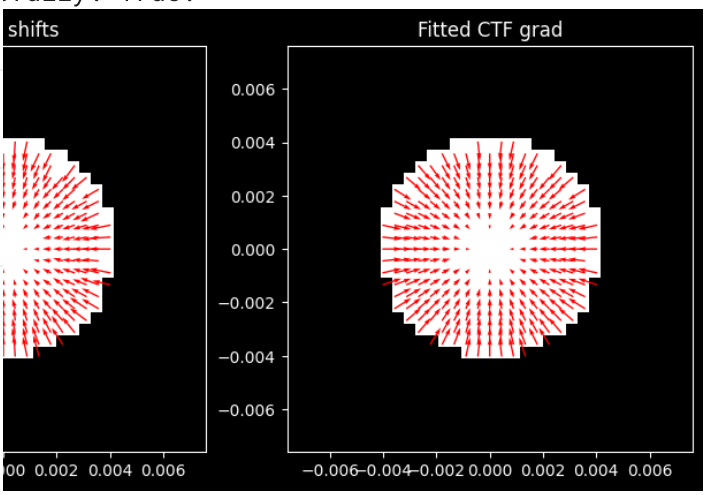
s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 4/30 of the CTF fit, this binning is 7
previous binning



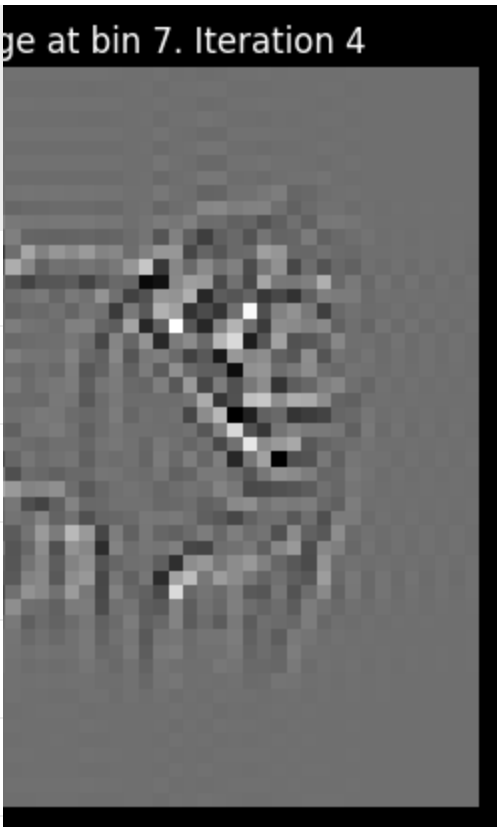
ifts for 258/287 pixels.

sfully: True.



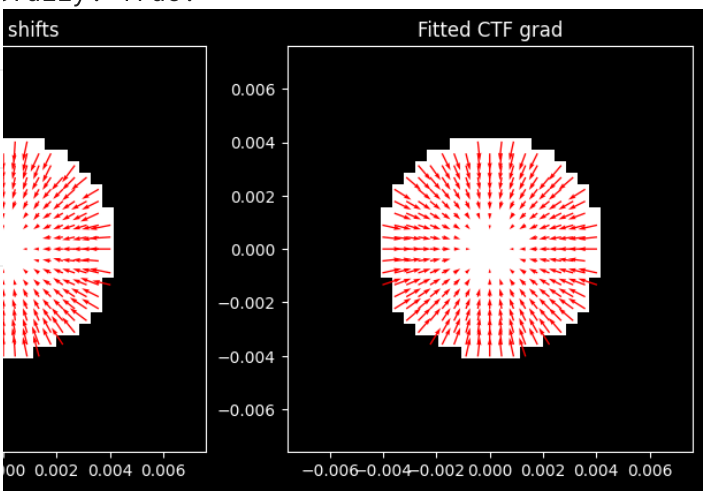
s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 5/30 of the CTF fit, this binning is 7
previous binning



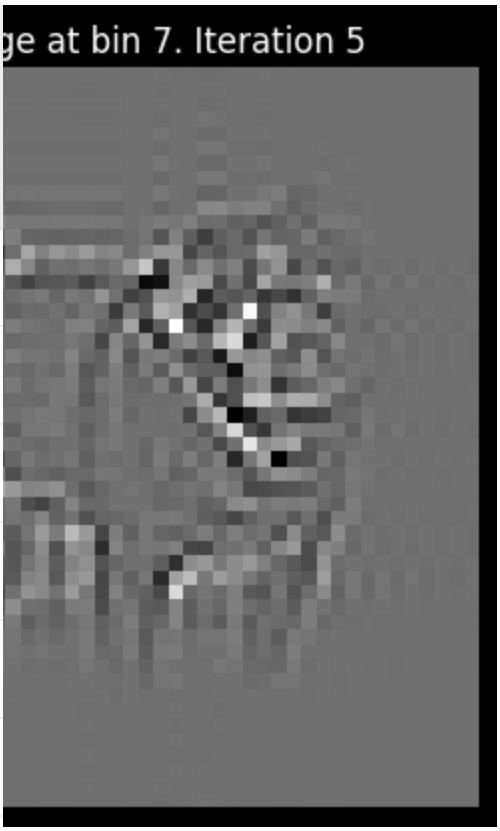
ifts for 258/287 pixels.

sfully: True.



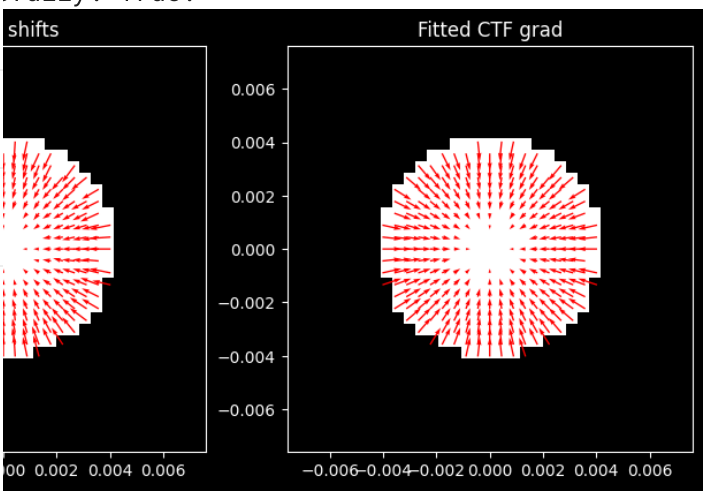
s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 6/30 of the CTF fit, this binning is 7
previous binning



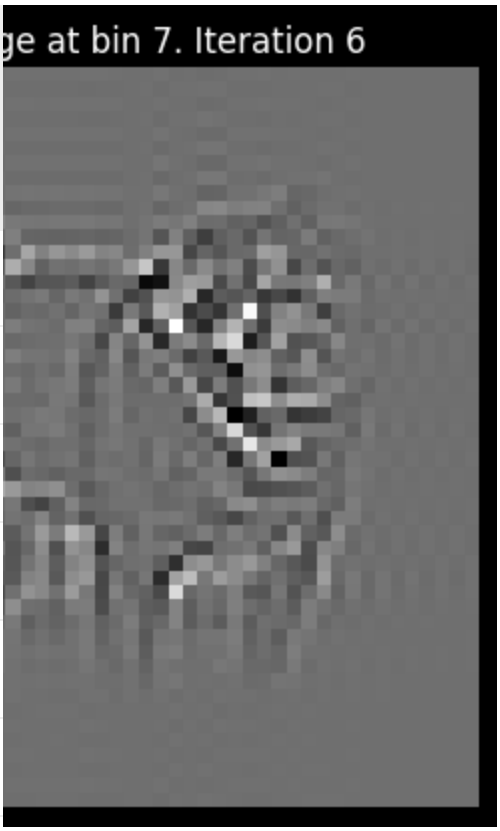
ifts for 258/287 pixels.

sfully: True.



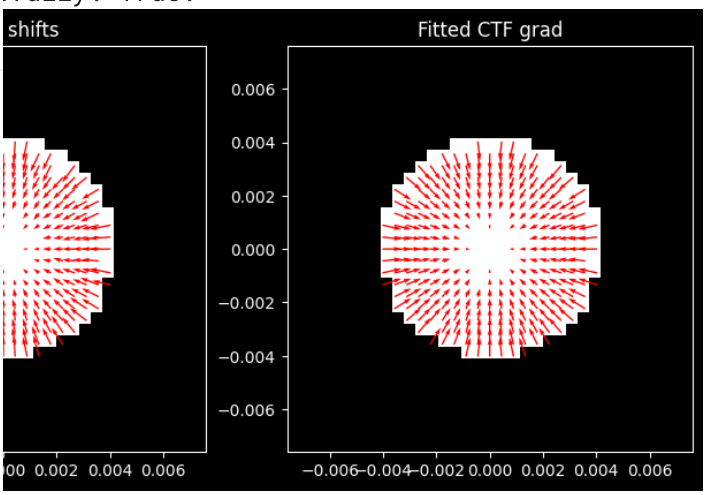
s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 7/30 of the CTF fit, this binning is 7
previous binning



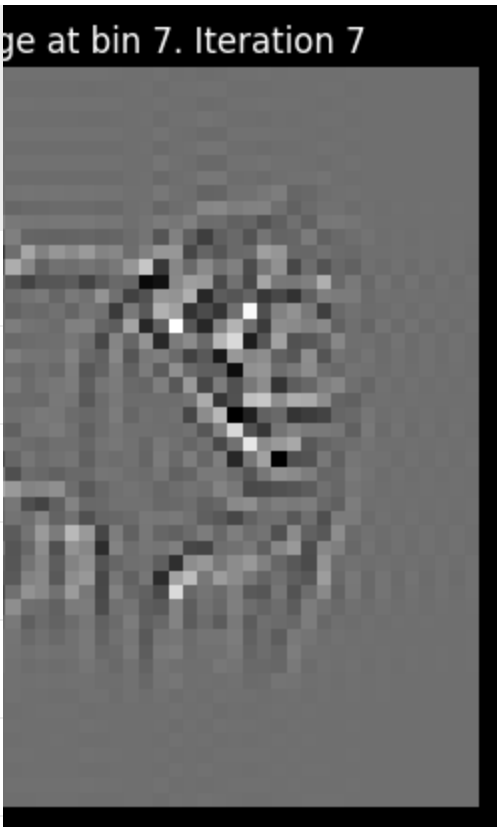
ifts for 258/287 pixels.

sfully: True.



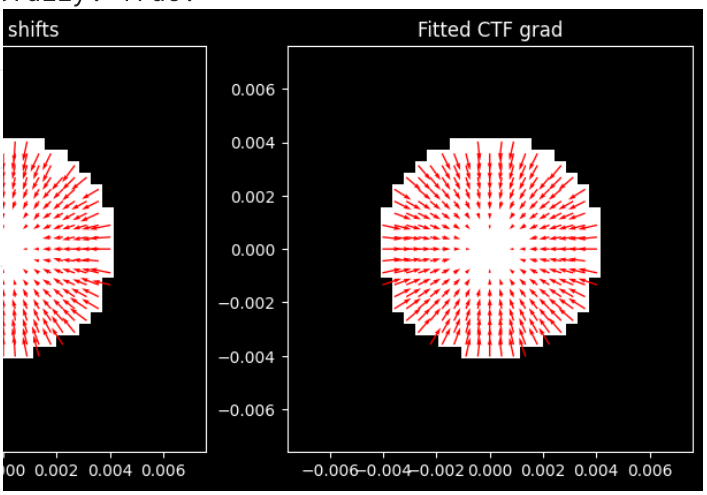
s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 8/30 of the CTF fit, this binning is 7
previous binning



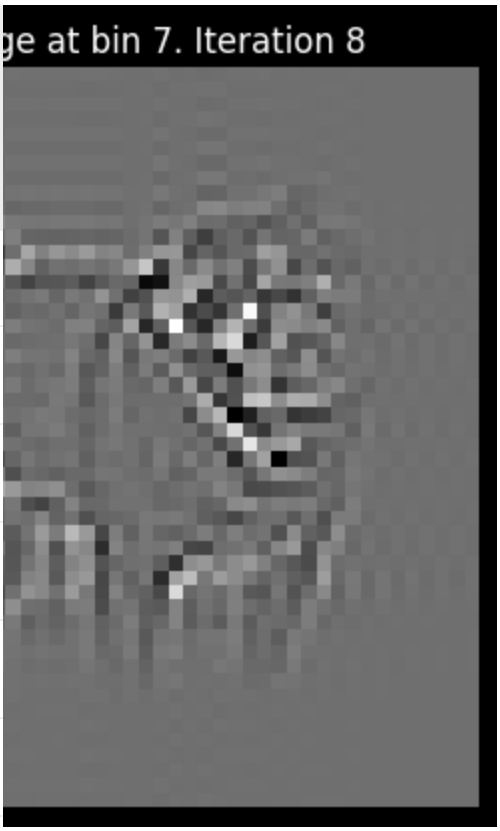
ifts for 258/287 pixels.

sfully: True.



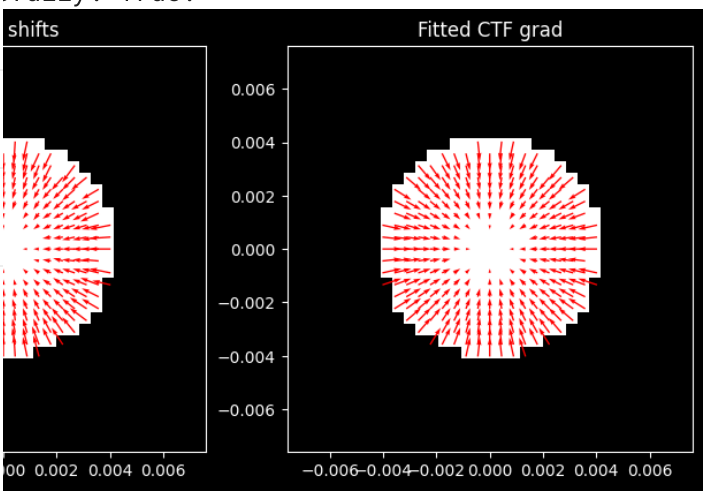
s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 9/30 of the CTF fit, this binning is 7
previous binning



ifts for 258/287 pixels.

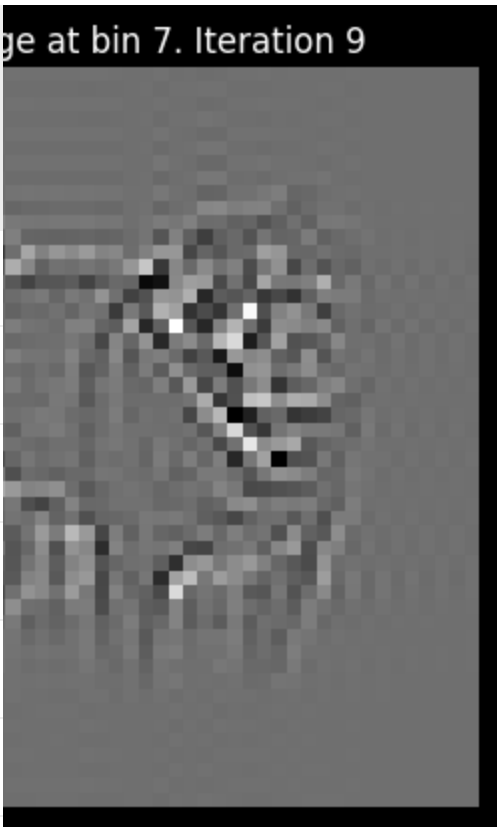
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

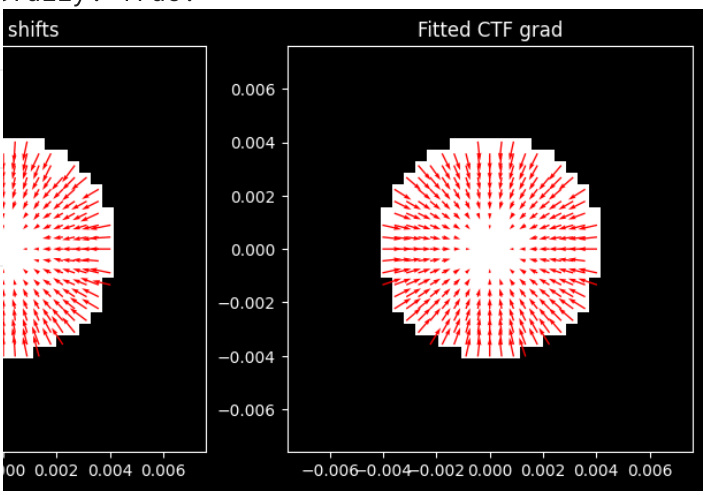
ration 10/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

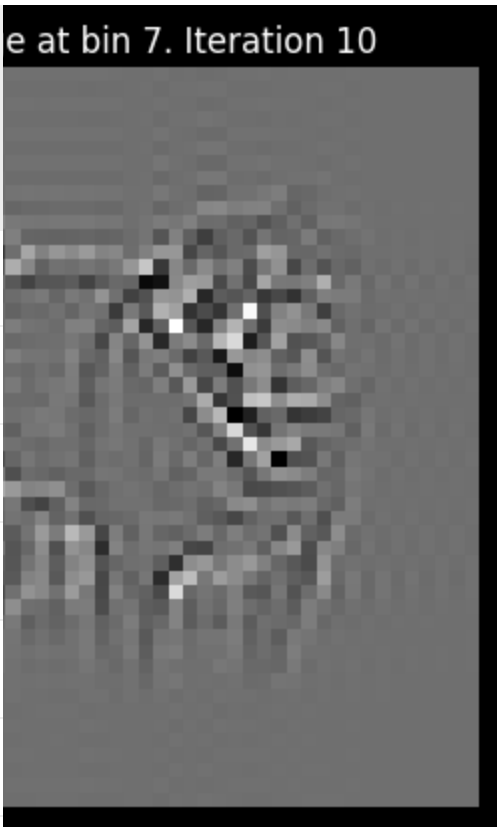
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

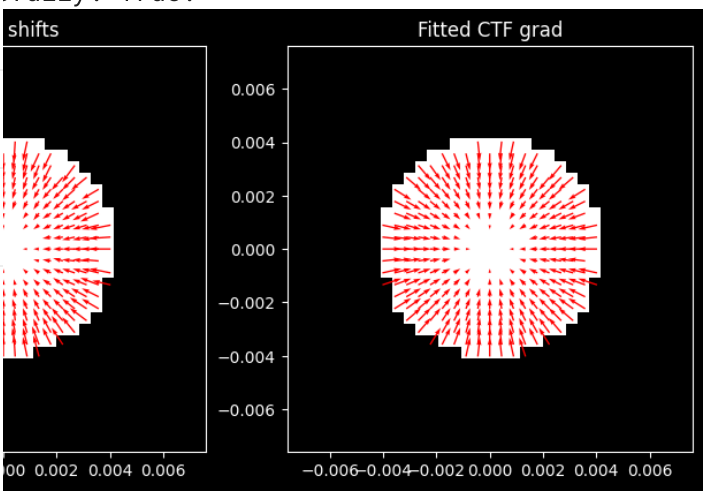
ration 11/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

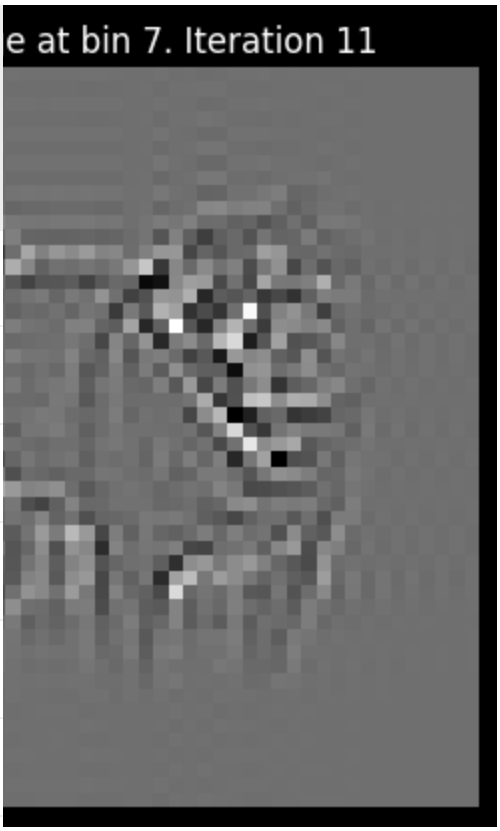
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

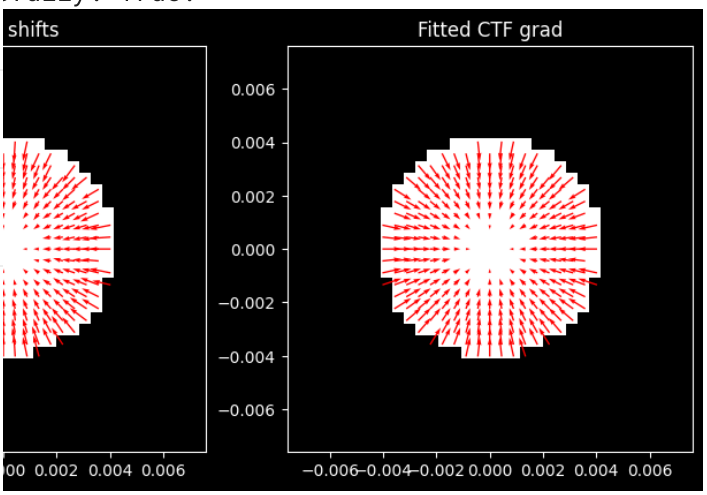
ration 12/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

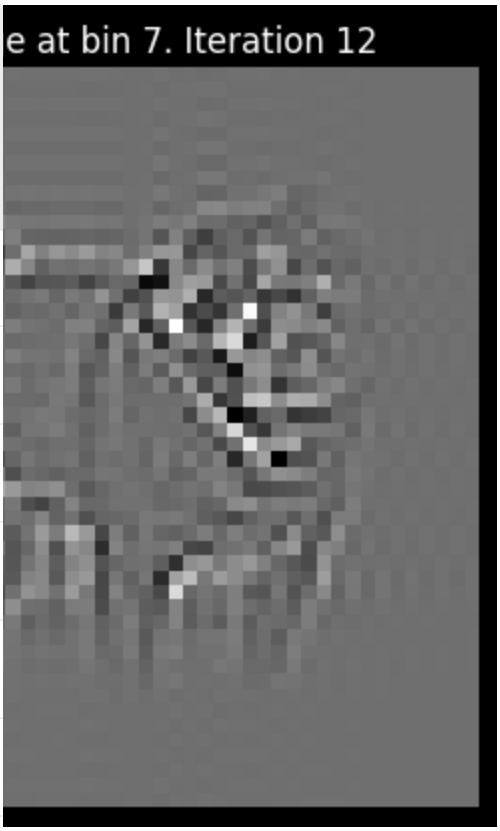
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

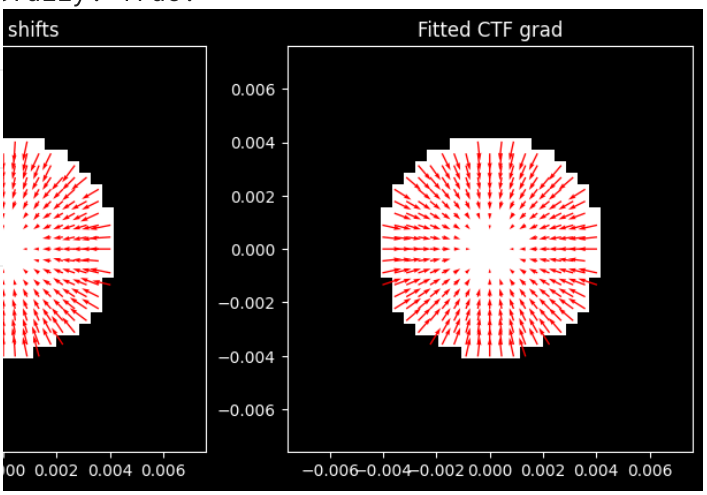
ration 13/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

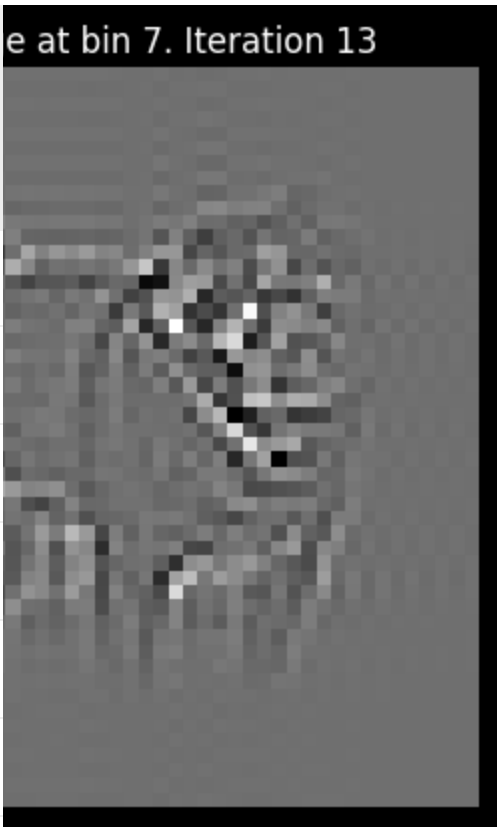
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

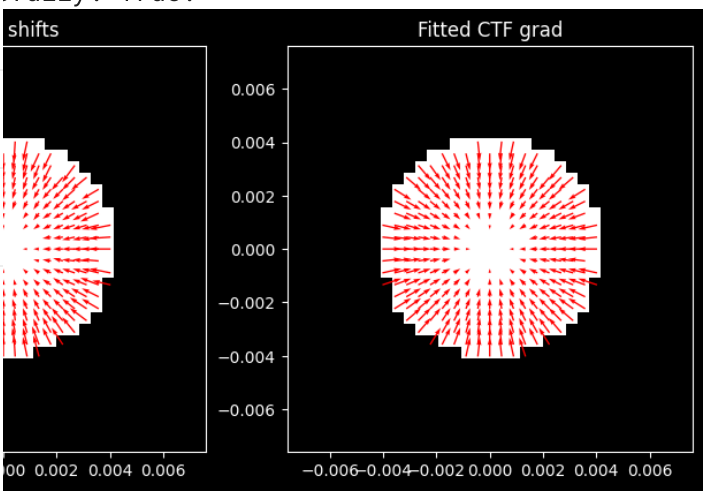
ration 14/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 15/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

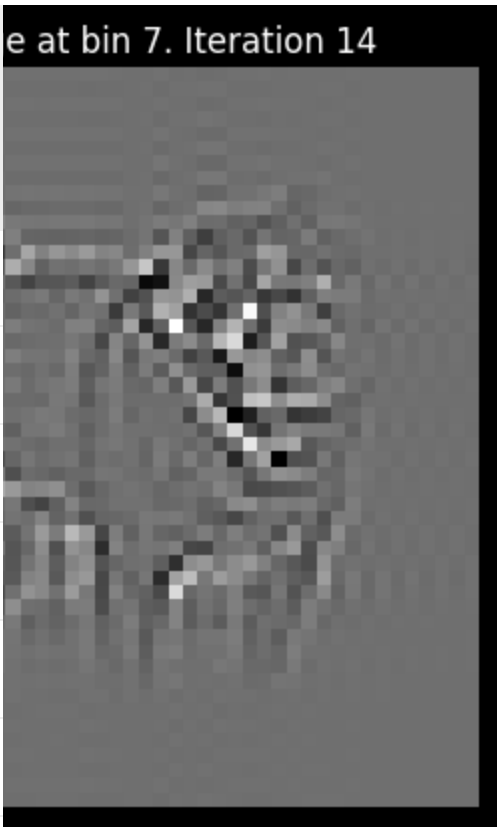
pypty.multislice_core

pypty.fft

pypty.utils

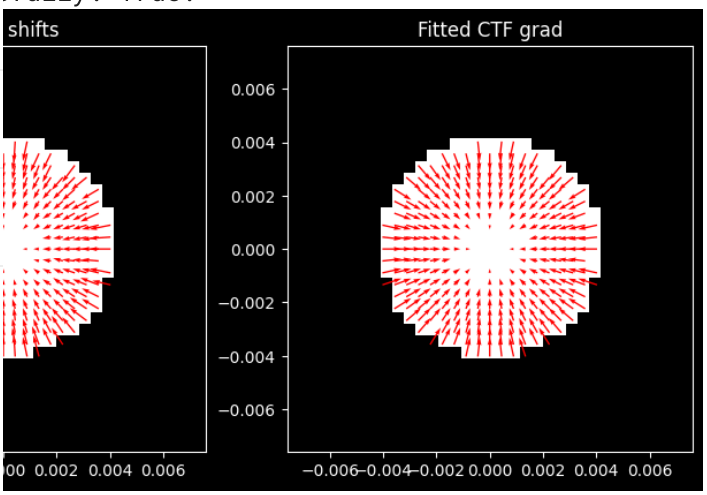
pypty.se

pypty.vaa



ifts for 258/287 pixels.

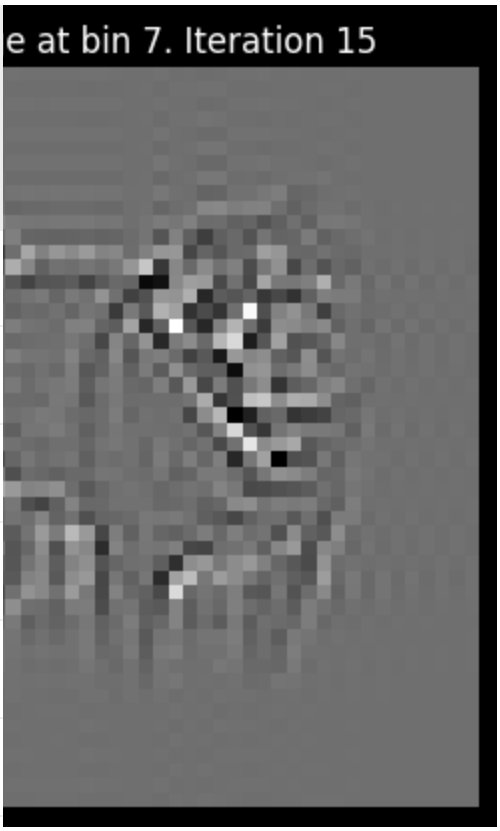
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

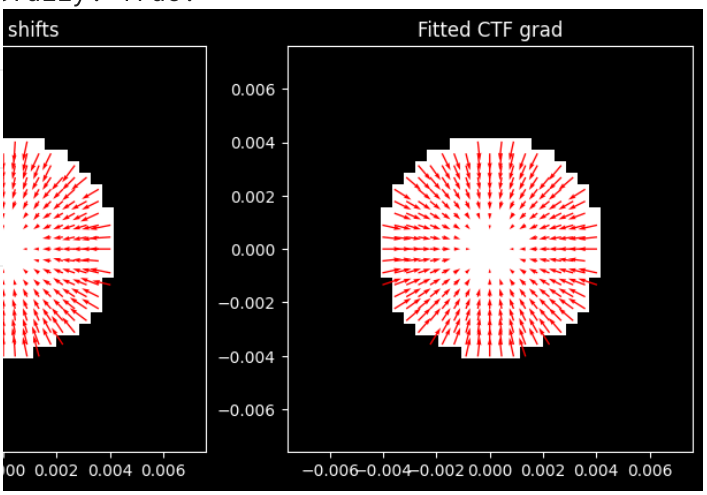
ration 16/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 17/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

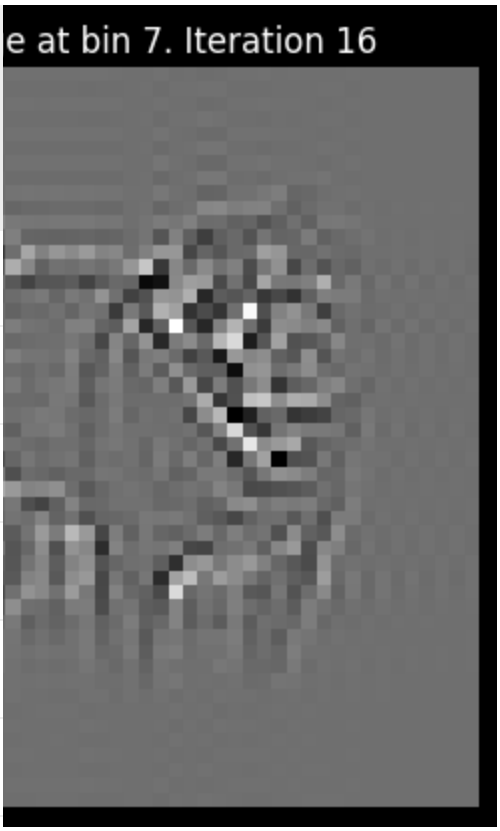
pypty.multislice_core

pypty.fft

pypty.utils

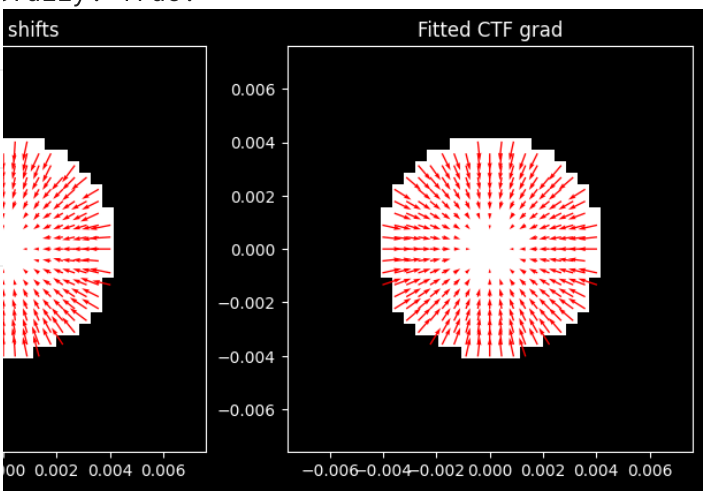
pypty.se

pypty.vaa



ifts for 258/287 pixels.

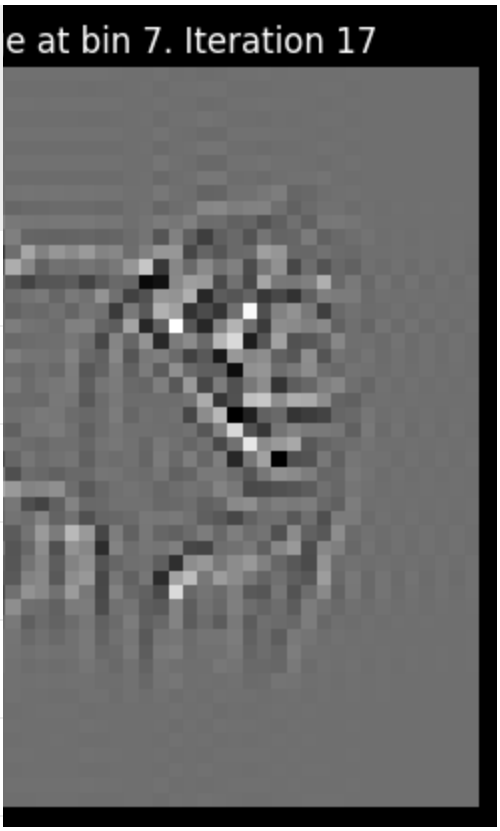
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

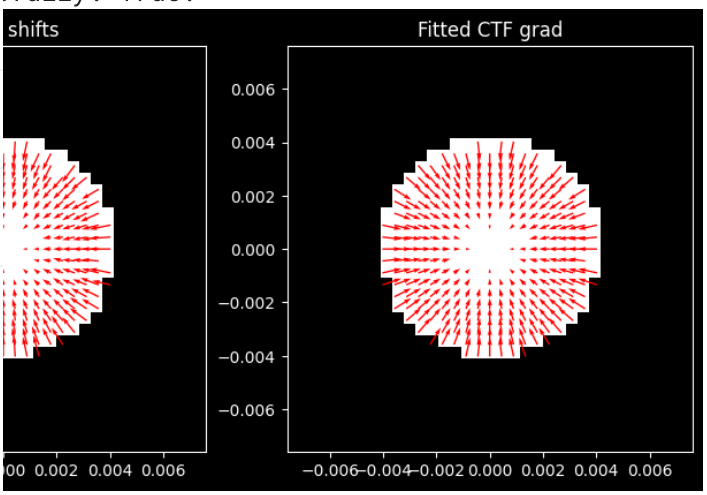
ration 18/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

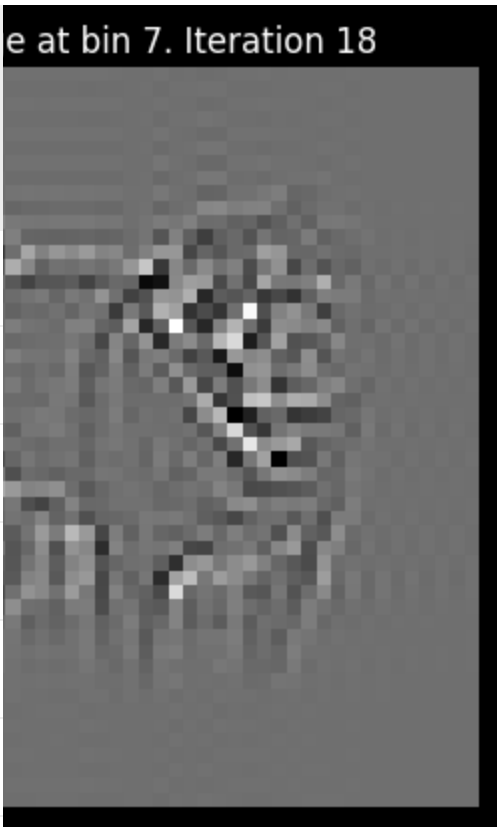
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

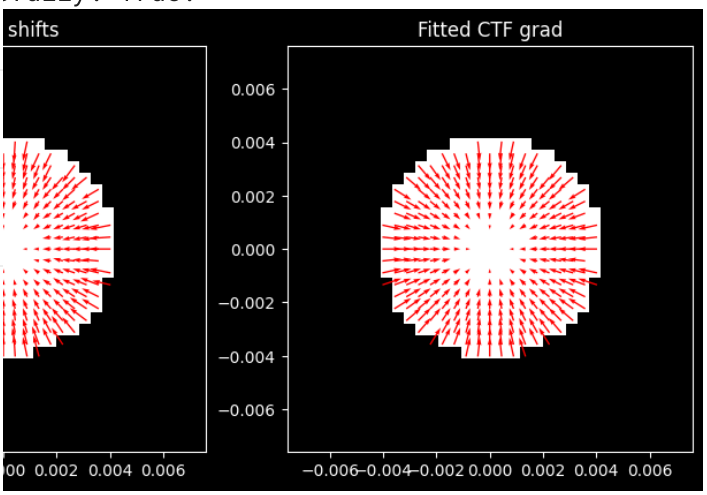
ration 19/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

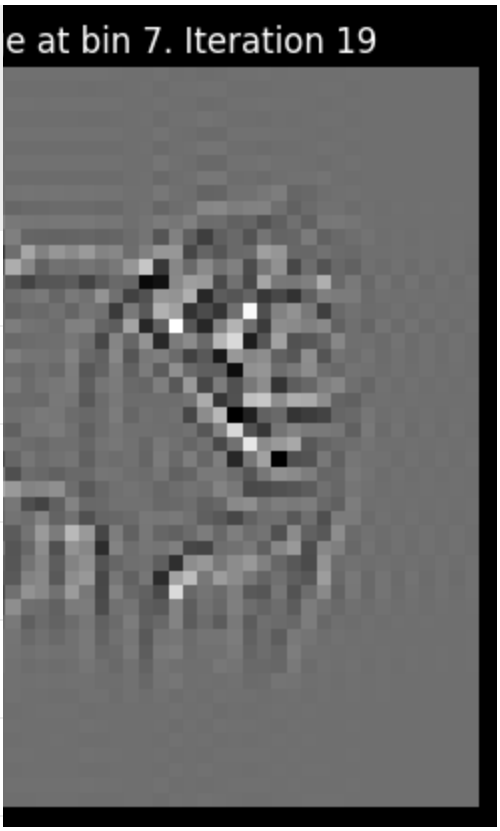
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

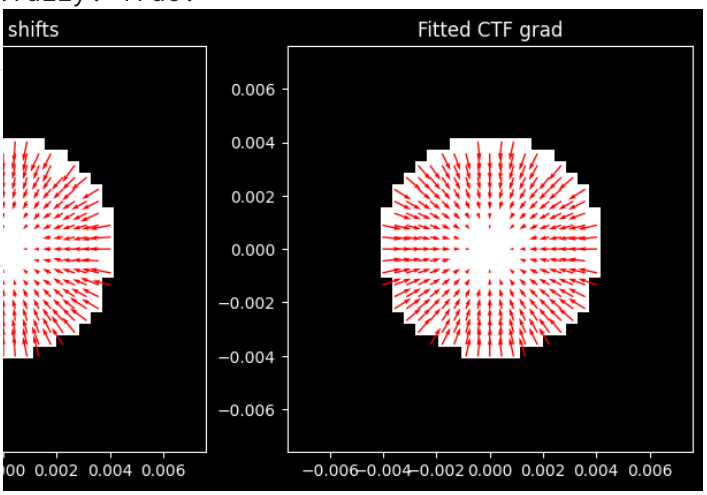
ration 20/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

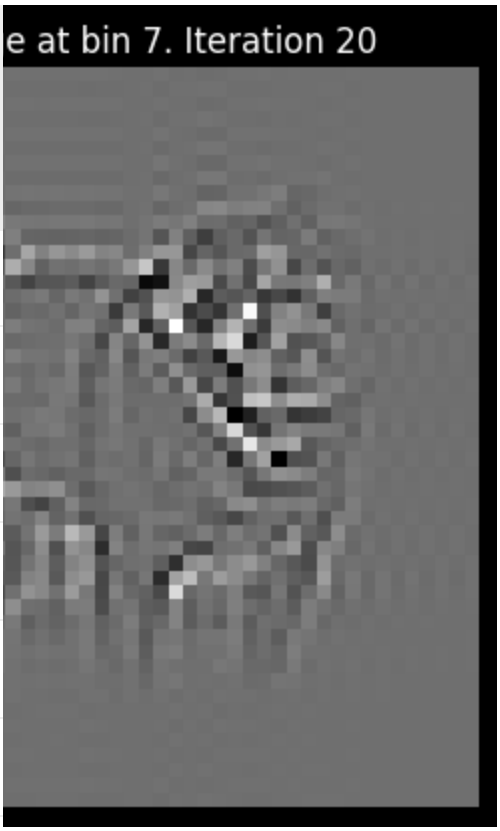
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

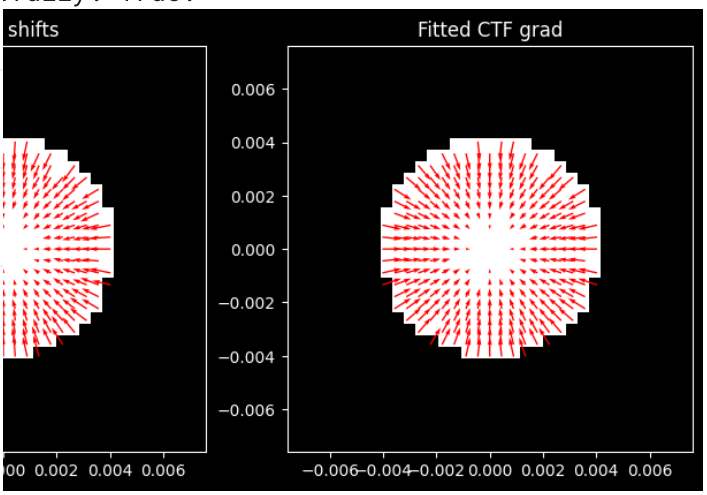
ration 21/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

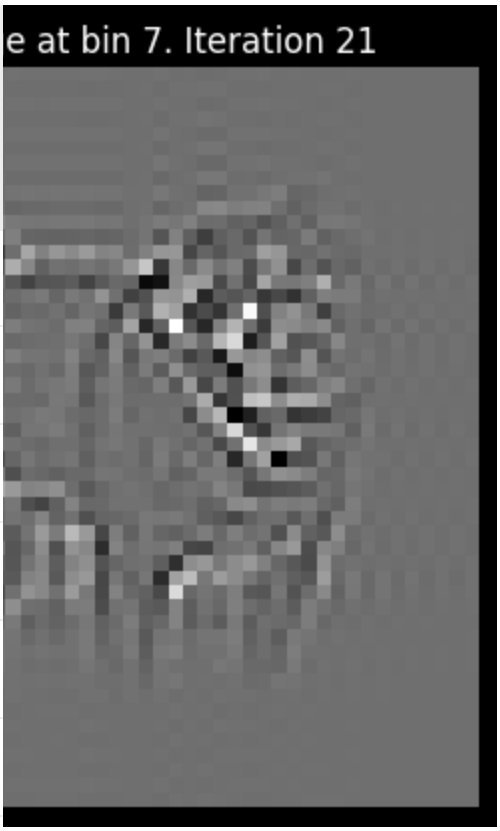
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

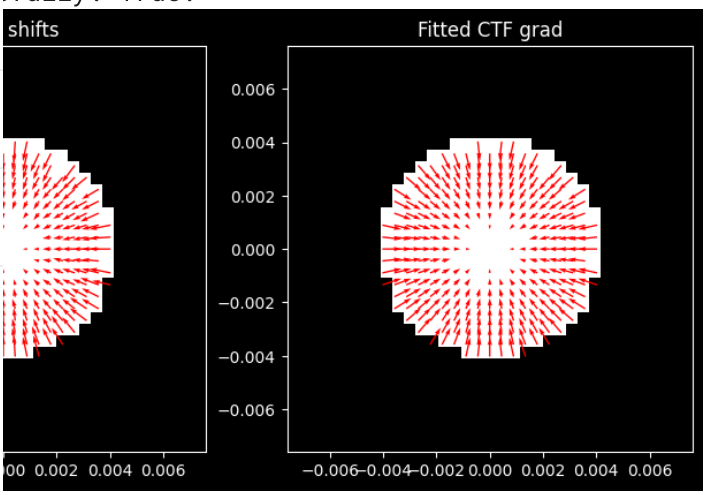
ration 22/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 23/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

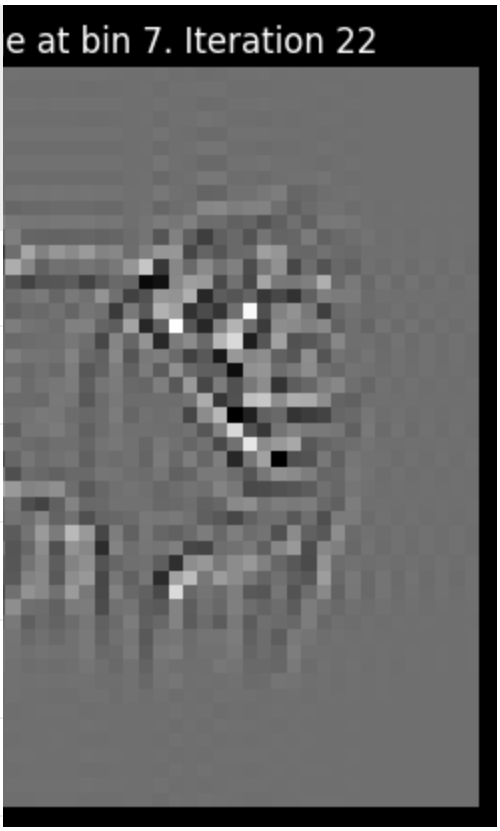
pypty.multislice_core

pypty.fft

pypty.utils

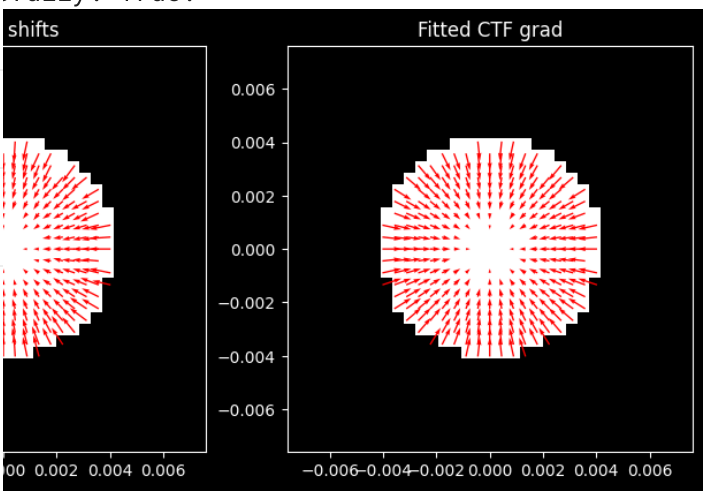
pypty.se

pypty.vaa



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 24/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

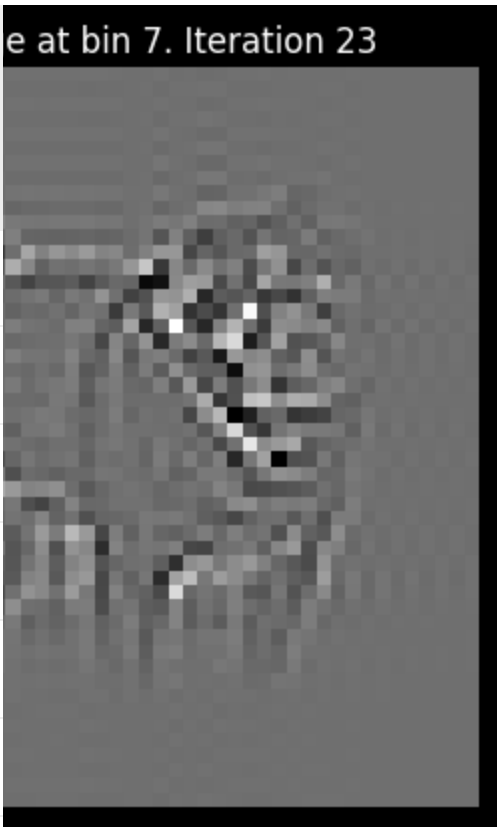
pypty.multislice_core

pypty.fft

pypty.utils

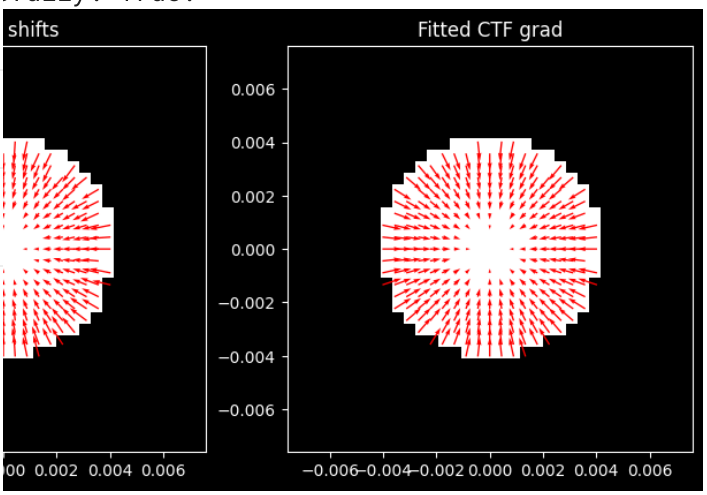
pypty.se

pypty.vaa



ifts for 258/287 pixels.

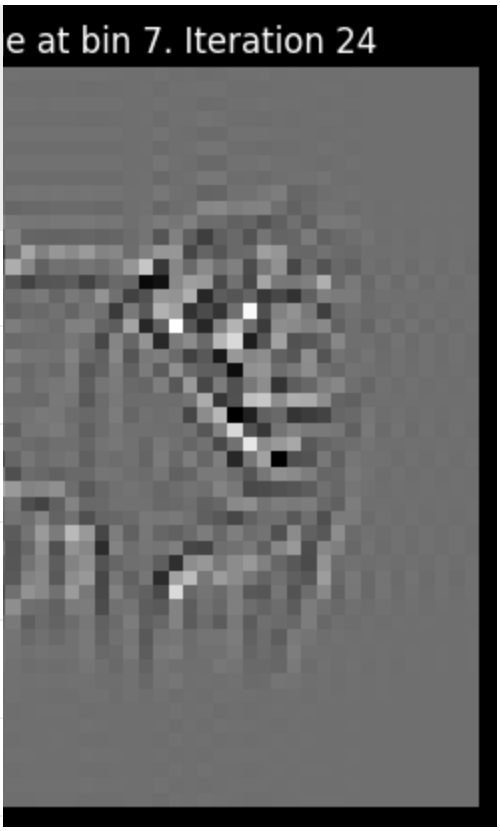
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

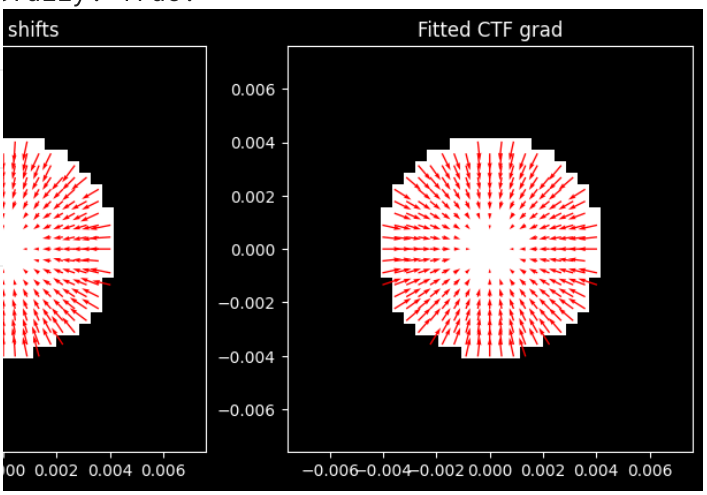
ration 25/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 26/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

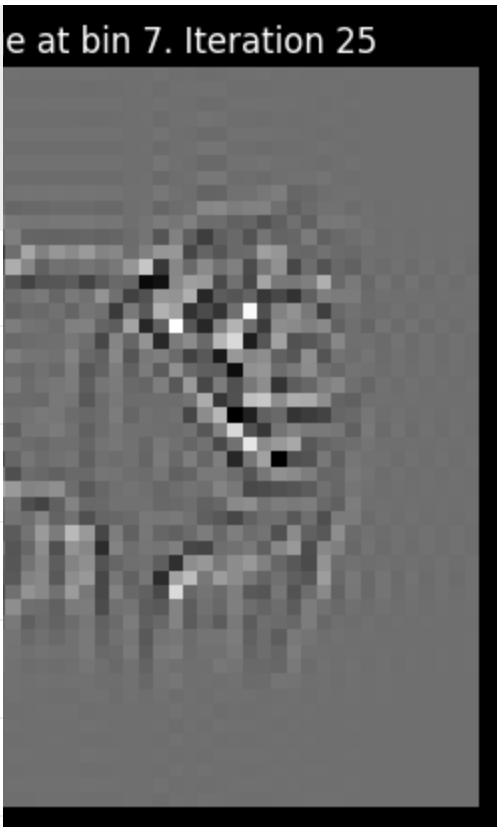
pypty.multislice_core

pypty.fft

pypty.utils

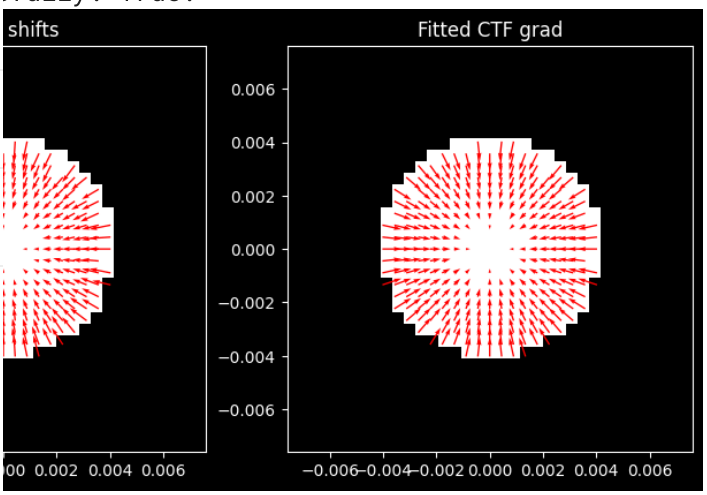
pypty.se

pypty.vaa



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 27/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

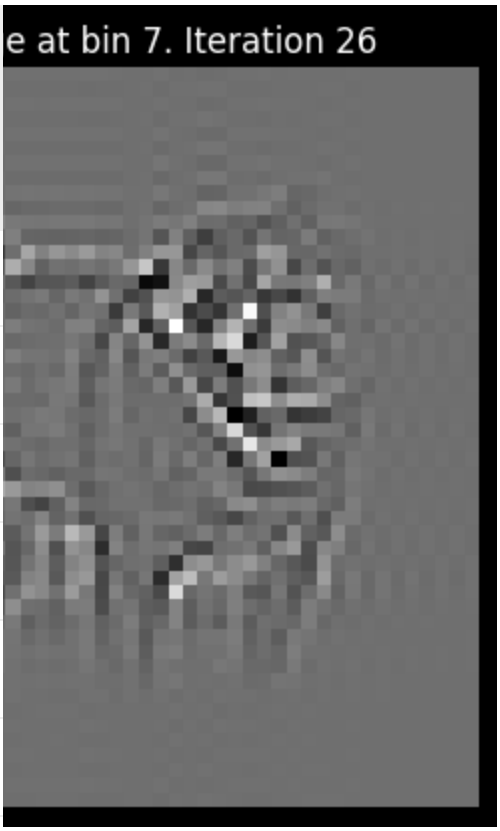
pypty.multislice_core

pypty.fft

pypty.utils

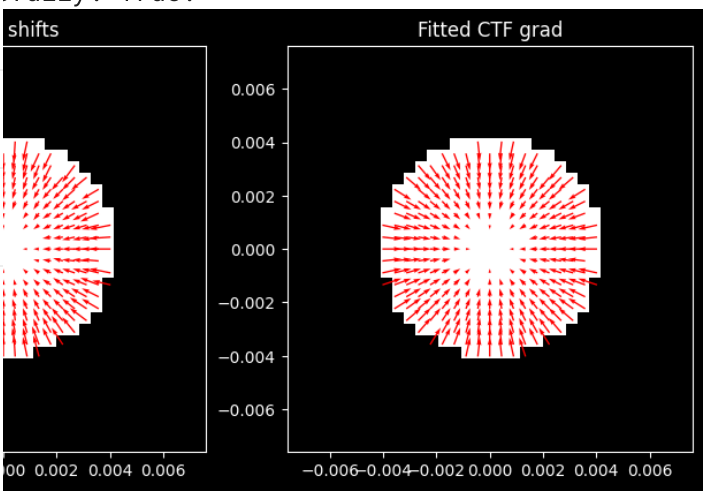
pypty.se

pypty.vaa



ifts for 258/287 pixels.

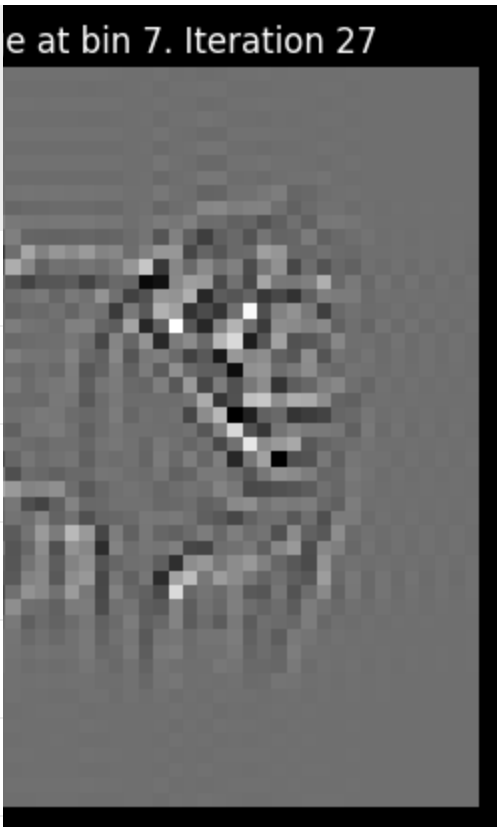
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

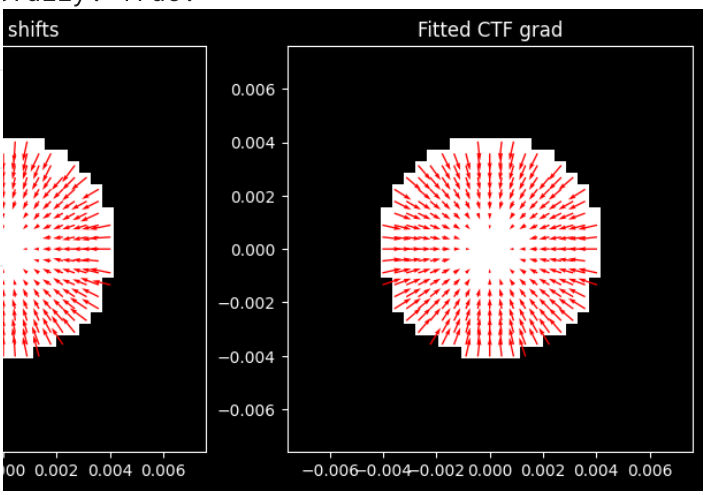
ration 28/30 of the CTF fit, this binning is

previous binning



ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

ration 29/30 of the CTF fit, this binning is

previous binning

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

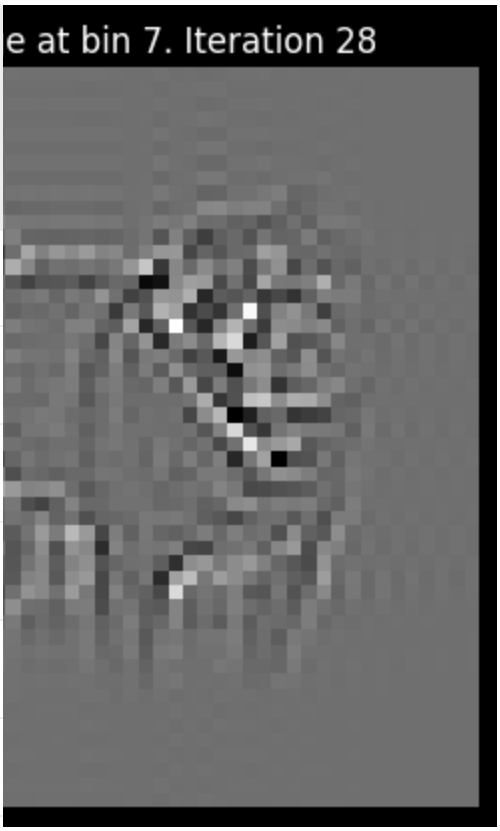
pypty.multislice_core

pypty.fft

pypty.utils

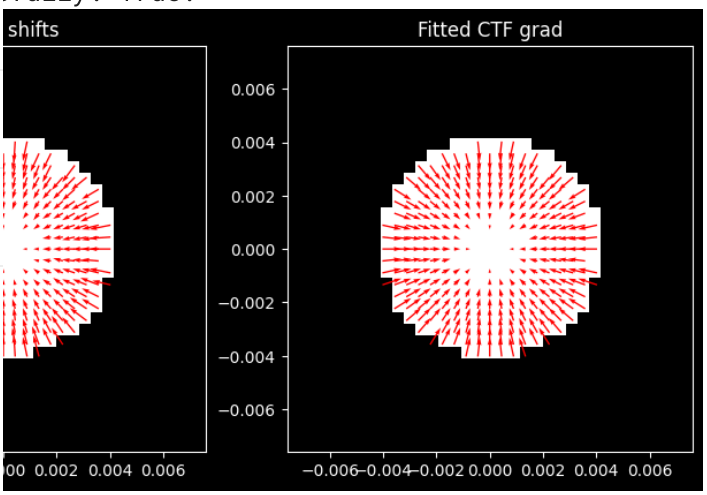
pypty.se

pypty.vaa



ifts for 258/287 pixels.

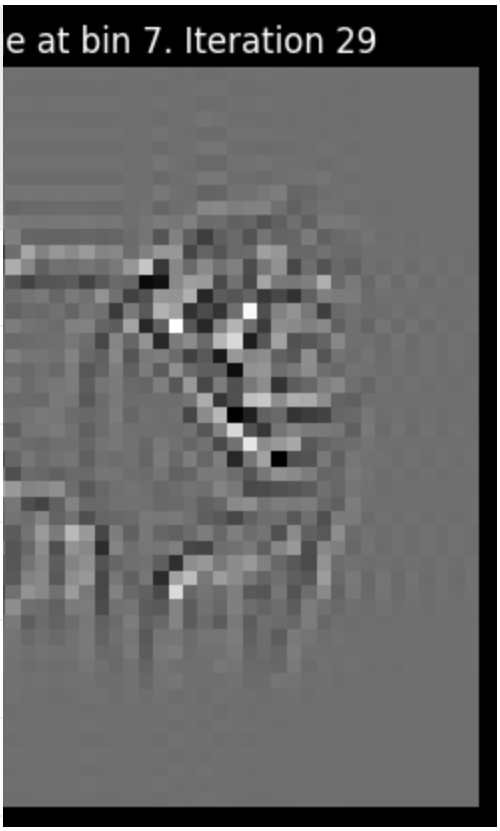
sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

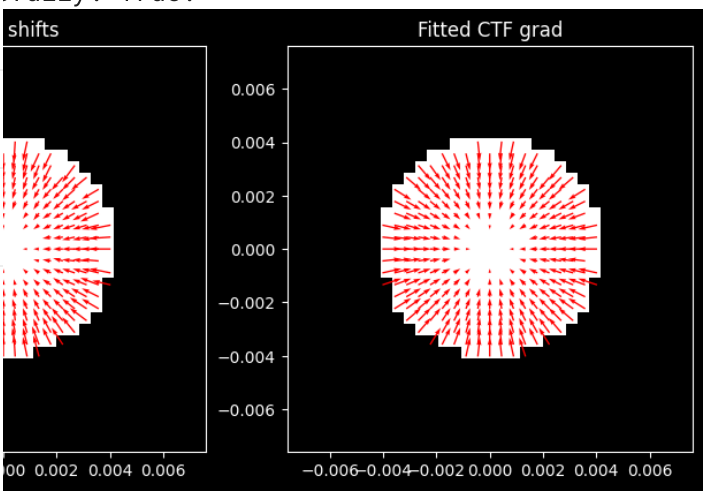
ration 30/30 of the CTF fit, this binning is

previous binning



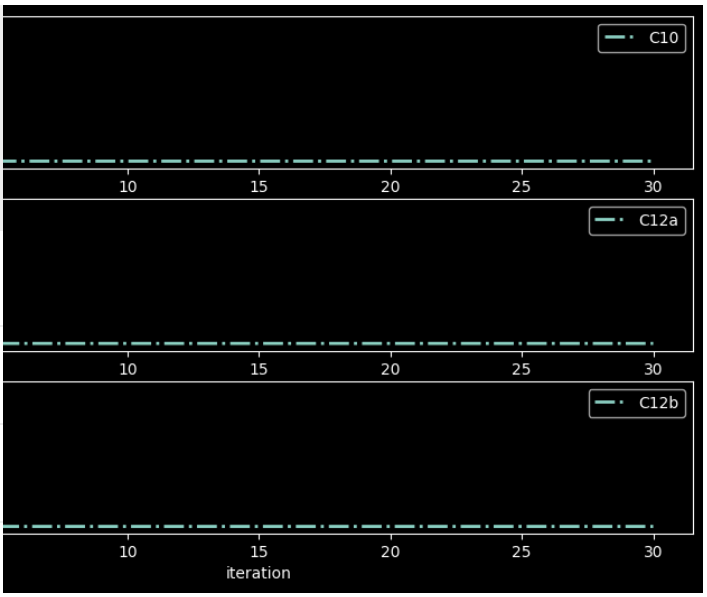
ifts for 258/287 pixels.

sfully: True.



s: C10 -2.23e+04 A, C12a 7.74e-01 A, C12b -1.

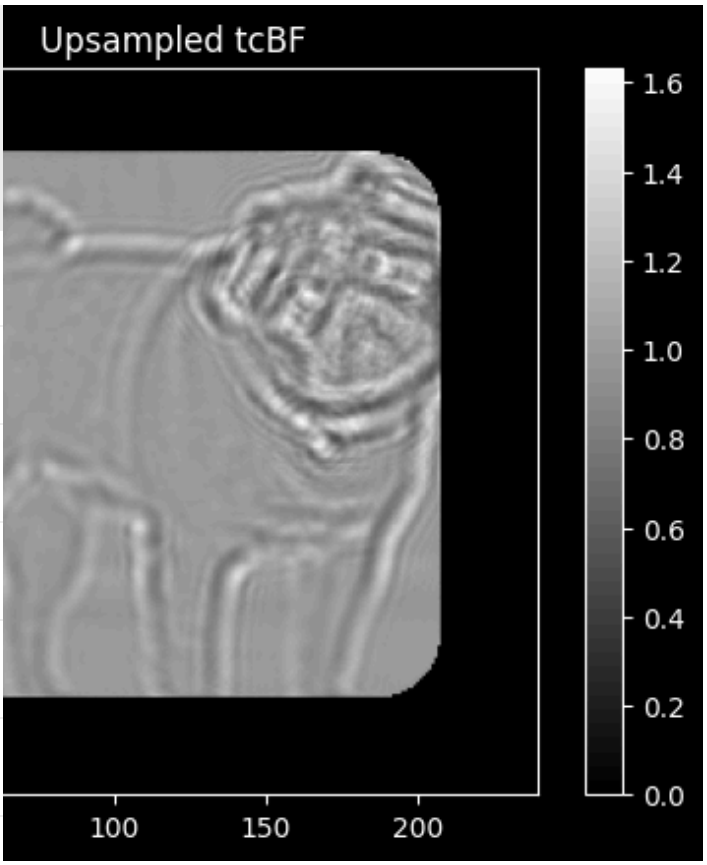
e!




```
/pypty/tcbf.py:625: FutureWarning: cupy.fft.c
ache is experimental. The interface can chang

lear_plan_cache()
```

```
cbf.upsampled_tcbf(pypty_params,
        round_shifts=True,
        upsample=5)

cmap="gray")
mpled tcBF")
```

```
**********************************************

****** Creating upsampled tcBF Image *******

**********************************************


pixel
size will be 4.00 Å
age will be: (240, 240)
12281/12281 [00:02<00:00, 4569.58it/s]
/pypty/tcbf.py:856: FutureWarning: cupy.fft.c
ache is experimental. The interface can chang

lear_plan_cache()
```



/

```
wdd(pypty_params)

ngle(o), cmap="gray")
Phase")
```

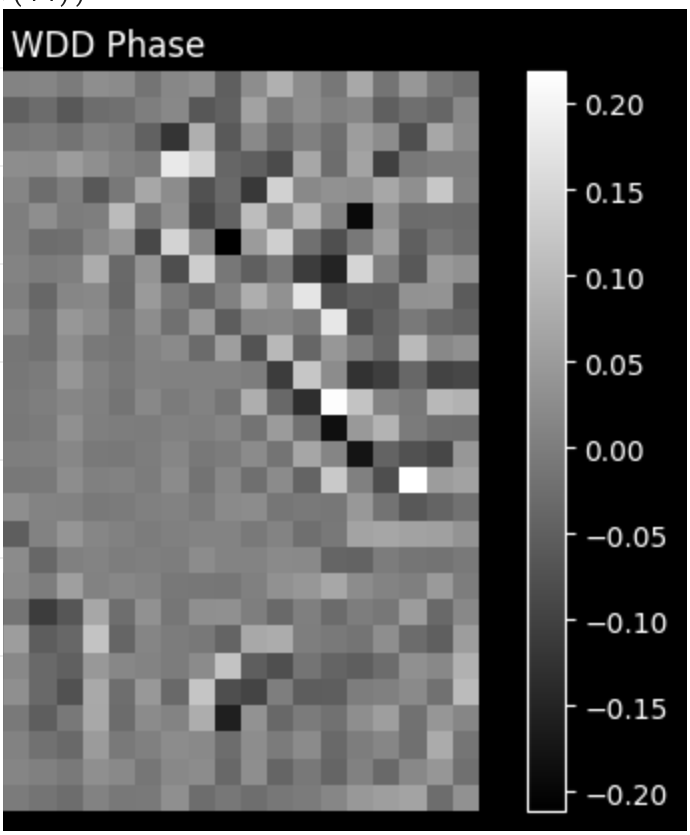
```
/pypty/direct.py:46: FutureWarning: cupy.fft.
cache is experimental. The interface can chan

lear_plan_cache()
ons: C10: -2.23e+04 Å; C12a: 7.74e-01 Å; C12
```

```
/pypty/direct.py:170: FutureWarning: cupy.ff
n_cache is experimental. The interface can ch
```

```
e.
lear_plan_cache()
256/256 [00:06<00:00, 39.32it/s]
/pypty/direct.py:182: FutureWarning: cupy.ff
n_cache is experimental. The interface can ch
e.
lear_plan_cache()
```

```
28/28 [00:00<00:00, 51.31it/s]
/pypty/direct.py:201: FutureWarning: cupy.ff
n_cache is experimental. The interface can ch
e.
lear_plan_cache()
```

```
(14))
```



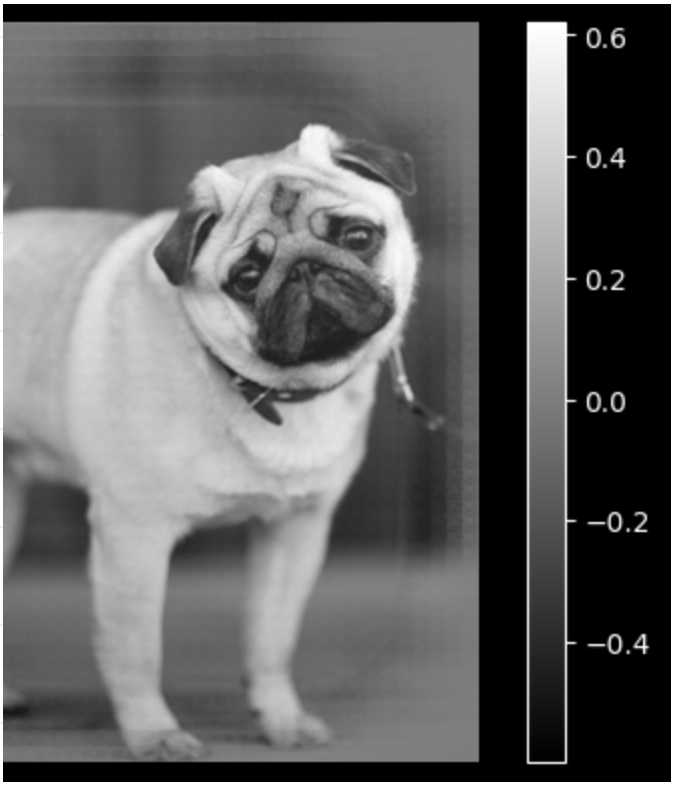
hy

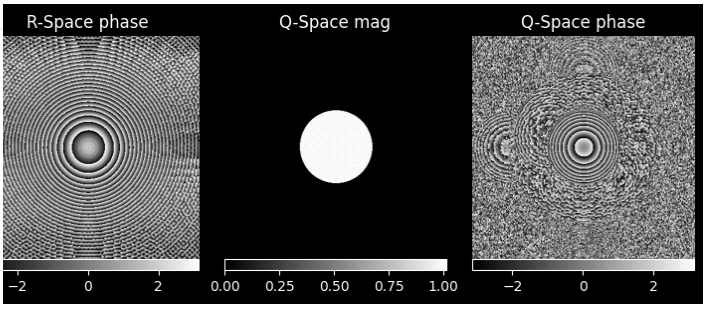
```
.run(pypty_params)
```

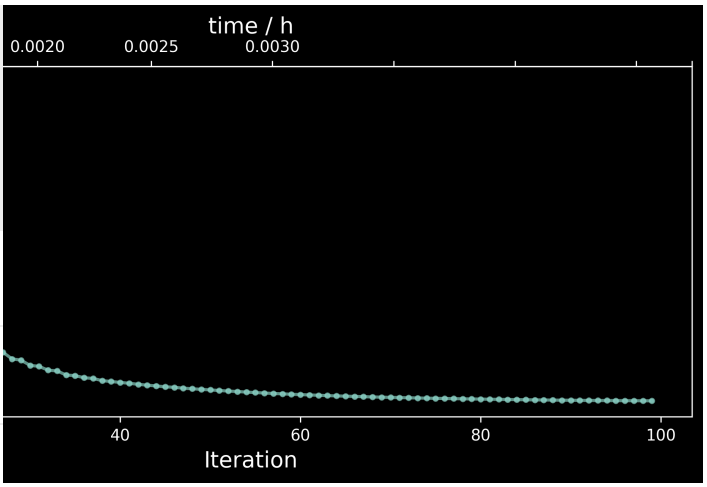
```
********** ********
****** Starting PyPty Reconstruction *******
**********

et: /home/anton/data/test_folder/mops.h5
s in /home/anton/data/test_folder/test_mops_

ters in /home/anton/data/test_folder/test_mop

be saved as /home/anton/data/test_folder/tes

**********
**********
**********

erated based on the mean pattern!
specified scan positions was larger than the
t, adding ones to the right and/or bottom. N
bject is 913 px in y, 913 px in x, 1 slice(-
!!!

ons: C10: -2.23e+04 Å; C12a: 7.74e-01 Å; C12

0: 0:17. Epoch 99. Using lsq_sqrt error metri
timzer. Loss: 5.07e+01. SSE: 7.34e+01. Updati

/pypty/iterative.py:336: FutureWarning: cupy.
lan_cache is experimental. The interface can
ure.
lear_plan_cache()
```

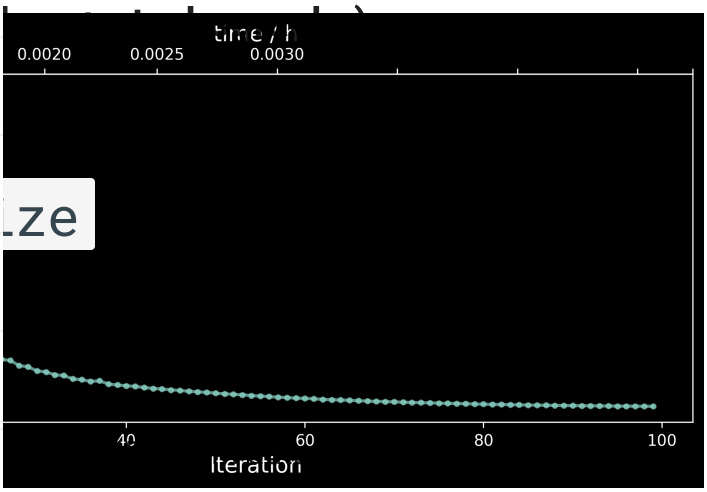
```
/home/anton/data/test_folder/test_mops_2/"
utput_folder+"co.npy"), np.load(output_folder
```

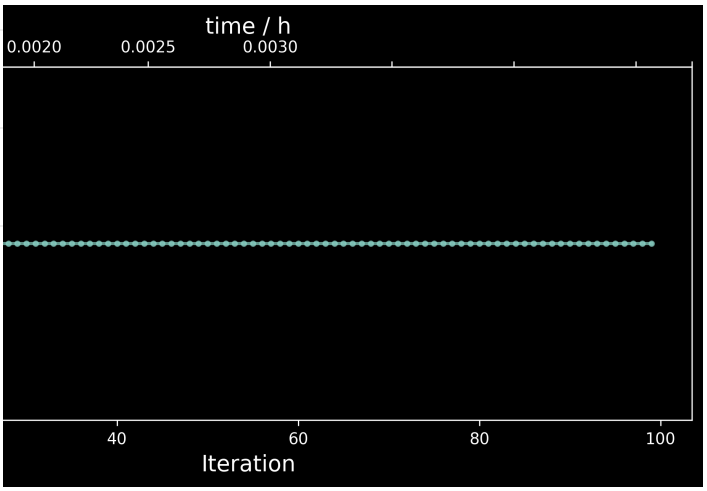
```
o[:,:,0,0])

e, cmap="gray")


modes(p)

outputlog_plots(output_folder+"loss.csv", 10,
```











`.npy` data file or a 4D numpy array [scan_y, scan_x, ky, kx].

`darray`

for the PyPty `.h5` dataset.

o axes (kx, ky). Default is False.

DEFAULT: `False`

ng specific axes. Default is False.

DEFAULT: `False`

ng specific axes. Default is False.

DEFAULT: `False`

ng specific axes. Default is False.

DEFAULT: `False`

ng specific axes. Default is False.

DEFAULT: `False`

rns to use to estimate center-of-mass. Default is 1000.

DEFAULT: `1000`

er-of-mass. If None, it will be computed.

`lone` **DEFAULT:** `None`

er-of-mass. If None, it will be computed.

`lone` **DEFAULT:** `None`

actor on the diffraction patterns. Default is 1.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DEFAULT: `1`

atterns. Defaults are None.

lone **DEFAULT:** `None`

atterns. Defaults are None.

lone **DEFAULT:** `None`

atterns. Defaults are None.

lone **DEFAULT:** `None`

atterns. Defaults are None.

lone **DEFAULT:** `None`

rn sums to 1. Default is False.

DEFAULT: `False`

farther than `cutoff_ratio × max_radius`. Default is None.

` None` **DEFAULT:** `None`

/ to diffraction patterns. Default is 0.

DEFAULT: `0`

. Default is np.float32.

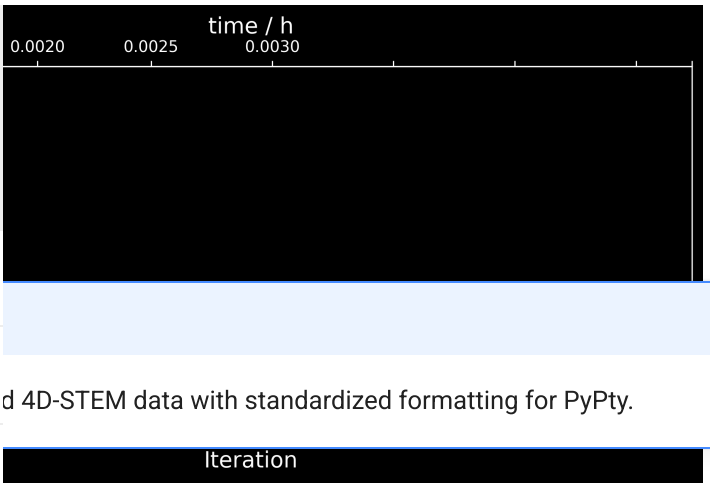
DEFAULT: `float32`

intensity. Default is 1.

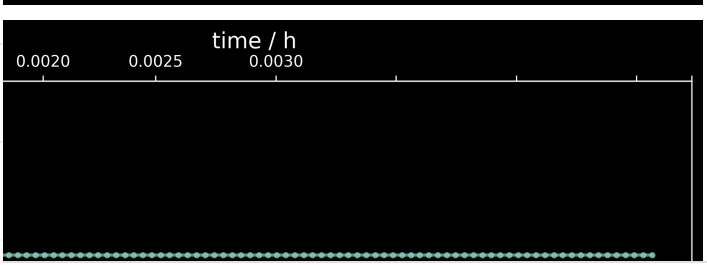
DEFAULT: `1`

ng if file exists. Default is True.

DEFAULT: `True`



d 4D-STEM data with standardized formatting for PyPty.



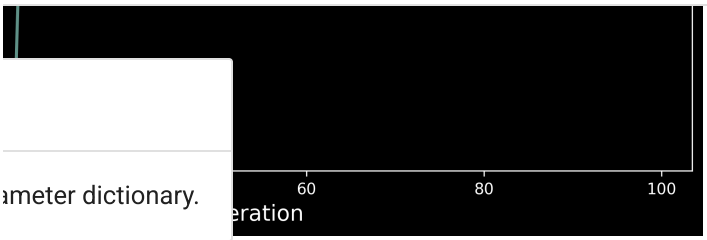
PTION

ry containing experimental metadata and setup for PyPty uction.

dict

PyPty preset to update, a filepath to a preset, or None to new one.

dict or str or None **DEFAULT:** None



ameter dictionary.

following entries:

```
le 3d .h5 file [N_measurements, ky,kx] or .npy Nion-st
 if data is compressed provide the virtual detectors (
utfolder where the results will be stored
e json file with metadata (optional)
ng voltage in kV
g callibrations:
al pixel size in Å^-1
rocal pixel size in mrad
convergence semi-angle in mrad
ry 2D mask
d to estimate an aperture, everything above threshold
e padding. If None (default), pading is 1/4 of the tot
1 (no upsampling)
array containing beam aberrations (in Å). Aberrations
ra probe defocus besides the one contained in aberrati
number of scan points along slow (y) and fast (x) axes
(STEM pixel size) in Å.
ast axis in nm.
rray, default None. If you acquiered a data on a speci
ix for postions transformation
n angle between scan and detector axes. Default None.
is False. If no PyPty-style h5 data was created, this
ss of a sample in Å. Has no effect if num_slices is 1
slices, default is 1.
True
s 1. If 0 nothing will be printed. 1 prints only thela
ean Flag. Default is True.
```

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

scan grid and reconstruction grid. In PyPty
otated to compensate the misalignment between
reconstruction grid is larger than the scanned FOV,
extent of the probe.

SCRIPTION
an dimensions. PE: `int`
an dimensions. PE: `int`
EM scan step size in Å. PE: `float`
ciprocal space pixel size in $Å^{-1}$. PE: `float`
ape of diffraction patterns. PE: `tuple`
tation between scan and detector axes (degrees). PE: `float` **DEFAULT:** `0`

TION
lues (in reconstruction pixels).

ze

nstruction pixel units.

:SCRIPTION

an grid size.

'PE: `int`

an grid size.

'PE: `int`

'EM scan step size (Å).

'PE: `float`

xel size in reciprocal space (Å⁻¹).

'PE: `float`

ape of the diffraction pattern.

'PE: `tuple`

an-detector rotation angle in degrees. Default is 0.

'PE: `float` **DEFAULT:** `0`

p scan axes. Default is False.

'PE: `bool` **DEFAULT:** `False`

p scan axes. Default is False.

'PE: `bool` **DEFAULT:** `False`

int pixel size. Default is False.

'PE: `bool` **DEFAULT:** `False`

tional 2x2 matrix to apply to positions.

'PE: `array_like` **DEFAULT:** `eye(2)`

econstruction pixels.

truction pixel in Å.

`_image`

ed image onto the reconstruction grid.

ixel of an arbitary image (e.g. upsampled tcBF
ponding to a ptychographic reconstruction.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

CRIPTION	
onary of PyPty reconstruction parameters. E: `dict`	
nage (e.g., upsampled tcBF) to map. E: `ndarray`	
size of the image in Å. E: `float`	
offset on left side of image relative to scan grid. Default is 0. E: `int`	**DEFAULT:** `0`
offset on top side of image relative to scan grid. Default is 0. E: `int`	**DEFAULT:** `0`

linates [[y, x], ...] in reconstruction grid units.	

an

s using interpolated phase and amplitude maps. You
:BF reconstructions to generate it.

N

eter dictionary.

ices or "auto" to estimate from max phase shift.

r str **DEFAULT:** None

p to interpolate.

ay **DEFAULT:** None

map to interpolate.

ay **DEFAULT:** None

for phase.

DEFAULT: 1

for amplitude.

DEFAULT: 1

ence grid for the input maps (in Å).

ay or None **DEFAULT:** None

tegy outside scanned region: None, "edge", or "median".

r None **DEFAULT:** None

ameter dictionary with object guess.

nks

scans with independent beam aberrations. Usefull
e beam is varyying. If applied, the iterative
he beam in each subscan, but apply a different CTF

dictionary.

scan region (in scan points).

ation coefficients per region.

dictionary with aberration array and marker.

unks

scans with independent beam aberrations. Usefull
e beam is varyying. If applied, the iterative
fferenet beam in each of these subscans.

dictionary.

scan region (in scan points).

with probe marker.

ence for local reconstructions.

dictionary.

oordinate.

nate.

on (in scan points).

ion (in scan points).

(take every Nth point).

dictionary with `sequence` key.

`oints`

d specified feature points.

dictionary.

feature points (in scan points).

`int`

feature points.

`int`

onstruction window around each point.

DEFAULT: `20`

to reconstruct.

grid.

ptychography)

dictionary.

of the curl of a rotated DPC vector field. This is the
ion. This particular function was copied from a DPC
l.

degrees.

adians.

itions and angle.

e DPC signal.

errations).

e DPC signal.

dictionary.

of the curl after rotation.

with conjugated probe and CTF.

url-based objective function with respect to rotation

`m_vacscan`

om a vacuum PACBED pattern.

adians.

e DPC signal.

e DPC signal.

with a probe estimate.

rl-based objective function.

the reconstruction parameters.

imizes the curl of the DPC signal.

dictionary.

e DPC signal (2D array).

e DPC signal (2D array).

with aperture.

gle in radians.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ne center of the measured PACBED pattern.

ction. If you setted up the pypty_params properly,
he hpass and lpass values, both are non-negative
dictionary.

nt ("com" or "cross_corr").

DEFAULT: `'com'`

with shifted probe.

ent beam tilt from PACBED.

dictionary with dataset and calibration settings.

g coefficient (default is 0).

DEFAULT: `0`

g coefficient (default is 0).

DEFAULT: `0`

the reconstructed phase (default is False).

DEFAULT: `False`

nter-of-mass x-component.

`or None` **DEFAULT:** `None`

nter-of-mass y-component.

`or None` **DEFAULT:** `None`

e phase reconstruction.

DEFAULT: `False`

D phase image.

ter dictionary with computed COM and rotation angle.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ion. If you setted up the pypty_params properly, you
pass and lpass values, both are non-negative floats.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ON

neter dictionary.

gradient descent iterations (default is 100).

DEFAULT: `100`

ion factor for backtracking (default is 0.5).

DEFAULT: `0.5`

ltering coefficient (default is 0).

DEFAULT: `0`

ltering coefficient (default is 0).

DEFAULT: `0`

ent descent step size (default is 0.1).

DEFAULT: `0.1`

nt of COM map.

DEFAULT: `None`

nt of COM map.

DEFAULT: `None`

ize in Ångströms.

DEFAULT: `None`

print progress information (default is False).

DEFAULT: `False`

save the result to disk (default is False).

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DEFAULT: `False`

ary mask to constrain reconstruction.

ray or None — DEFAULT: `None`

the reconstruction result.

DEFAULT: `True`

use backtracking line search (default is True).

DEFAULT: `True`

D phase image.

boundary artifacts (default is 5).

DEFAULT: `5`

+ ptychography)

er

ransfer function (CTF) using 4D-STEM data.

tion given a Laplacian map.

berrations by aligning individual pixel images using
TF model. It supports iterative fitting with various
w-frequency drift compensation.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Dictionary containing PyPTY experimental and reconstruction settings.
List (for integers) of binning factors for each iteration of the CTF fit. **TYPE:** `list` **DEFAULT:** `[8]`
Whether to save intermediate tcBF images and shift estimates. **TYPE:** `bool` **DEFAULT:** `True`
Whether to include probe rotation angle in the fit. **TYPE:** `bool` **DEFAULT:** `True`
Initial guess for aberrations. If None, n_aberrations_to_fit zeros will be used. **TYPE:** `list` **DEFAULT:** `None`
Number of aberrations to fit if no initial guess is provided. **TYPE:** `int` **DEFAULT:** `12`
Reference used for cross-correlation ("bf" or "zero"). **TYPE:** `string` **DEFAULT:** `'bf'`
Radius (in pixels) of the interpolation box for sub-pixel shift refinement. **TYPE:** `int` **DEFAULT:** `10`
Factor for refining cross-correlation to estimate sub-pixel shifts. **TYPE:** `int` **DEFAULT:** `3`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Type of cross-correlation to use ("phase" or "classical").

TYPE: `str` **DEFAULT:** `'phase'`

Threshold (0–1) to ignore large shift outliers in the fit.

TYPE: `float` **DEFAULT:** `None`

Radius for optional circular blur mask applied to patterns.

TYPE: `int` **DEFAULT:** `None`

Number of scan pixels to pad around the dataset (auto if None).

TYPE: `int` **DEFAULT:** `None`

Aperture mask. If None, attempts to extract from parameters.

TYPE: `ndarray` **DEFAULT:** `None`

Subregion for CTF fitting: [left, top, right, bottom].

TYPE: `list` **DEFAULT:** `None`

Whether to compensate for pattern drift in large FOVs.

TYPE: `bool` **DEFAULT:** `False`

Whether to store low-frequency drift corrections in `params`.

TYPE: `bool` **DEFAULT:** `True`

Factor to upsample the scan via interpolation (experimental).

TYPE: `int` **DEFAULT:** `1`

Binning factor before peak detection in cross-correlation.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `int` **DEFAULT:** `1`

Use analytical peak refinement formula for phase correlation.

TYPE: `bool` **DEFAULT:** `False`

Scaling factor for residuals in `least_squares`.

TYPE: `float` **DEFAULT:** `1`

Scaling for initial step size in `least_squares`.

TYPE: `float` **DEFAULT:** `1`

Loss function for `least_squares` optimization.

TYPE: `string` **DEFAULT:** `'linear'`

Tolerance (`ftol`) for stopping criterion in optimization.

TYPE: `float` **DEFAULT:** `1e-08`

dictionary with fitted aberrations, defocus, and potentially
ions and rotation.

nsmission coherent Bright Field) reconstruction.

F image on an upsampled scan grid from 4D-STEM
ifts to align the bright field pixel images and
e. Prior to calling this function, it is recommended
e to update `pypty_params`.

:SCRIPTION	
ctionary containing experimental parameters and construction settings. This should include keys such as ıta_path', 'scan_size', 'aperture_mask', 'acc_voltage', etc.	'PE: `dict`
sampling factor for the scan grid. Default is 5.	'PE: `int` **DEFAULT:** `5`
ımber of additional scan positions to pad on each side to oid wrap-around artifacts. Default is 10.	'PE: `int` **DEFAULT:** `10`
True, compensates for low-frequency drift of the aperture. quires that run_tcbf_alignment has been executed to provide ft corrections. Default is False.	'PE: `bool` **DEFAULT:** `False`
ecision for floating point computations. Use 64 for higher ecision or 32 for lower memory usage. Default is 64.	'PE: `(64, 32)` **DEFAULT:** `64`
True, shifts are rounded and alignment is performed using ay roll operations. If False, FFT-based subpixel shifting is ed. Default is False.	'PE: `bool` **DEFAULT:** `False`
ckend array module (e.g., numpy or cupy). Default is cupy.	'PE: `module` **DEFAULT:** `numpy`
ag to save the output image. If True, the image is saved to sk. Ignored if 'save_preprocessing_files' is set in `pty_params`. Default is 0 (False).	'PE: `bool or int` **DEFAULT:** `0`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

aximum number of FFTs to perform in parallel for vectorized
ocessing. Default is 100.

'PE: `int` **DEFAULT:** `100`

nning factor for the data in reciprocal space. Default is 1 (no
ning).

'PE: `int` **DEFAULT:** `1`

nage reconstructed on the upsampled grid.

ngströms after upsampling.

`ignment`

compressed 4D-STEM data and masked bright-field

to the shifts between the individual pixel images of
ne as run_tcbf_alignment, but for compressed data.
cross-correaltion.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION

Dictionary containing experimental and reconstruction settings.

TYPE: `dict`

Number of fitting iterations to perform.

TYPE: `int`

Whether to save intermediate tcBF images and shift maps. Default is True.

TYPE: `bool` **DEFAULT:** `True`

Whether to include probe rotation angle in the CTF fit. Default is True.

TYPE: `bool` **DEFAULT:** `True`

Initial guess for the aberration coefficients. If None, it will be inferred or zero-initialized.

TYPE: `list or None` **DEFAULT:** `None`

Number of aberration coefficients to fit if `aberrations` is not provided. Default is 12.

TYPE: `int` **DEFAULT:** `12`

"bf" to use the tcBF image as a reference, "zero" to use the central pixel. Default is "bf".

TYPE: `str` **DEFAULT:** `'bf'`

Size of the cropped region around the cross-correlation peak for sub-pixel refinement. Default is 10.

TYPE: `int` **DEFAULT:** `10`

Upsampling factor for sub-pixel interpolation. Default is 3.

TYPE: `int` **DEFAULT:** `3`

Cross-correlation method: "phase" (recommended) or "classic". Default is "phase".

TYPE: `str` **DEFAULT:** `'phase'`

Threshold to reject large shift outliers during fitting. Value between 0 and 1. Default is None.

TYPE: `float or None` **DEFAULT:** `None`

Width of blur kernel for patterns prior to analysis. Default is None.

TYPE: `int or None` **DEFAULT:** `None`

Number of scan pixels to pad around the scan to prevent wrap-around. Default is auto.

TYPE: `int or None` **DEFAULT:** `None`

Aperture mask defining pixels to analyze. If None, it will be loaded from `params`.

TYPE: `ndarray or None` **DEFAULT:** `None`

Optional subregion [left, top, right, bottom] on which to perform the alignment. Default is None.

TYPE: `list or None` **DEFAULT:** `None`

Whether to compute and correct for slow drifting of the aperture over time. Default is False.

TYPE: `bool` **DEFAULT:** `False`

If True, saves the low-frequency correction back into `params`. Default is True.

TYPE: `bool` **DEFAULT:** `True`

Experimental: interpolate scan grid by this factor (e.g., 2

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

for 2x upsampled grid). Default is 1.

TYPE: `int` **DEFAULT:** `1`

Binning factor applied to cross-correlation maps before refinement. Default is 1.

TYPE: `int` **DEFAULT:** `1`

If True, uses analytical subpixel peak estimation for phase correlation. Default is False.

TYPE: `bool` **DEFAULT:** `False`

Scaling factor for least squares residuals (`f_scale`).

n + ptychography)

of reconstruction parameters including fitted aberrations and

TYPE: `float` **DEFAULT:** `1`

Loss type for least squares optimizer. E.g., "linear", "huber". Default is "linear".

convolution (WDD).

TYPE: `str` **DEFAULT:** `'linear'`

stribution Deconvolution to the provided data,

Tolerance for optimizer convergence (`ftol`). Default is 1e-8.

uction of complex objects and probes. It handles

ia NumPy) computations based on the availability

TYPE: `float` **DEFAULT:** `1e-08`

tionary must be prepared via the `initialize`

ining calibrated parameters, including paths and settings for
.

er for the Wiener filter. Default is 1e-3.

DEFAULT: `0.001`

`_wiener` is ignored. Denominator values below this threshold
the corresponding numerator values are set to 0.

DEFAULT: `None`

output files (False by default). Ignored if
`sing_files` is specified in `pypty_params`.

DEFAULT: `0`

TION

volved complex object.

`larray`

non + ptychography)

essary paths and data arrays prior to calling this function.

ve

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ining calibrated parameters, including paths and settings for
.

tructed from a predefined preset and a given dataset via
st of expected entries can be found in the documentation,

s of PyPty. It performs updates of all active
ect, probe, positions, tilts, etc.) via l-BFGS algorithm.

escribe right now

SCRIPTION

tal loss value.

PE: `float`

m of squared errors.

PE: `float`

t of reg. constraint values for this epoch

PE: `list`

esearch step that was found during this iteration

PE: `float`

thon + ptychography)

mber of linesearch iterations (calls of loss_and_direction)
at was required during this epoch

PE: `int`

e

lue of the direction derivative at this epoch

PE: `float`

lue of the direction derivative at the newly estimated point

ients for ptychographic reconstruction.

PE: `float`

y that performs forward and backward propagation,
rnings during this epoch
asured and simulated patterns, and computes the
PE: `string`
on parameters (object, probe, positions, tilts, etc.).

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

	DESCRIPTION
	Complex 4D object (current estimate) **TYPE:** `ndarray`
	Complex probe (y,x,modes) optionally 4D (y,x,modes, scenatios) **TYPE:** `ndarray`
	Integer beam postions in pixels [[y0,x0],.. [yn, xn]]. Note: units are pixels, not angstrom! **TYPE:** `ndarray`
	Float sub-pixel postions for more precise beam shift. Note: units are pixels, not angstrom! **TYPE:** `ndarray`
	Beam tilts in radians, shape should be (N_measurements, 6), where first two tilts are applied before the sample, second and third are applied inside (tilted propagator) and two last are applied after the sample **TYPE:** `ndarray`
	legacy paramter, actually is not really required. It is a correction that is added to the tilts array. **TYPE:** `ndarray`
	either just one value for a common slice spacing or list of values for each slice. If object has N slices, it should have N-1 entries. **TYPE:** `ndarray`
	array or h5-dataset with diffraction patterns

Should be 3D, [N_measurements, y,x]

TYPE: `ndarray`

string indicating the loss function (error metric)

Electron wavelength in Angstrom

TYPE: `float`

do you refine the beam?

TYPE: `float`

do you refine the object?

TYPE: `float`

do you refine the positions?

do you refine the tilts?

TYPE: `float`

optional, if the data is compressed, you should provide the 3D array with virtual detectors [N_detectors, y,x].

TYPE: `ndarray`

real-space pixel size in x-direction (Angstrom).

TYPE: `float`

real-space pixel size in y-direction (Angstrom).

TYPE: `float`

"far_field" or "near_field". Changes the exit-wave propagation regime.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `string`

Spherical aberration (Angstrom). Only needed for near-field propagation.

TYPE: `float`

Array of exit-wave defocus values (Angstrom). Only needed for near-field propagation.

TYPE: `ndarray`

Flux-preserving correction for near-field propagation.

TYPE: `float`

Cutoff (fraction smaller than 1) beyond which the Fouirer-space is cleaned.

TYPE: `float`

Smooth rolloff for Fourier masking

TYPE: `float`

string indicating the method for split-step integration

TYPE: `string`

sequence of measruement indices used for loss and grad calculation (should be sorted)

TYPE: `ndarray`

boolean flag. should be True for lazy loading.

TYPE: `bool`

multiplicative factor applied to data on the fly.

TYPE: `int`

padding factor applied to data on the fly.

TYPE: `int`

path to h5 dataset containing phase plates for each measurement.

TYPE: `string`

weight applied to the main part of the loss

TYPE: `float`

binning factor applied to data on the fly

TYPE: `int`

shift vector in pixels (y,x) applied to data on the fly

TYPE: `tuple`

virtual "decompression" of the data used to enlarge the probe window

TYPE: `int`

real-valued array descripting the square root of the static offset on the diffraction patterns.

TYPE: `ndarray or float`

do you refine the static background?

TYPE: `float`

flag for tilting

TYPE: `int`

TYPE: `ndarray`

TYPE: `ndarray`

TYPE: `ndarray`

TYPE: `int`

TYPE: `bool`

TYPE: `ndarray`

TYPE: `float`

TYPE: `float`

TYPE: `dtype`

TYPE: `dtype`

TYPE: `module`

TYPE: `bool`

TYPE: `tuple`

TYPE: `float`

TYPE: `float`

TYPE: `float`

`positions`	**TYPE:** `float`
`tilts`	**TYPE:** `float`
	TYPE: `float`
	TYPE: `ndarray`
	TYPE: `float`
	TYPE: `float`
	TYPE: `ndarray`
	TYPE: `float`
	TYPE: `float`
	TYPE: `float`
	TYPE: `float`
	TYPE: `float`
	TYPE: `float`
	TYPE: `float`
	do you want to prevent memory fragmentation? Makes the reconstrcution slightly slower

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `bool`

verbodity level

TYPE: `int`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

:SCRIPTION
tal loss value. 'PE: `float`
m of squared errors. 'PE: `float`
adient of the loss with respect to the object. 'PE: `ndarray`
adient of the loss with respect to the probe. 'PE: `ndarray`
adient of the loss with respect to scan position corrections. 'PE: `ndarray`
adient of the loss with respect to tilts. 'PE: `ndarray`
adient of the loss with respect to static background. 'PE: `ndarray`
adient of the loss with respect to aberration coefficients. 'PE: `ndarray`
adient of the loss with respect to beam current. 'PE: `ndarray`
dividual regularization loss terms added to the total loss. 'PE: `list`

API Reference

pypty.initialize
pypty.dpc
pypty.tcbf
pypty.direct
pypty.iterative
pypty.objective
pypty.multislice_core
pypty.fft
pypty.utils
pypty.se
pypty.vaa

their respective positions in the full probe array.
ort older CuPy version.

t array.

where to add each batch.

to scatter-add.

s to their respective positions in the full aberration
support older CuPy version.

dient array.

where to add each batch.

to scatter-add.

ject regularization to enhance phase and absorption

o regularize.

atio.

DEFAULT: `0.03`

old ratio.

DEFAULT: `0.14`

er for low-phase regions.

DEFAULT: `-0.95`

er for low-absorption regions.

DEFAULT: `-0.95`

hape gaussian kernel sigmas (for phase, absorption).

`None` **DEFAULT:** `None`

ex object.

lization to probe modes.

multiple modes.

be states.

vectors using Gram-Schmidt.

s to orthogonalize.

is.

s by applying a cone filter in 3D FFT space.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

bject.

y

g x (Å).

g y (Å).

een slices (Å).

ss parameter.

reduced missing wedge effects.

traint_on_grid

ation along the fast scan axis.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

s to regularize.

n grid.

weight.

ization term.

ularization.

traint_on_grid

ation along the slow scan axis.

s to regularize.

n grid.

weight.

ct to the input.

nstraint_on_grid

ransformations in local scan patches.

(positions or tilts).

scan.

ight.

.

with respect to the grid.

aint

the object's phase and absorption.

ION

bject array.

rray

absorption norm.

at

phase norm.

at

strict gradient computation.

rray

urn the gradient.

l

fficient option.

l

direction is True, else None.

None

int

predefined window region in real-space.

RIPTION

lex probe array.

: ndarray

w mask.

: ndarray

t of the constraint.

: float

loss.

with respect to the probe.

nt

nt to the probe using an aperture mask. Penalize

e.

ON

obe array.

ray

k or scalar defining aperture radius.

ray or float

on weight.

t

rn the gradient.

frequency components.

straint if requested.

None

TV) regularization to the object.

ON

ject.

ray

on weight.

t

ameter.

t

ameter.

t

ong x (Å).

t

ong y (Å).

t

adient mask.

ray or None

n the gradient.

ty (python + ptychography)

memory-efficient computation.

ce_core

gation using a classic split-step integrator (2nd order
nickness if beam is optimized).

value.

ct to the object.

`constraint`

nt in 3D reciprocal space.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Probe wavefunction with shape [N_batch, y,x, modes] **TYPE:** `ndarray`
Object slices with shape [N_batch, y,x, z, modes]. **TYPE:** `ndarray`
Number of object slices. **TYPE:** `int`
Number of object modes. **TYPE:** `int`
Number of probe modes. **TYPE:** `int`
Slice thicknesses. **TYPE:** `ndarray`
Electron wavelength. **TYPE:** `float`
Spatial frequency grids. **TYPE:** `ndarray`
Spatial frequency grids. **TYPE:** `ndarray`
Spatial frequency grids. **TYPE:** `ndarray`
Mask to exclude undesired frequencies.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `ndarray`

If True, use the same distance for all slices.

TYPE: `bool`

Beam tilts with shape N_batch

TYPE: `ndarray`

Beam tilts with shape N_batch

TYPE: `ndarray`

Damping frequency cutoff.

TYPE: `float`

Rolloff rate for the damping filter.

TYPE: `float`

Full propagator in Fourier space (optional).

TYPE: `ndarray or None`

ace

Half-step propagator (optional).

TYPE: `ndarray or None`

Clean propagation mask.

TYPE: `ndarray`

This array contains interediate exit-waves

TYPE: `ndarray`

This array contains final exit-wave

TYPE: `ndarray`

Numerical types.

TYPE: `dtype`

Numerical types.

TYPE: `dtype`

ON

s.

tack of propagated waves.

`ray`

ave.

`ray`

kernel width in frequency space.

`t`

n the gradient.

ultislice propagation model (object, probe, and

y-efficient FFT loops.

s.

ct to the object if `return_direction` is True.

`None`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Gradient of the loss with respect to the final propagated wave. **TYPE:** `ndarray`
Intermediate wave stack from the forward multislice pass. **TYPE:** `ndarray`
4D sliced object [batch, y, x, z, modes]. **TYPE:** `ndarray`
Gradient accumulator for object slices. **TYPE:** `ndarray`
Accumulator for tilt gradients. **TYPE:** `ndarray`
If True, slice distances are constant. **TYPE:** `bool`
Per-slice thicknesses. **TYPE:** `ndarray`
Frequency mask for FFT operations. **TYPE:** `ndarray`
Probe wavelength (Å). **TYPE:** `float`
Spatial frequency grids. **TYPE:** `ndarray`

Spatial frequency grids.

TYPE: `ndarray`

Spatial frequency grids.

TYPE: `ndarray`

Beam tilt values per batch.

TYPE: `float`

Beam tilt values per batch.

TYPE: `float`

Number of slices.

TYPE: `int`

Number of object modes.

TYPE: `int`

Index in tilt update array.

TYPE: `int`

Full Fourier propagation kernel.

TYPE: `ndarray`

Whether tilt gradient is updated.

TYPE: `int`

Damping cutoff for high-frequency noise.

TYPE: `float`

Width of damping transition.

TYPE: `float`

Mode selector for tilt optimization.

TYPE: `int`

Current batch size.

TYPE: `int`

FFT domain mask.

TYPE: `ndarray`

Whether to compute probe gradient.

TYPE: `int`

Whether to compute object gradient.

TYPE: `int`

(Unused) Flag for positional corrections.

TYPE: `int`

Indices for applying gradients to global object.

TYPE: `ndarray`

Indices for applying gradients to global object.

TYPE: `ndarray`

Floating-point type.

TYPE: `dtype`

Complex type.

TYPE: `dtype`

If True, return probe gradient; else return None.

TYPE: `bool`

ION
or object slices. `rray`
or input probe (if helper_flag_4 is True). `rray or None`
lt gradient. `rray`

gation using an additive split-step method (5th order
hickness).

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Probe wavefunction with shape [N_batch, y,x, modes] **TYPE:** `ndarray`
Object slices with shape [N_batch, y,x, z, modes]. **TYPE:** `ndarray`
Number of object slices. **TYPE:** `int`
Number of object modes. **TYPE:** `int`
Number of probe modes. **TYPE:** `int`
Slice thicknesses. **TYPE:** `ndarray`
Electron wavelength. **TYPE:** `float`
Spatial frequency grids. **TYPE:** `ndarray`
Spatial frequency grids. **TYPE:** `ndarray`
Spatial frequency grids. **TYPE:** `ndarray`
Mask to exclude undesired frequencies.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `ndarray`

If True, use the same distance for all slices.

TYPE: `bool`

Beam tilts with shape N_batch

TYPE: `ndarray`

Beam tilts with shape N_batch

TYPE: `ndarray`

Damping frequency cutoff.

TYPE: `float`

Rolloff rate for the damping filter.

TYPE: `float`

Full propagator in Fourier space (optional).

TYPE: `ndarray or None`

ace

Half-step propagator (optional).

TYPE: `ndarray or None`

Clean propagation mask.

TYPE: `ndarray`

This array contains interediate exit-waves

TYPE: `ndarray`

This array contains final exit-wave

TYPE: `ndarray`

Numerical types.

TYPE: `dtype`

Numerical types.

TYPE: `dtype`

ON

tack of propagated waves.

`ray`

ave.

`ray`

S

obe, and tilts for the "better_multislice" wave

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Gradient of the loss with respect to the final exit wave. **TYPE:** `ndarray`
Stack of intermediate wavefields saved during the forward pass. **TYPE:** `ndarray`
The sliced object with shape [batch, y, x, z, modes]. **TYPE:** `ndarray`
Gradient accumulator for the object slices. **TYPE:** `ndarray`
Gradient accumulator for the tilts. **TYPE:** `ndarray`
Whether all slices have the same thickness. **TYPE:** `bool`
Thickness per slice. **TYPE:** `ndarray`
Frequency mask used in propagation. **TYPE:** `ndarray`
Wavelength of the probe in Ångströms. **TYPE:** `float`
Spatial frequency grids. **TYPE:** `ndarray`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Spatial frequency grids.

TYPE: `ndarray`

Spatial frequency grids.

TYPE: `ndarray`

Beam tilt values per batch.

TYPE: `float`

Beam tilt values per batch.

TYPE: `float`

Number of slices in the object.

TYPE: `int`

Number of probe modes.

TYPE: `int`

Number of object modes.

TYPE: `int`

Index for updating `tilts_grad`.

TYPE: `int`

Whether to update tilts (0 = off).

TYPE: `int`

Full propagator for the current slice.

TYPE: `ndarray`

ace

Half-step propagator.

TYPE: `ndarray`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Cutoff for high frequencies in damping.

TYPE: `float`

Smoothing width for damping filter.

TYPE: `float`

Specifies which tilts to optimize.

TYPE: `int`

Number of scan positions processed in batch.

TYPE: `int`

FFT mask used to remove unstable frequencies.

TYPE: `ndarray`

Indices to scatter object gradients into global coordinates.

TYPE: `ndarray`

Indices to scatter object gradients into global coordinates.

TYPE: `ndarray`

Floating point precision.

TYPE: `dtype`

Complex precision.

TYPE: `dtype`

ION
radient of the object slices. `rray`
f the probe (summed over object modes). `rray`
lt gradients. `rray`

gation using an yoshida integrator (5th order
hickness).

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Probe wavefunction with shape [N_batch, y,x, modes] **TYPE:** `ndarray`
Object slices with shape [N_batch, y,x, z, modes]. **TYPE:** `ndarray`
Number of object slices. **TYPE:** `int`
Number of object modes. **TYPE:** `int`
Number of probe modes. **TYPE:** `int`
Slice thicknesses. **TYPE:** `ndarray`
Electron wavelength. **TYPE:** `float`
Spatial frequency grids. **TYPE:** `ndarray`
Spatial frequency grids. **TYPE:** `ndarray`
Spatial frequency grids. **TYPE:** `ndarray`
Mask to exclude undesired frequencies.

TYPE: `ndarray`

If True, use the same distance for all slices.

TYPE: `bool`

Beam tilts with shape N_batch

TYPE: `ndarray`

Beam tilts with shape N_batch

TYPE: `ndarray`

Damping frequency cutoff.

TYPE: `float`

Rolloff rate for the damping filter.

TYPE: `float`

Full propagator in Fourier space (optional).

TYPE: `ndarray or None`

ace

Half-step propagator (optional).

TYPE: `ndarray or None`

Clean propagation mask.

TYPE: `ndarray`

This array contains interediate exit-waves

TYPE: `ndarray`

This array contains final exit-wave

TYPE: `ndarray`

Numerical types.

TYPE: `dtype`

Numerical types.

TYPE: `dtype`

ON

tack of propagated waves.

`ray`

ave.

`ray`

ds

robe, and tilt parameters using Yoshida multislice

DESCRIPTION
Gradient of the loss with respect to the output wave. **TYPE:** `ndarray`
Stored intermediate wavefields from forward Yoshida multislice pass. **TYPE:** `ndarray`
Object slices with shape [batch, y, x, z, modes]. **TYPE:** `ndarray`
Gradient buffer for object update. **TYPE:** `ndarray`
Gradient buffer for tilt update. **TYPE:** `ndarray`
Whether slice distances are constant. **TYPE:** `bool`
Thickness of each slice. **TYPE:** `ndarray`
FFT mask for excluding unstable frequencies. **TYPE:** `ndarray`
Probe wavelength in Ångströms. **TYPE:** `float`
FFT spatial frequency grids. **TYPE:** `ndarray`

ptychography)

patial frequency grids.

ut array.

default is 1).

DEFAULT: `1`

mbined over object modes).

nsform sample frequencies.

Number of object slices.

TYPE: `int`

their respective positions in the full object array.

Number of probe modes.

ort older CuPy versions.

TYPE: `int`

of object modes.

ut array.

`t`

current tilt in `tilts_grad`.

default is 1).

DEFAULT: `1`

`t`

tilt updates are enabled.

`t`

x-axis

ier domain.

urier Transform sample frequencies.

DESCRIPTION

`None`

TYPE: `ndarray`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

Frequency cutoff for damping high frequencies.

TYPE: `float`

hifted.

RETURNS	DESCRIPTION
`ndarray`	Shifted array.

timize.

TYPE: `int`

Number of scan points in the current batch.

TYPE: `int`

domain mask to stabilize calculations.

nverse shifted.

RETURNS	DESCRIPTION
`ndarray`	Inverse shifted array.

TYPE: `ndarray`

Indices for inserting gradients into global object.

TYPE: `ndarray`

fft2(...)) functions.

Float precision.

TYPE: `dtype`

Complex precision.

TYPE: `dtype`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ransformed.

o compute the FFT (default is (0, 1)).

int **DEFAULT:** `(0, 1)`

writing the input array (default is False).

DEFAULT: `False`

m of the input array, shifted.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ransformed.

o compute the FFT (default is (0, 1)).

int **DEFAULT:** `(0, 1)`

writing the input array (default is False).

DEFAULT: `False`

m of the input array.

ransformed.

o compute the inverse FFT (default is (0, 1)).

`int` **DEFAULT:** `(0, 1)`

writing the input array (default is False).

DEFAULT: `False`

Transform of the input array.

shift(...)) function.

ransformed.

o compute the inverse FFT (default is (0, 1)).

int **DEFAULT:** `(0, 1)`

writing the input array (default is False).

DEFAULT: `False`

Transform of the input array, after shifting.

fftn(...)) function.

ansformed.

o compute the FFT (default is (0, 1, 2)).

int **DEFAULT:** `(0, 1, 2)`

+ ptychography)

ier Transform of the input array, shifted.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

shift(...)) function.

put array.
ansformed.

o be filtered.

ike

y (default is 0.66).

DEFAULT: `0.66`

sk to apply. If None, a mask is generated.

ike or None **DEFAULT:** `None`

ter for smoothing the mask (default is 0).

DEFAULT: `0`

omputations (default is cp.float32).

type **DEFAULT:** `float32`

default is cp).

DEFAULT: `numpy`

DEFAULT: `(0, 1, 2)`

fter applying the Fourier filter.

rse Fourier Transform of the input array.

r shifting.

array. Supports 2D or 3D arrays.

or 3D) to be filtered.	
`ike`	
cy (default is 0.66).	**DEFAULT:** `0.66`
sk to apply. If None, a mask is generated.	
`ike or None`	**DEFAULT:** `None`
ter for smoothing the mask (default is 0).	**DEFAULT:** `0`
omputations (default is cp.float32).	
`type`	**DEFAULT:** `float32`
default is cp).	**DEFAULT:** `numpy`

fter applying the Fourier filter.	

`cies`

s and corresponding masks for multislice

ESCRIPTION
ixel size in the x-direction. **YPE:** `float`
ixel size in the y-direction. **YPE:** `float`
ize of the grid. **YPE:** `int`
amping cutoff factor for multislice simulations. **YPE:** `float`
moothing rolloff parameter. **YPE:** `float`
ata type for computations. **YPE:** `data - type`

q2: 2D array of squared spatial frequencies. - qx: 2D array of in x. - qy: 2D array of spatial frequencies in y. - exclude_mask: ace. - exclude_mask_ishift: Unshifted mask.

applying a phase ramp.

ray.

e

).

float

e shifted probe, the phase mask, the Fourier transform of the
ial frequency grids (maskx, masky).

_from_pos

iven positions and footprint dimensions.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ION

he mask.

he mask.

) positions where the mask should be activated.

t of tuple

vidth.

eight.

or to adjust the footprint (default is 0).

DEFAULT: `0`

nary mask.

grad

the gradient with respect to the phase.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

vative (dL/dz*).

(z = |z| exp(i*phase)).

t (dL/dp).

abs_grad

d negative amplitude gradient from a complex

vative (dL/dz*).

(z = exp(-a + i*phase)).

TION

ntaining: - Phase gradient (dL/dp). - Negative amplitude
dL/da).

API Reference

pypty.initialize
pypty.dpc
pypty.tcbf
pypty.direct
pypty.iterative
pypty.objective
pypty.multislice_core
pypty.fft
pypty.utils
pypty.se
pypty.vaa

ad

from a complex gradient and separate magnitude

ient.

ay.

RETURNS	DESCRIPTION
`array_like`	The magnitude gradient.

roto_phase

ed along phase gradients.

e object.

t array.

late for the object.

lfe condition 1) for line search.

on value.

after the step.

rivative at the current point.

onstant (default is 0.5).

DEFAULT: `0.5`

n is satisfied, False otherwise.

Wolfe condition 2) for line search.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

erivative at the current point.

erivative after the step.

on constant (default is 0.9).

DEFAULT: `0.9`

n is satisfied, False otherwise.

ast two axes.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

upsampled.

upsampled result to conserve total sum (default is True).

DEFAULT: `True`

, numpy or cupy (default is numpy).

DEFAULT: `numpy`

array.

d

he last two axes.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

downsampled.

ctor.

, numpy or cupy.

3D array.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

upsampled.

.

result to conserve total sum (default is True).

DEFAULT: `True`

ault is numpy).

DEFAULT: `numpy`

ay.

downsampled.

ctor.

, numpy or cupy.

array.

e dataset including shifting, binning, padding, and

ION

dataset.

rray

ata is loaded incrementally.

l

construction algorithm.

construction (e.g., near_field, far_field).

cating pixel shift in y and x.

t of int

ctor.

ze.

g factor for the pattern.

cale data intensity.

at

ule, e.g., numpy or cupy.

lule

ly forced padding.

l

preprocessed dataset - data_shift_vector - data_bin - data_pad

window mask.

window.

tapering begins (normalized).

mask falls to zero (normalized).

mask (default is True).

DEFAULT: `True`

the specified shape.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

terms to (n, m, ab) indices based on Krivanek

ion coefficients.

and ab strings ('', 'a', or 'b') for each aberration mode.

string identifiers in Krivanek notation.

es.

nt

indices.

nt

mode types ('', 'a', 'b').

tr

aberration identifiers like 'C30a', 'C11', etc.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ributions for all aberration modes.

in x-direction.

in y-direction.

ion coefficients.

th.

ault is cupy).

DEFAULT: `numpy`

nomials (num_abs, height, width) with phase contributions.

nsfer function (CTF) from aberrations.

n coefficients.

ndarray

y in x-direction.

y in y-direction.

ngth.

on angle in radians (default is 0).

DEFAULT: `0`

he CTF with respect to kx and ky.

n coefficients.

ndarray

y in x-direction.

y in y-direction.

ngth.

on angle (default is 0).

DEFAULT: `0`

ON

of CTF with respect to kx and ky.

`ion_angle`

ase with respect to rotation angle.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

n coefficients.

ndarray

y in x-direction.

y in y-direction.

ngth.

ar offset (default is 0).

DEFAULT: `0`

N

f the CTF gradient in x and y directions with respect to angular

probe in Fourier space.

N

be wavefunction.

ay

ance in meters.

voltage in kiloelectronvolts (keV).

ng x in angstroms.

ng y in angstroms.

a type for computation.

pe for computation.

ackend (NumPy or CuPy).

e

pe.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ace by padding its FFT.

added.

o pad on each side.

spatial domain.

pace to match far-field data dimensions.

PTION

be wavefunction.

darray

f the measured data.

uple

applied to the data.

nt

ling factor used in the reconstruction.

nt

vefunction.

d

d reconstruction.

wavefunction by padding or cropping it to match the

sions after upsampling and padding.

PTION
ıt probe wavefunction. darray
f the measured data. uple
size applied to the data. nt
ling factor applied to the measured data. nt

wavefunction.

e and log loss metrics during training.

s for object, probe, tilts, scan positions, static
nts, and beam current if specified. It also logs loss
a CSV file if logging is enabled.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Directory where files will be saved. **TYPE:** `str`
Current epoch number. **TYPE:** `int`
Whether to save the current probe. **TYPE:** `bool`
Whether to save current scan positions. **TYPE:** `bool`
Whether to save current tilts. **TYPE:** `bool`
Whether to save the current object. **TYPE:** `bool`
Object array to save. **TYPE:** `ndarray or ndarray`
Probe array to save. **TYPE:** `ndarray or ndarray`
Tilt correction values. **TYPE:** `ndarray`
Sub-pixel scan position correction. **TYPE:** `ndarray`
Integer scan positions.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `ndarray`

Original tilt values.

TYPE: `ndarray`

Static background array.

TYPE: `ndarray or ndarray`

Whether to save aberration array.

TYPE: `bool`

Whether to save static background.

TYPE: `bool`

Not used inside the function.

TYPE: `int`

Current loss value.

TYPE: `float`

Current sum of squared errors.

TYPE: `float`

Array of aberration coefficients.

TYPE: `ndarray or ndarray`

Array of beam current values.

TYPE: `ndarray or ndarray`

Whether to save beam current.

TYPE: `bool`

Whether to trigger checkpoint saving.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `bool`

Whether to log loss. If set to 2, log full breakdown of constraints.

TYPE: `bool or int`

List of constraint term contributions to the loss.

TYPE: `list`

Step size applied in the optimizer.

TYPE: `float`

Number of line search iterations.

TYPE: `int`

Initial directional derivative.

TYPE: `float`

New directional derivative after the step.

TYPE: `float`

Step size suggested by BFGS or optimizer.

TYPE: `float`

Start time of the epoch (used for timing).

TYPE: `float`

NumPy or CuPy module used for computation.

TYPE: `module`

Warning string to be logged.

TYPE: `str`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

`nd_from_nothing`

ound if none is provided.

ESCRIPTION
itial static background value or None. **YPE:** `float or ndarray`
robe wavefunction. **YPE:** `ndarray`
laximum spatial frequency used. **YPE:** `float`
adding to be applied. **YPE:** `int`
psampling factor used. **YPE:** `int`
ata type for output. **YPE:** `dtype`
ype of reconstruction ('near_field' or 'far_field'). **YPE:** `str`
ckground.

ing

when no valid probe is provided.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ction either uses an aperture mask, computes a
or adjusts an existing mean pattern to generate a
g, padding, and scaling to produce a probe suitable
type.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Input probe. If set to "aperture", the aperture mask is used. If None, the probe is generated based on the mean pattern. **TYPE:** `ndarray, str, or None`
Padding size applied to the data. **TYPE:** `int`
Mean pattern used to generate the probe if no probe is provided. **TYPE:** `ndarray or None`
Aperture mask used when probe is set to "aperture". **TYPE:** `ndarray`
Flag indicating if tilt mode is active. **TYPE:** `bool`
Tilt values. **TYPE:** `ndarray`
Measured dataset. **TYPE:** `ndarray`
`·y` Factor used to threshold the dataset for aperture estimation. **TYPE:** `bool or float`
Pixel size in the x-direction in angstroms. **TYPE:** `float`
Acceleration voltage in keV.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `float`

Factor to scale the data intensity.

TYPE: `float`

Optional masks to apply to the mean pattern.

TYPE: `ndarray or None`

Vector indicating the shift to be applied to the data.

TYPE: `list or tuple of int`

Binning factor.

TYPE: `int`

Upsampling factor applied to the pattern.

TYPE: `int`

Complex data type for CPU computations.

TYPE: `dtype`

Flag controlling verbosity.

TYPE: `int`

Identifier for the reconstruction algorithm.

TYPE: `str`

Shape of the measured data.

TYPE: `tuple`

Number of object modes.

TYPE: `int`

Marker array for probe scenarios.

TYPE: `ndarray or None`

Type of reconstruction ("near_field" or "far_field").

TYPE: `str`

Array of defocus values.

TYPE: `ndarray`

Spherical aberration coefficient.

TYPE: `float`

e as a complex array.

ased probe modes from a main mode.

N

be mode.

ay

of Hermite polynomials in x.

of Hermite polynomials in y.

a type to use.

ackend.

e

ased probe modes.

mite mode generation, and other modulations to the

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

RIPTION
probe. : `ndarray`
us distance to apply. : `float`
erating voltage in keV. : `float`
size in x (Å). : `float`
size in y (Å). : `float`
f aberration coefficients. : `list or ndarray`
ner to print info. : `bool`
nal beam CTF to apply. : `ndarray or None`
er of Hermite modes in (y, x). : `tuple or None`
us values to generate additional modes. : `list or None`
assignment array for multi-scenario.

: ndarray or None

lex type.

: dtype

type.

: dtype

rical backend.

: module

rray.

ms

or reconstruction.

ions, pads the object if necessary, handles subpixel

electron wavelength based on the accelerating

CRIPTION

tifier for the reconstruction algorithm.

E: `any`

probe array.

E: `ndarray`

object array.

E: `ndarray`

of scan positions.

E: `ndarray`

of tilt angles.

E: `ndarray`

e of the measured data.

E: `tuple`

lerating voltage in keV.

E: `float`

ue, compute subpixel corrections (default is True).

E: `bool` **DEFAULT:** `True`

uence of indices for positions (default is None, which uses full e).

E: `list or callable` **DEFAULT:** `None`

ue, use full field-of-view adjustments (default is False).

E: `bool` **DEFAULT:** `False`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

osity flag (default is 0).

E: `int` **DEFAULT:** `0`

: data type for CPU computations (default is np.float64).

E: `data - type` **DEFAULT:** `float64`

plex data type for CPU computations (default is
omplex128).

E: `data - type` **DEFAULT:** `complex128`

er data type for CPU computations (default is np.int64).

E: `data - type` **DEFAULT:** `int64`

nal mask for probe constraints.

E: `ndarray or None` **DEFAULT:** `None`

nal aperture mask.

E: `ndarray or None` **DEFAULT:** `None`

padding (in pixels) to add to scan positions (default is 0).

E: `int` **DEFAULT:** `0`

- obj : ndarray The padded object array. - positions : ndarray
scan positions. - int A placeholder zero (reserved for future
st The sequence of indices used. - wavelength : float
wavelength in angstroms. - full_pos_correction : ndarray
s for scan positions. - tilts_correction : ndarray Array of zeros
s tilts (tilt corrections). - aperture_mask : ndarray or None The
ask or aperture mask if provided.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

saving training logs.

sults.

e loss values.

index.

RETURNS	DESCRIPTION
None	

ing start of reconstruction.

aset.

e results are saved.

gging is enabled.

RETURNS	DESCRIPTION
None	

`t_obj_probe`

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

n data as checkpoints.

state of the object, probe, tilt corrections, scan
d aberrations to disk. It is intended to allow
ne last checkpoint.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Directory where checkpoint files will be saved. **TYPE:** `str`
The current object array. **TYPE:** `ndarray or GPU array`
The current probe array. **TYPE:** `ndarray or GPU array`
Correction values for tilt angles. **TYPE:** `ndarray`
Sub-pixel correction values for scan positions. **TYPE:** `ndarray`
Scan positions array. **TYPE:** `ndarray`
Tilt angles array. **TYPE:** `ndarray`
Static background array. **TYPE:** `ndarray`
Flag indicating whether to save the probe. **TYPE:** `bool`
Flag indicating whether to save the object. **TYPE:** `bool`
Flag indicating whether to save the scan positions.

TYPE: `bool`

Flag indicating whether to save the tilt angles.

TYPE: `bool`

Flag indicating whether to save the static background.

TYPE: `bool`

Flag indicating whether to save the aberrations array.

TYPE: `bool`

The current aberrations array.

TYPE: `ndarray or GPU array`

The current beam current array.

TYPE: `ndarray or GPU array or None`

Flag indicating whether to save the beam current.

TYPE: `bool`

Numerical backend (e.g., numpy or cupy).

TYPE: `module`

rogress including loss, optimization state, and

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

DESCRIPTION
Start time of the epoch (Unix timestamp). **TYPE:** `float`
Name of the loss or optimization algorithm used. **TYPE:** `str`
Current training epoch. **TYPE:** `int`
Loss value at current epoch. **TYPE:** `float`
Sum of squared errors. **TYPE:** `float`
Whether the object is being updated. **TYPE:** `bool`
Whether the probe is being updated. **TYPE:** `bool`
Whether the scan grid is being updated. **TYPE:** `bool`
Whether tilt corrections are being updated. **TYPE:** `bool`
Whether static background is being updated. **TYPE:** `bool`
Whether aberration coefficients are being updated.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

TYPE: `bool`

Whether beam current is being updated.

TYPE: `bool`

Optimizer memory length (0=GD, 1=CG, >1=BFGS).

TYPE: `int`

Verbosity flag: 0 = silent, 1 = single-line print, 2 = verbose.

TYPE: `int`

ariables to GPU arrays if using CuPy.

PTION

rray.

darray

ray.

darray

scan positions.

darray

l scan grid correction.

darray

es.

darray

ections.

darray

segmentation or region masks.

darray or None

defocus values per position.

darray

acing in multislice simulations.

darray

perture mask.

darray or None

ed dataset.

darray

dataset is streamed from disk.

bool

ackground array.

darray or None

aberration coefficients.

darray or None

urrent scaling factor.

darray or None

ecision dtype for casting.

type

precision dtype for casting.

type

dtype for casting.

type

al backend (numpy or cupy).

odule

in GPU format (if using CuPy), with proper types.

_current

ad if it's too short.

PTION

beam current values.

darray or None

f measured dataset.

uple

e for the array.

ptychography)

or CuPy.

odule

d beam current.

from a 4D-STEM dataset.

ions at the current epoch.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ining parameters including data path, scan size, plotting
ut folder.

the resulting HAADF image as a .npy file. Default is True.

DEFAULT: `True`

type.

`list` Evaluated values.

al HAADF image.

ent epoch.

based on the mean diffraction pattern.

er, callable) or fixed values.

ing parameters including data path, data padding, plotting
threshold.

s dictionary containing the generated aperture mask.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

...as a string and save it to a new file.

...PTION
...ed string source of the lambda.

...original dataset.

...ve the binned dataset.

... to downsample the dataset.

...ambda serialization.

...riginal parameter dictionary.

...ft

...YPE: `dict`

...on patterns via phase correlation.

...Whether to exclude 'dataset' key (default is True).

...ULT: `True`

...e mask used for phase correlation.

...tterns to be compensated for drift.

... function.

...diffraction patterns.

tector

om annular masks.
to evaluate.

ining parameters including data path, scan size, plotting
ut folder.

he annular mask. Default is 0.

DEFAULT: `0`

he annular mask. Default is 1.

DEFAULT: `1`

the resulting virtual detector signal as a .npy file. Default is

DEFAULT: `False`

annular mask. Default is 0.

· ptychography)

DEFAULT: `0`

annular mask. Default is 0.

DEFAULT: `0`

al detector signal.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

ontaining the modes to be plotted. s.

e

with callables

d on step size and angle.

otionally removing the dataset.

ESCRIPTION

utput path for the parameter file.

YPE: `str`

arameter dictionary to save.

YPE: `dict`

True, remove the dataset entry.

YPE: `bool`

ION

eal-space wave

rray

DEFAULT: `0`

`[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]`

DEFAULT: `True`

DEFAULT: `1e-20`

DEFAULT: `1e-20`

DEFAULT: `'linear'`

DEFAULT: `inf`

DEFAULT: `None`

ions (in Angstrom)

econd image.

d on step size and angle.

tistics.

he model.

ns.

.

based on GPU memory usage.

.

coordinates.

ns

imization.

e optimization.

. instead of fully loaded.

. .

inates. t.

inates. y to use.

vided memory strategy.

l

d differences.

nt

id object array.

uple
le for a scan grid.

f the probe array.

uple

e string ('single' or 'double').

tr

tion method name.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

sitions.

ne_by_one flag, and memory strategy.

an grid.

s from a NeXus (.nxs) HDF5 file.

ngle in degrees.

DESCRIPTION
Dictionary of extracted parameters.

`dict`

ion matrix from positions.

meter file.

sitions.

ers file.

an grid.

RETURNS	DESCRIPTION
`None`	

roms.

RETURNS	DESCRIPTION
`ndarray`	The deformation matrix.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

the scale bar will be added.

of the bottom left corner of the scale bar.

of the bottom left corner of the scale bar.

cale bar in pixels.

scale bar in pixels.

or the text label.

or the text label.

ne same units as the width and height.

rement for the scale bar.

CSV file.

terations to skip (default is 0).

DEFAULT: `0`

-axis showing time in seconds will be added on top of the

DEFAULT: `True`

es.

2D array.

.

e radial average.

ius for averaging.

us for averaging.

roms.

verage will be plotted (default is True).

DEFAULT: `True`

g the plot if `plot` is True.

API Reference

pypty.initialize

pypty.dpc

pypty.tcbf

pypty.direct

pypty.iterative

pypty.objective

pypty.multislice_core

pypty.fft

pypty.utils

pypty.se

pypty.vaa

rray of shape (N_y, N_x, N_obs).

rincipal components to retain.

rray of shape (N_y, N_x, n_components).

format.

complex numbers.

ither 'dark' or 'light' (default is 'dark').

DEFAULT: `'dark'`

e value for normalization (default is None).

DEFAULT: `None`

ation of the input array.

mat.

complex modes.

odes to plot.

vs for the subplot layout.

ng the plotted complex modes.