

¹ tomopari: A plugin for accelerated tomographic reconstruction

³ **Marcos Obando^{1,2,3*}, Minh Nhat Trinh^{4*}, David Paleček⁴, Germán Mato^{1,5*},**
⁴ **and Teresa M Correia^{4,6*¶}**

⁵ **1** Instituto Balseiro, Bariloche, Argentina **2** Centrum Wiskunde & Informatica, Amsterdam, the
⁶ Netherlands **3** University of Eastern Finland, Kuopio, Finland **4** Centro de Ciências do Mar do Algarve
⁷ (CCMAR/CIMAR LA), University of Algarve, Faro, Portugal **5** Medical Physics Department, Centro
⁸ Atómico Bariloche, Bariloche, Argentina **6** School of Biomedical Engineering and Imaging Sciences,
⁹ King's College London, London, United Kingdom ¶ Corresponding author * These authors contributed
¹⁰ equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: ↗

Submitted: 19 December 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a

Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))

¹¹ Summary

¹² Recent advances in various tomographic methodologies have contributed to a broad range of
¹³ applications and research areas, from X-ray imaging dealing with medical ([Ginat & Gupta, 2014](#)) and industrial ([De Chiffre et al., 2014](#)) applications to optical sectioning, which provides
¹⁴ a mesoscopic framework to visualise translucent samples ([Sharpe, 2004](#)), to name a few.
¹⁵ Tomographic imaging involves acquiring raw data (2D projections) from various angles around
¹⁶ the object of interest. Once data are acquired, there are several challenges: first, if necessary,
¹⁷ artefacts are corrected in a preprocessing stage; then, raw projections are reconstructed using an
¹⁸ appropriate reconstruction algorithm; and, finally, the results are visualised as 2D cross-sectional
¹⁹ images (slices) and 3D renderings (combined slices), providing a complete and interactive view
²⁰ of the internal structure of the sample.

²¹ Here, we introduce *tomopari*, a plugin for napari ([Chiu et al., 2022](#)) (open-source, interactive,
²² N-dimensional image viewer) that contains four main tomographic reconstruction methods,
²³ often requiring separate software: filtered backprojection (FBP) ([Kak & Slaney, 2001](#)), Two-
²⁴ step Iterative Shrinkage/Thresholding (TwIST) ([Bioucas-Dias & Figueiredo, 2007](#)) ([Correia et](#)
²⁵ [al., 2015](#)), U-Net ([Ronneberger et al., 2015](#)) ([Davis et al., 2019](#)), and Tomographic Model-
²⁶ based Deep learning reconstruction (ToMoDL) ([Obando et al., 2023](#)), which is our recently
²⁷ developed model-based deep learning reconstruction method for accelerated Optical Projection
²⁸ Tomography (OPT).

²⁹ The input to the *tomopari* plugin is an ordered stack of projection images acquired during a
³⁰ tomography scan, typically provided in standard image formats such as TIFF, JPEG, or PNG.
³¹ The user only needs to specify the rotation axis of the system (vertical or horizontal) and
³² select the desired reconstruction method. Several additional options are available to enhance
³³ reconstruction quality and performance, including **projection image resizing or compression**,
³⁴ **manual or automatic center-of-rotation alignment** ([Walls et al., 2005](#)), **clipping to circular**
³⁵ **regions**, **filter selection for FBP methods**, **partial or full volume reconstruction**, **intensity**
³⁶ **inversion**, and **CPU/GPU computation selection**. Together, these features make *tomopari*
³⁷ a flexible and efficient framework for performing tomographic reconstructions directly within
³⁸ napari. Thus, enabling data pre-processing, 3D image reconstruction, visualisation and analysis,
³⁹ all within a single environment.

⁴⁰ *tomopari* is integrally based on well-established open-source software libraries such as NumPy
⁴¹ ([Harris et al., 2020](#)), Scipy ([Virtanen et al., 2020](#)) and scikit-image ([Walt et al., 2014](#)). The
⁴² U-Net and ToMoDL are neural network-based techniques, which have been trained using the

44 PyTorch framework ([Paszke et al., 2019](#)) and demonstrate excellent performance for accelerated
45 sparse reconstruction. The computational burden imposed by the iterative application of the
46 Radon transform (forward model) is mitigated through the use of an adapted version of
47 QBI-radon — a fast, differentiable routine for computed tomography reconstruction ([Trinh
& Correia, 2025](#)). This implementation, developed as a PyTorch 2.0 extension, enables
48 efficient execution on both CPU and GPU across all major operating systems. Combining QBI-
49 radon with *tomopari* enables high-performance reconstructions while maintaining compatibility
50 with modern deep learning workflows, makes it suitable for both analytical and model-based
51 reconstruction methods.

53 Statement of need

54 Tomographic image reconstruction is crucial across many domains where internal structures
55 must be visualised non-invasively. However, it often relies on slow iterative optimisation
56 reconstruction algorithms, particularly when reconstructing 3D images from undersampled
57 acquisitions. For this purpose, several Python libraries have been introduced to alleviate
58 this burden, such as scikit-image ([Walt et al., 2014](#)), ASTRA ([Van Aarle et al., 2016](#)) and
59 TorchRadon ([Ronchetti, 2020](#)). Nevertheless, these tools require expertise in image processing
60 techniques and computer programming and therefore do not offer, by themselves, an accessible
61 and efficient framework for data analysis and visualisation, which can be easily used by
62 researchers performing experiments using tomographic systems.

63 Napari ([Chiu et al., 2022](#)) provides a fast, flexible and user-friendly viewer for 2D and 3D
64 large-scale images, and has rapidly emerged as a hub for high-performance applications
65 including microscopy, medical imaging, astronomy, etc. Therefore, there is a context with an
66 extensive offer of tomographic reconstruction algorithms and a lack of software integration for
67 image analysts, enabling them to seamlessly access other complex tasks such as segmentation
68 ([Ronneberger et al., 2015](#)) and tracking ([Wu et al., 2016](#)).

69 The user-friendly software presented here aims to bridge the gap between a wide variety of
70 reconstruction techniques and napari by introducing a ready-to-use widget that offers state-
71 of-the-art methods for tomographic reconstruction and provides a flexible framework that
72 supports the inclusion of new methods in the future.

73 Methods and Workflow

74 The reconstruction methods implemented in the *tomopari* package are:

- 75 ▪ **FBP** (Filtered backprojection) is a widely used method for tomographic reconstruction.
76 Typically, it involves filtering the data in the frequency domain using a ramp filter, which
77 amplifies high-frequency components, and then backprojecting the filtered projections
78 from multiple angles into the image domain. The filter used in FBP is typically a
79 (modified) ramp filter, which enhances high-frequency components to correct for the
80 blurring caused by backprojection. FBP is computationally efficient and works well for
81 simple geometries, such as parallel-beam tomography.
- 82 ▪ **TwIST** is an iterative method for compressed sensing image reconstruction adapted for
83 tomographic reconstruction ([Correia et al., 2015](#)), which involves solving a non-convex
84 optimisation problem using the shrinkage/thresholding technique for each 2D slice. In
85 this implementation, we chose to minimise the total variation norm as our regularising
86 function. TwIST can handle a wide range of geometries and produce high-quality
87 reconstructions. However, it is computationally expensive and requires careful tuning of
88 algorithm parameters.
- 89 ▪ **U-Net** is a deep learning architecture for tomographic reconstruction that uses a U-shaped
90 network with skip connections ([Ronneberger et al., 2015](#)). The proposed network in
91 ([Davis et al., 2019](#)) processes undersampled FBP reconstructions and outputs streak-free

92 2D images. Skip connections help preserve fine details in the reconstruction, so that the
 93 network can handle complex geometries and noisy data. While reconstruction is fast,
 94 making it suitable for real-time imaging, training a U-Net requires large amounts of data.

95 • **ToMoDL** is a model-based deep learning framework that combines iterations over a data
 96 consistency step and an image domain artefact removal step achieved by a convolutional
 97 neural network. The data consistency step is implemented using the conjugate gradient
 98 algorithm and the artefact removal via a deep neural network with shared weights across
 99 iterations. As the forward model is explicitly accounted for, the number of network
 100 parameters to be learned is significantly reduced compared to direct inversion approaches,
 101 providing better performance in settings where the amount of training data is limited
 102 ([Obando et al., 2023](#)).

103 In [Figure 1](#), a complete pipeline describing the usage of *tomopari* is presented. The input is a
 104 single channel raw data acquired in a parallel beam tomography, loaded as an ordered stack of
 105 files. Two user modes are provided: a basic mode for users without deep learning expertise,
 106 and an advanced mode for fine control over smoothing, alignment, flat-field correction, and
 107 compression trade-offs. Processing steps labeled 1-10 in [Figure 1](#) are:

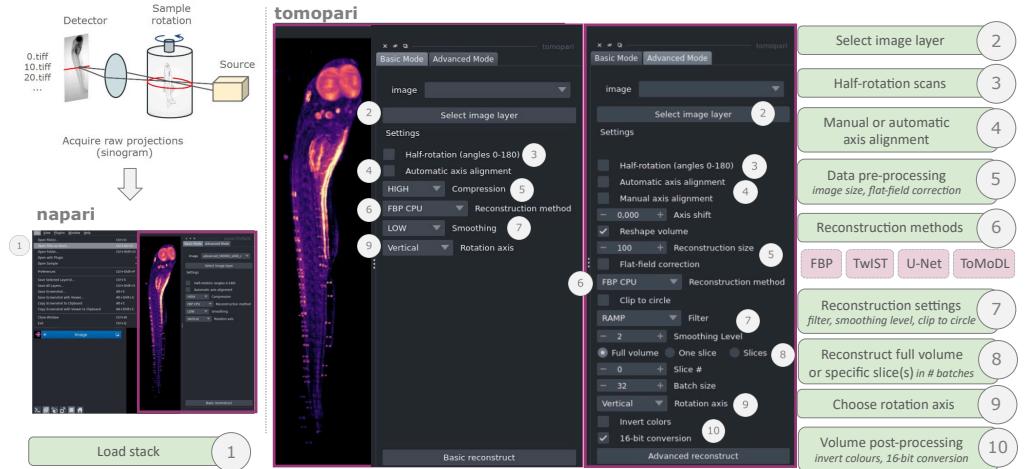


Figure 1: tomopari basic and advanced mode pipelines. Step-by-step from a stack of raw projection acquisition to reconstruction of a single specific slice or full volume.

- 108 1. **Load stack** – The workflow begins by importing the ordered stack of raw projection
 109 images (sinograms) into napari using its file manager. This generates a new 3D image
 110 layer representing the raw data to be reconstructed.
- 111 2. **Select image layer** – Within the *tomopari* plugin interface, the user selects the desired
 112 input layer by clicking on *Select image layer*. This step defines which dataset will be
 113 used for reconstruction.
- 114 3. **Half-rotation scans** – If the acquisition corresponds to a 180° half-rotation instead of a
 115 full 360° dataset, this option needs to be enabled to correctly interpret the projection
 116 geometry during reconstruction.
- 117 4. **Manual or automatic axis alignment** – The rotation axis alignment can be adjusted
 118 automatically or manually.
 - 119 • *Automatic alignment* applies an efficient implementation of the variance maximisation
 120 method ([Walls et al., 2005](#)) to estimate the correct center of rotation.
 - 121 • *Manual alignment* allows the user to specify the pixel offset corresponding to the
 122 rotation axis shift. For fine-tuning, a single-slice reconstruction is recommended for
 123 iterative manual adjustment.

- 124 5. **Data pre-processing** – Optional pre-processing steps such as flat-field correction and image
125 resizing can be applied to normalise projection intensities and adapt image dimensions
126 before reconstruction. As an alternative, in the basic mode, users can select:
127 ▪ *Compression/projection image resizing* - Resize the volume dimension corresponding to
128 the detector's width/height to reduce memory usage or accelerate computation. The
129 plugin provides four compression levels: **HIGH**, **MEDIUM**, **LOW**, or **NO**, producing
130 resolutions of 100, 256, 512, or the full uncompressed size, respectively
- 131 6. **Reconstruction methods** – Users can select between different reconstruction algorithms
132 according to their application:
133 ▪ *FBP* (Filtered Back Projection) — classical analytical approach;
134 ▪ *TwIST* — iterative algorithm for sparse or undersampled data;
135 ▪ *U-Net* — deep learning-based reconstruction using a convolutional network;
136 ▪ *ToMoDL* — hybrid deep learning approach combining model-based priors and
137 learned representations.
- 138 **CPU/GPU selection** Users can choose whether to perform reconstruction on the CPU or
139 accelerate computations using the GPU, depending on available hardware and the selected
140 algorithm. When running on the GPU, *tomopari* supports **batch reconstruction**, allowing
141 multiple slices to be reconstructed **in parallel** to significantly improve processing speed. Batch
142 size can be adjusted depending on available GPU memory in the advanced mode.
- 143 7. **Reconstruction settings** – Parameters controlling reconstruction quality can be adjusted,
144 including the choice of filter (for FBP), smoothing level, and whether to clip the
145 reconstruction to a circular field of view.
146 ▪ *Clip to circle* Restricts reconstruction to a circular field of view (FOV), i.e. the recon-
147 structed image is masked so that everything outside the FOV is set to zero, removing
148 background noise and improving visual quality.
149 ▪ *Filter selection (for FBP methods)* Users can choose the desired filtering kernel (e.g.,
150 *Ram-Lak*, *Shepp-Logan*, etc.) for the filtered backprojection algorithm, balancing noise
151 suppression and edge preservation.
152 ▪ *Smoothing level* corresponds to the number of ToMoDL iterations, which controls the
153 sharpness/smoothness of the reconstructed images. As an alternative, in the basic mode,
154 users can select the smoothing strength as **HIGH**, **MEDIUM**, **LOW**, that correspond to
155 6, 4 and 2 iterations of tomndl respectively.
- 156 8. **Reconstruct the full volume or selected slices** – Users can choose to reconstruct the entire
157 volume, a single slice, or a specific range of slices. For large datasets, reconstructing
158 only a few slices is useful for quick testing. Reconstruction can also be performed in
159 multiple batches to optimize memory usage and improve computational efficiency.
- 160 9. **Choose rotation axis** – Depending on the experimental setup, the user defines whether
161 the rotation axis is vertical or horizontal to ensure that projections are correctly aligned
162 during reconstruction.
- 163 10. **Volume post-processing** – Finally, reconstructed volumes can optionally be post-processed,
164 for example by applying color inversion or converting the data to 16-bit depth for faster
165 rendering. The reconstructed volume is always produced in 32-bit float precision. However,
166 the plugin provides an option to convert the final volume to 16-bit, which can significantly
167 improve 3D rendering performance in napari without affecting the reconstruction step
168 itself.
169 ▪ *Full or partial volume reconstruction* Enables fast testing or memory-efficient reconstruc-
170 tion by limiting computation to a subset of slices along the detector axis.
171 ▪ *Intensity inversion* Inverts grayscale values in the reconstructed image volume, which
172 can be useful when projection data were acquired with inverted intensity mapping.

173 Once these steps are completed, the ‘Reconstruction’ button executes the desired specifications
174 for image recovery from projections. In napari, outputs are written as image layers, which can
175 be analysed by other plugins and saved in different formats. One special feature that napari
176 offers on top of 3D images is volume rendering, useful once a full volume is computed with
177 the presented plugin. Normalisation of intensity and contrast can also be applied to specific
178 layers using napari’s built-in tools in the top-left bar.

179 Use cases

180 We present three parallel beam tomography use cases for the *tomopari* plugin:

- 181 1. **Optical projection tomography (OPT)** Projection data of wild-type zebrafish (*Danio*
182 *rerio*) at 5 days post fertilisation were obtained using a 4× magnification objective.
183 Using a rotatory cylinder, transmitted projection images were acquired with an angle
184 step of 1 degree. The acquired projections have 2506×768 pixels with a resolution of
185 1.3 μm per pixel (Bassi et al., 2015). These projections were downsampled to 627×192
186 pixels in order to reduce the computational complexity. Note that deep learning-based
187 reconstruction methods were only trained with this OPT dataset.
- 188 2. **High-resolution X-ray parallel tomography (X-ray CT)**. Projection data from a foraminifera
189 were obtained using 20 KeV X rays and a high-resolution detector with 1024×1280 pixels
190 (5 μm per pixel). A rotatory support was used to acquire 360 projections with 1-degree
191 interval. The projections were downsampled to 256×320 to reduce computational
192 complexity. The raw data was pre-processed using phase contrast techniques to improve
193 contrast (Paganin et al., 2002).
- 194 3. **High-Throughput Tomography (HiTT)**. Synchrotron X-ray projection data from an ant,
195 fixed in a mixture of PFA (paraformaldehyde) and GA (glutaraldehyde), dehydrated
196 and mounted in ethanol, were obtained using a phase-contrast imaging platform for
197 life-science samples on the EMBL beamline P14 (Albers et al., 2024). The HiTT
198 dataset contains 1800 projections over 180 degrees, acquired at 0.1-degree intervals,
199 each composed of 3 tiles with a size of 2048×2048 pixels each (0.65 μm per pixel).
200 The projections were downsampled by a factor of 2.

201 In Figure 2 we show representative examples of the 2D reconstructions obtained with FBP and
202 ToMoDL and 3D volumes obtained using the plugin with the ToMoDL option. The volumes
203 were fully rendered using the built-in napari capabilities, allowing for full integration of the
204 data analysis workflow in napari.

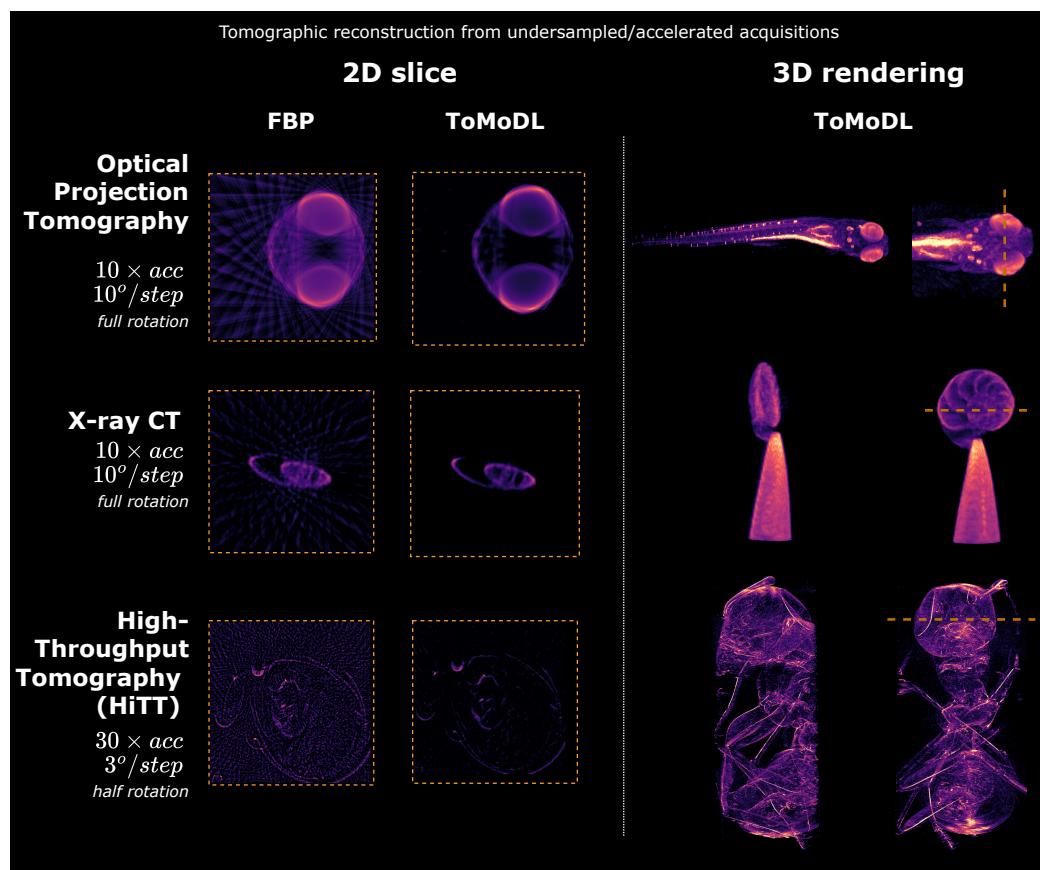


Figure 2: Reconstruction use cases. Left panels: 2D slices reconstructed from undersampled data using FBP and ToMoDL methods (OPT, X-ray CT and synchrotron X-ray HiTT). For each case, the acceleration factor, degrees per step and rotation range are indicated. Right panels: 3D renderings of ToMoDL reconstructions.

205 Acknowledgements

206 This study received Portuguese national funds from FCT - Foundation for Science and
 207 Technology through contracts UID/04326/2025, UID/PRR/04326/2025 and LA/P/0101/2020
 208 (DOI:10.54499/LA/P/0101/2020), 'la Caixa' Foundation and FCT, I P under the Project
 209 code LCF/PR/HR22/00533, European Union's Horizon 2020 research and innovation
 210 program under the Marie Skłodowska-Curie OPTIMAR grant with agreement no 867450
 211 (DOI:10.3030/867450), European Union's Horizon Europe Programme IMAGINE under grant
 212 agreement no. 101094250 (DOI:10.3030/101094250) and NVIDIA GPU hardware grant.
 213 M.O was supported by the European Union (GA 101072354) and UKRI (grant number
 214 EP/X030733/1). Views and opinions expressed are however those of the author(s) only and
 215 do not necessarily reflect those of the European Union or the European Research Executive
 216 Agency. Neither the European Union nor the granting authority can be held responsible for
 217 them. M.O. would like to thank Dr Ezgi Demircan-Tureyen and Dr Vladyslav Andriashen
 218 (Centrum Wiskunde & Informatica) for useful discussions. The authors would like to express
 219 sincere gratitude to: Dr. Ksenia Denisova, Dr. Elizabeth Duke and Dr. Yannick Schwab from
 220 EMBL for providing HiTT data and support in data preparation; Dr. Andrea Bassi from
 221 Politecnico di Milano for providing access to the OPT data; and Dr. Jose Lipovetzky and
 222 Damian Leonel Corsi (Centro Atomico Bariloche) for providing access to the high-resolution
 223 X-ray data.

224 References

- 225 Albers, J., Nikolova, M., Svetlove, A., Darif, N., Lawson, M. J., Schneider, T. R., Schwab,
226 Y., Bourenkov, G., & Duke, E. (2024). High throughput tomography (HiTT) on EMBL
227 beamline P14 on PETRA III. *Journal of Synchrotron Radiation*, 31(1).
- 228 Bassi, A., Schmid, B., & Huisken, J. (2015). Optical tomography complements light sheet
229 microscopy for *in toto* imaging of zebrafish development. *Development*, 142(5), 1016–1020.
- 230 Bioucas-Dias, J. M., & Figueiredo, M. A. (2007). A new TwIST: Two-step iterative shrink-
231 age/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*,
232 16(12), 2992–3004.
- 233 Chiu, C.-L., Clack, N., & others. (2022). Napari: A python multi-dimensional image viewer
234 platform for the research community. *Microscopy and Microanalysis*, 28(S1), 1576–1577.
- 235 Correia, T., Lockwood, N., Kumar, S., Yin, J., Ramel, M.-C., Andrews, N., Katan, M.,
236 Bugeon, L., Dallman, M. J., McGinty, J., & others. (2015). Accelerated optical projection
237 tomography applied to *in vivo* imaging of zebrafish. *PLOS One*, 10(8), e0136213.
- 238 Davis, S. P., Kumar, S., Alexandrov, Y., Bhargava, A., Silva Xavier, G. da, Rutter, G. A.,
239 Frankel, P., Sahai, E., Flaxman, S., French, P. M., & others. (2019). Convolutional neural
240 networks for reconstruction of undersampled optical projection tomography data applied to
241 *in vivo* imaging of zebrafish. *Journal of Biophotonics*, 12(12), e201900128.
- 242 De Chiffre, L., Carmignato, S., Kruth, J.-P., Schmitt, R., & Weckenmann, A. (2014). Industrial
243 applications of computed tomography. *CIRP Annals*, 63(2), 655–677.
- 244 Ginat, D. T., & Gupta, R. (2014). Advances in computed tomography imaging technology.
245 *Annual Review of Biomedical Engineering*, 16, 431–453.
- 246 Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau,
247 D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., & others. (2020). Array programming
248 with NumPy. *Nature*, 585(7825), 357–362.
- 249 Kak, A. C., & Slaney, M. (2001). *Principles of computerized tomographic imaging*. SIAM.
- 250 Obando, M., Bassi, A., Ducros, N., Mato, G., & Correia, T. M. (2023). Model-based deep
251 learning framework for accelerated optical projection tomography. *Scientific Reports*, 13(1),
252 21735.
- 253 Paganin, D., Mayo, S. C., Gureyev, T. E., Miller, P. R., & Wilkins, S. W. (2002). Simultaneous
254 phase and amplitude extraction from a single defocused image of a homogeneous object.
255 *Journal of Microscopy*, 206(1), 33–40.
- 256 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z.,
257 Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M.,
258 Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An
259 imperative style, high-performance deep learning library. In *Advances in neural information
260 processing systems 32* (pp. 8024–8035). Curran Associates, Inc. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- 261 Ronchetti, M. (2020). Torchradon: Fast differentiable routines for computed tomography.
262 *arXiv Preprint arXiv:2009.14788*.
- 263 Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for bio-
264 medical image segmentation. *Medical Image Computing and Computer-Assisted Interven-
265 tion—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015,
266 Proceedings, Part III* 18, 234–241.
- 267 Sharpe, J. (2004). Optical projection tomography. *Annu. Rev. Biomed. Eng.*, 6, 209–228.
- 268 Trinh, M.-N., & Correia, T. (2025). *QBIImaging/QBI_radon* (Version v1.7). Zenodo.

- 270 <https://doi.org/10.5281/zenodo.16416059>
- 271 Van Aarle, W., Palenstijn, W. J., Cant, J., Janssens, E., Bleichrodt, F., Dabravolski, A., De
272 Beenhouwer, J., Batenburg, K. J., & Sijbers, J. (2016). Fast and flexible x-ray tomography
273 using the ASTRA toolbox. *Optics Express*, 24(22), 25129–25147.
- 274 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
275 Burovski, E., Peterson, P., Weckesser, W., Bright, J., & others. (2020). SciPy 1.0:
276 Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3),
277 261–272.
- 278 Walls, J. R., Sled, J. G., Sharpe, J., & Henkelman, R. M. (2005). Correction of artefacts in
279 optical projection tomography. *Physics in Medicine & Biology*, 50(19), 4645.
- 280 Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager,
281 N., Gouillart, E., Yu, T., & contributors, the scikit-image. (2014). Scikit-image: Image
282 processing in python. *PeerJ*, 2, e453. <https://doi.org/10.7717/peerj.453>
- 283 Wu, A., Xu, Z., Gao, M., Buty, M., & Mollura, D. J. (2016). Deep vessel tracking: A generalized
284 probabilistic approach via deep learning. *2016 IEEE 13th International Symposium on
285 Biomedical Imaging (ISBI)*, 1363–1367.

DRAFT