Open Source Software for MRI Reconstruction

2026-05-09 – ISMRM26 | Cape Town

Pierre-Antoine Comby

Neurospin, CEA, Paris-Saclay University

Declaration of Financial Interests or Relationships

Speaker Name: Pierre-Antoine Comby

I have no financial interests or relationships to disclose with regard to the subject matter of this presentation.

Outline

- MRI Reconstruction

1. Landscape of MRI Reconstruction
   Overview of current usage in languages and toolboxes
2. MRI-NUFFT as a case study
3. Beyond toolboxes and software
   Dataset, Benchmarking, Community Initiative

- Good practice to take home

MRI Reconstruction

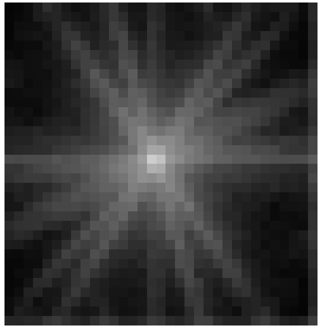
- Solving an (or many) *ill-posed* **inverse problem**

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon}$$

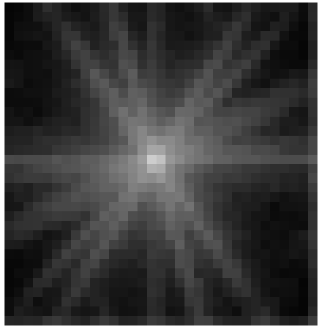
MRI Reconstruction

- Solving an (or many) *ill-posed* **inverse problem**

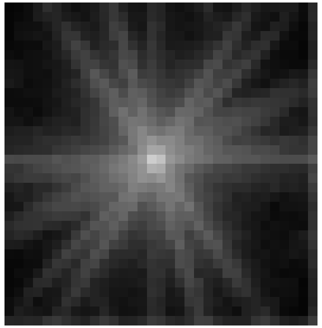
$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon}$$



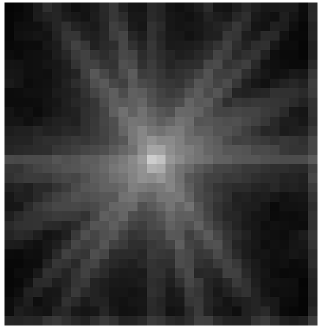
Anatomical MRI



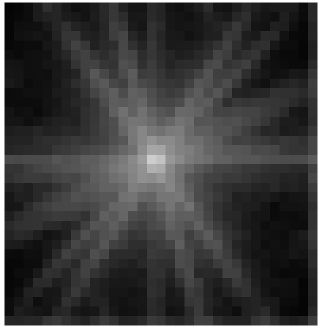
Cardiac /CINE



Quantitative MRI



Diffusion MRI

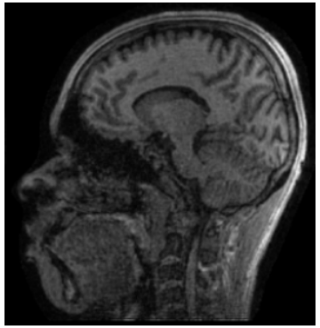


Functional MRI

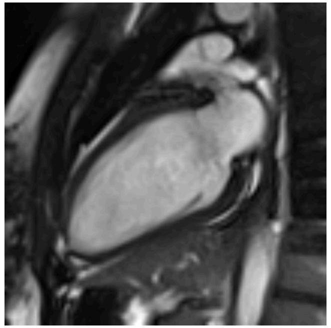
MRI Reconstruction

- Solving an (or many) *ill-posed* **inverse problem**

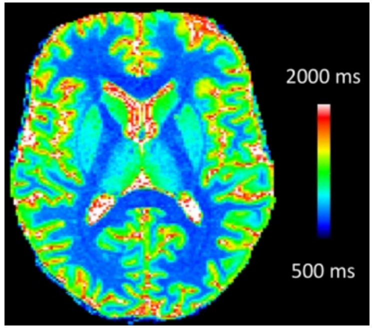
$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \varepsilon \quad \Longrightarrow \quad \boldsymbol{x} = \mathcal{M}(\boldsymbol{A}, \boldsymbol{y}, \textit{priors})$$



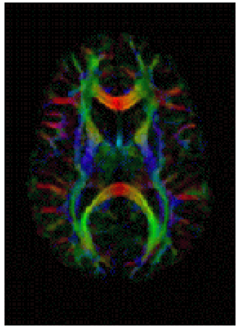
Anatomical MRI



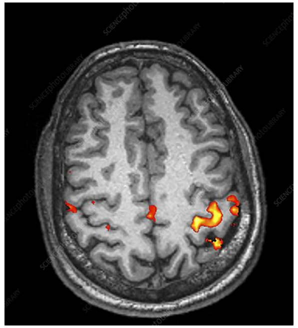
Cardiac /CINE



Quantitative MRI



Diffusion MRI

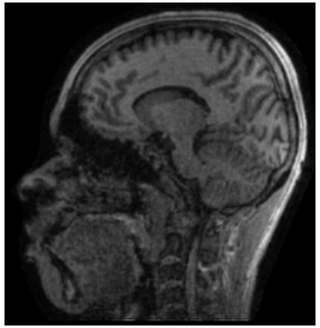


Functional MRI

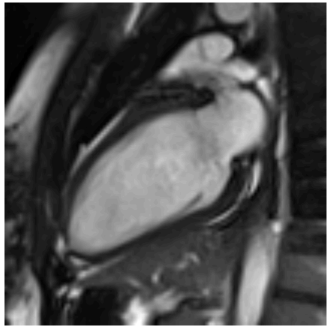
MRI Reconstruction

- Solving an (or many) *ill-posed* **inverse problem**

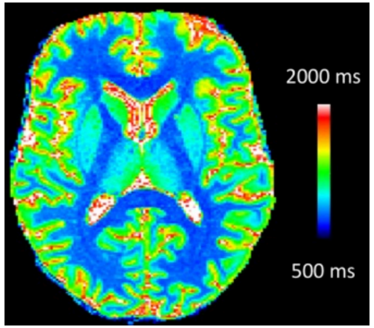
$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \varepsilon \quad \Longrightarrow \quad \boldsymbol{x} = \mathcal{M}(\boldsymbol{A}, \boldsymbol{y}, \textit{priors})$$



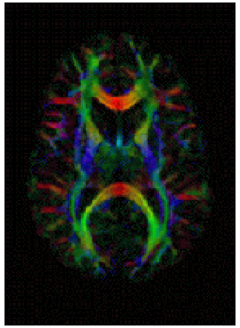
Anatomical MRI



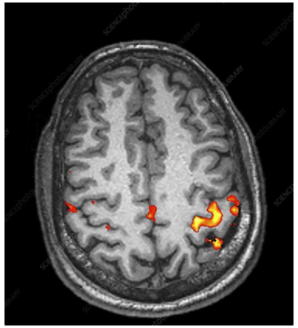
Cardiac /CINE



Quantitative MRI



Diffusion MRI



Functional MRI

Common Goal: Solve the inverse problem

Design (Learn ?) and validate $\mathcal{M}$, $\boldsymbol{A}$ and *priors*

Open-source software

- Open source vs free software
  - “Free as in beer vs free as in freedom”

Open-source software

- Open source vs free software
  - “Free as in beer vs free as in freedom”
- Open the black box
  - Access to source code, and (sometimes) freedom of modification
- Licences to (un)protect the code !

Open-source software

- Open source vs free software
  - "Free as in beer vs free as in freedom"
- Open the black box
  - Access to source code, and (sometimes) freedom of modification
- Licences to (un)protect the code !
- Good license choices
  - MIT, BSD-3-clause, (A)GPL[1]
  - Or whatever your legal departement requires

[1] https://opensource.stackexchange.com/questions/217/what-are-the-essential-differences-between-the-bsd-and-mit-licences

Where do we stand with open-source in MRI ?

- Acquisition:
  Pulseq, MTRK, …

Where do we stand with open-source in MRI ?

- Acquisition:
  Pulseq, MTRK, …

- Simulation
  KomaMRI.jl, Jemris, *SNAKE*, …

Where do we stand with open-source in MRI ?

- Acquisition:
  Pulseq, MTRK, …
- Simulation
  KomaMRI.jl, Jemris, *SNAKE*, …
- Open-source Hardware
  Low-Field MRIs

Where do we stand with open-source in MRI ?

- Acquisition:
  Pulseq, MTRK, …
- Simulation
  KomaMRI.jl, Jemris, *SNAKE*, …
- Open-source Hardware
  Low-Field MRIs
- Reconstruction Software

$\Rightarrow$ This talk !

Where do we stand with open-source in MRI ?

- Acquisition:
  Pulseq, MTRK, …
- Simulation
  KomaMRI.jl, Jemris, *SNAKE*, …
- Open-source Hardware
  Low-Field MRIs
- Reconstruction Software
- Analysis
  Nilearn, **f/s/d mriprep**, …

⇒ This talk !

1. MRI Reconstruction software

Software usage in the Literature

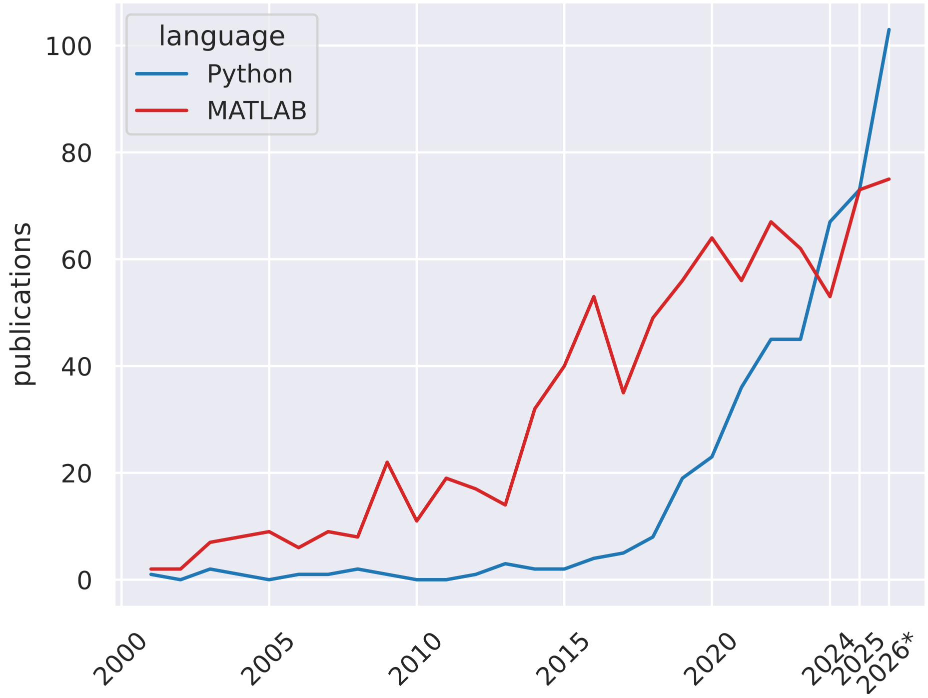


Figure 1: Published articles with *“MRI”+Language* in their abstract or title.

2026: Extrapolated results from Q1

Software usage in the Literature

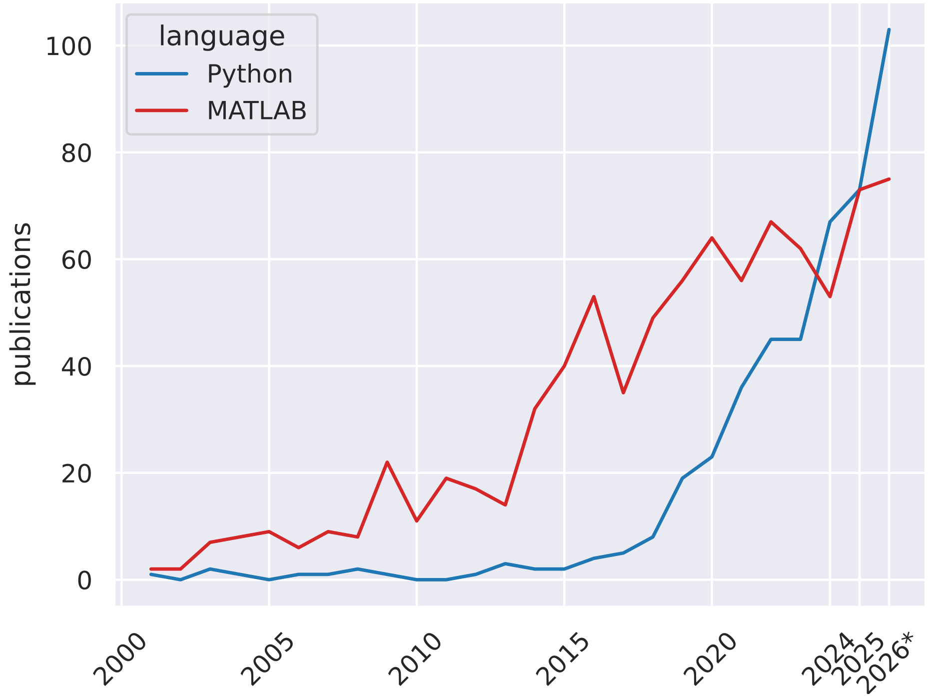


Figure 1: Published articles with *“MRI”+Language* in their abstract or title.

2026: Extrapolated results from Q1

- Python adoption accelerates, and has stronger ecosystem
  - LLMs loves it
  - PyTorch / JAX for deep learning and GPU based reconstruction
  - 2026 is (finally) the year of Python for MRI ?

Software usage in the Literature

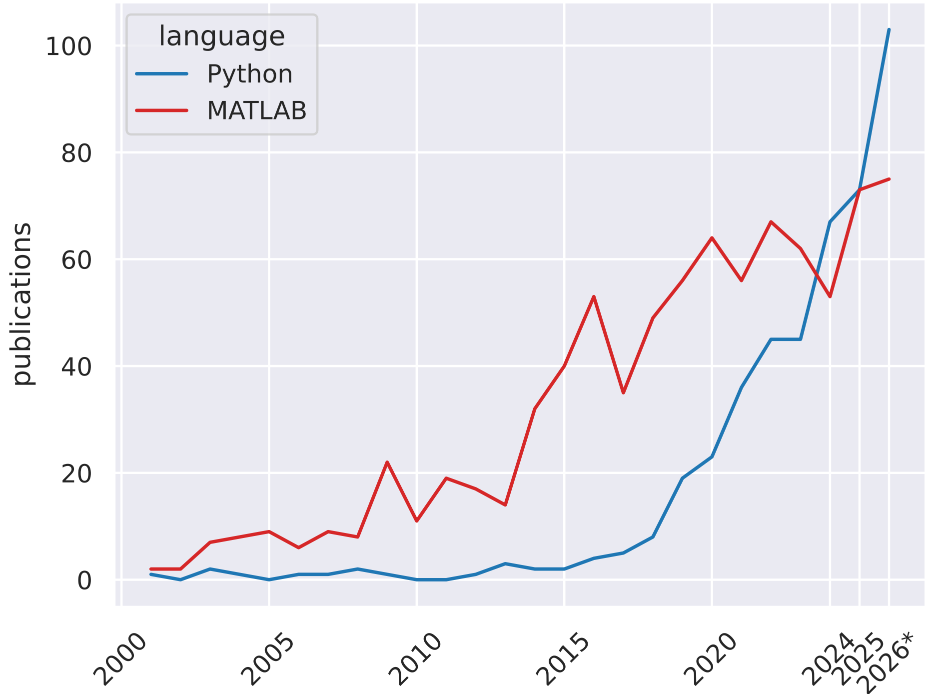


Figure 1: Published articles with *“MRI”+Language* in their abstract or title.

2026: Extrapolated results from Q1

- Python adoption accelerates, and has stronger ecosystem
  - LLMs loves it
  - PyTorch / JAX for deep learning and GPU based reconstruction
  - 2026 is (finally) the year of Python for MRI ?
- MATLAB is still widely use (legacy code base)
  - But Julia grows !

Open Source Repositories

- $\simeq$ 150 Repos filter with topic `mri` & `mri-reconstruction`

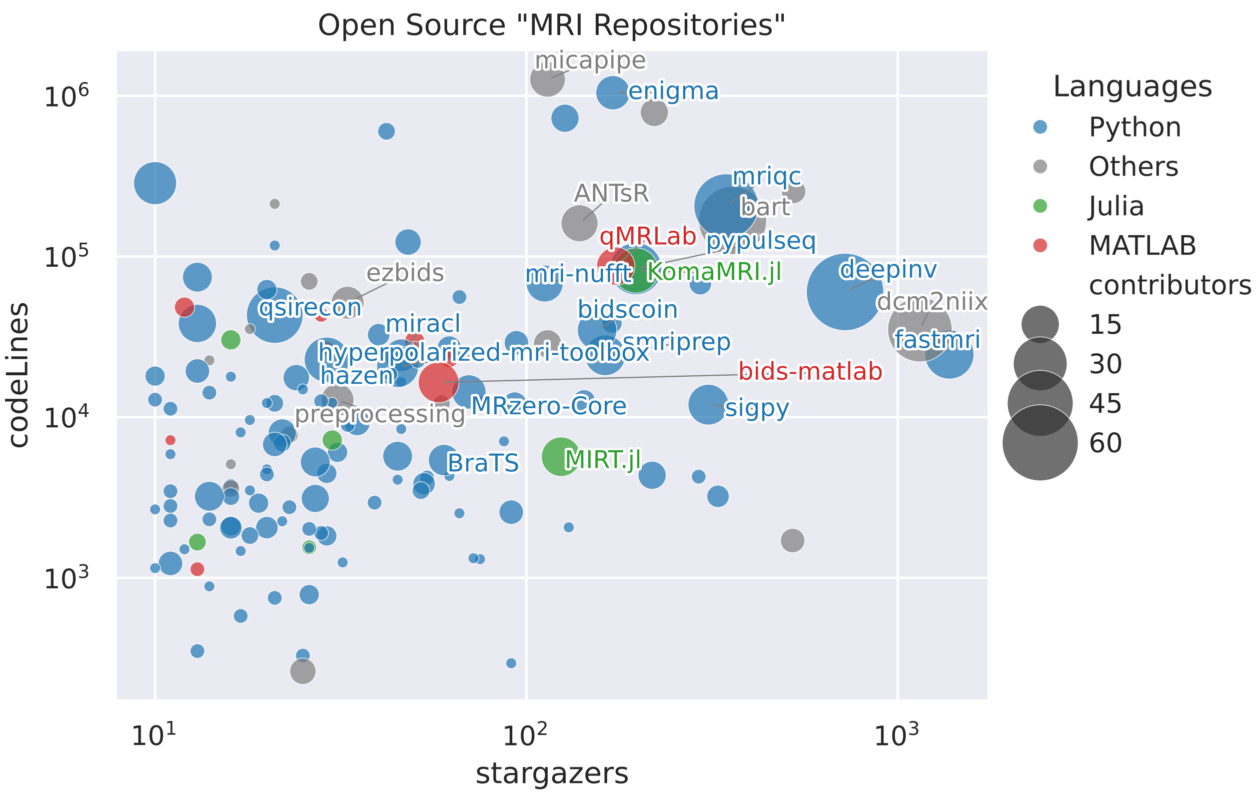


Figure 2: Main GitHub Repositories with 'mri' topic

Open Source Repositories

- $\simeq$ 150 Repos filter with topic `mri` & `mri-reconstruction`
- Python/Notebook is the most represented on GitHub
- Julia is a strong and growing alternative (esp. for MATLABers)

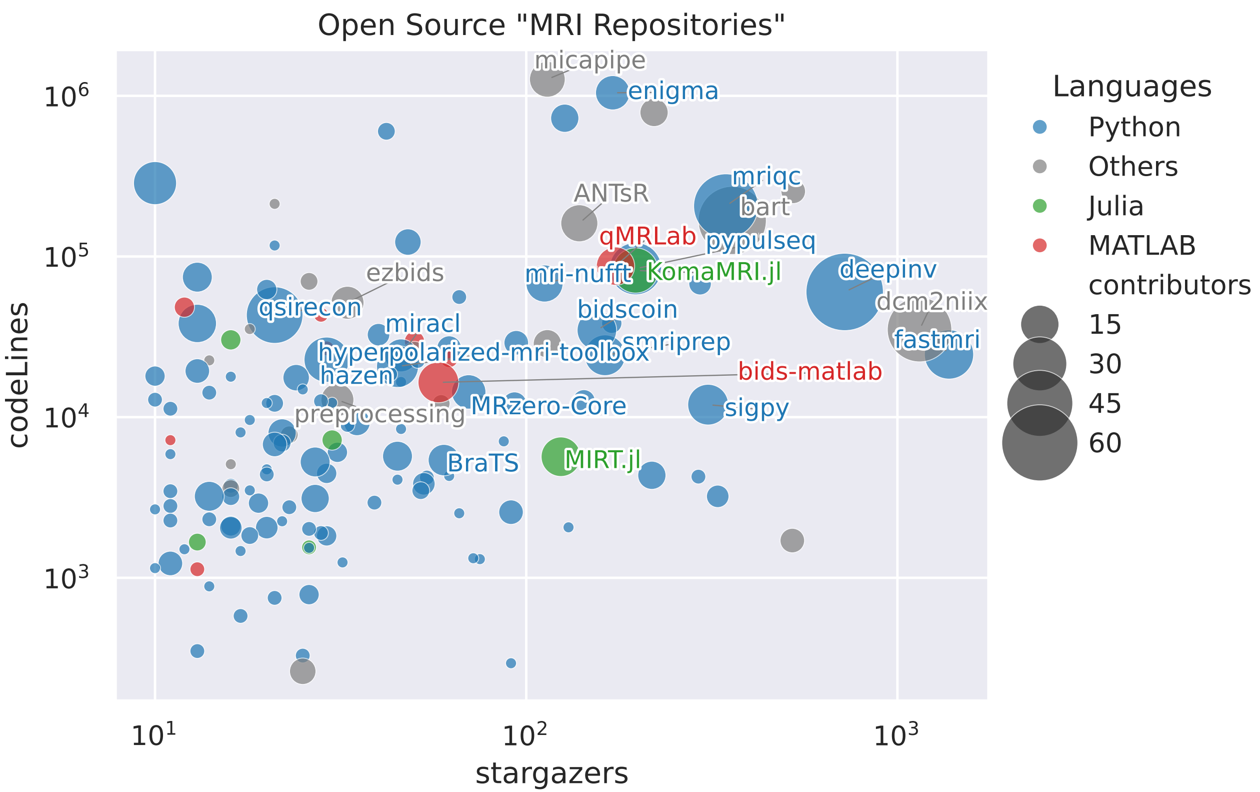


Figure 2: Main GitHub Repositories with 'mri' topic

Open Source Repositories

- $\simeq$ 150 Repos filter with topic `mri` & `mri-reconstruction`
- Python/Notebook is the most represented on GitHub
- Julia is a strong and growing alternative (esp. for MATLABers)

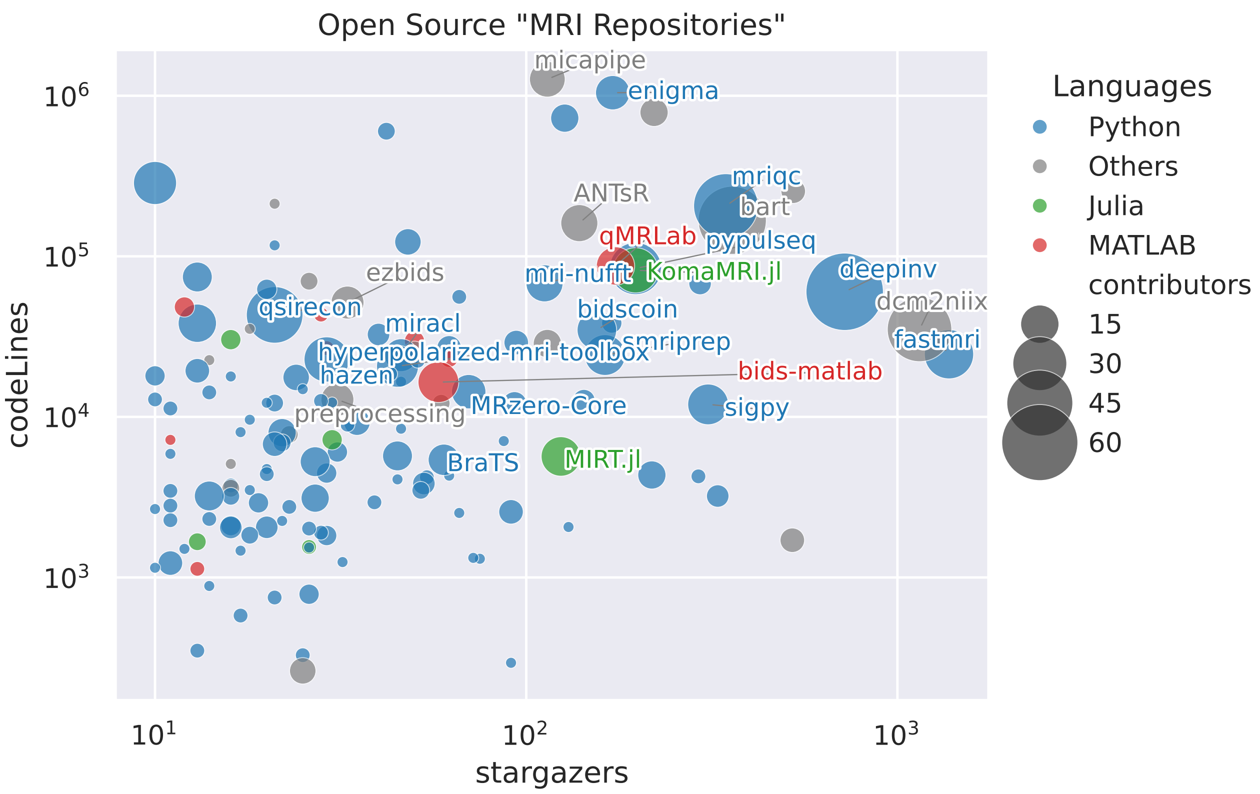


Figure 2: Main GitHub Repositories with 'mri' topic

Open Source Repositories

- $\simeq$ 150 Repos filter with topic `mri` & `mri-reconstruction`
- Python/Notebook is the most represented on GitHub
- Julia is a strong and growing alternative (esp. for MATLABers)
- Only 56% report homepage/ documentation!

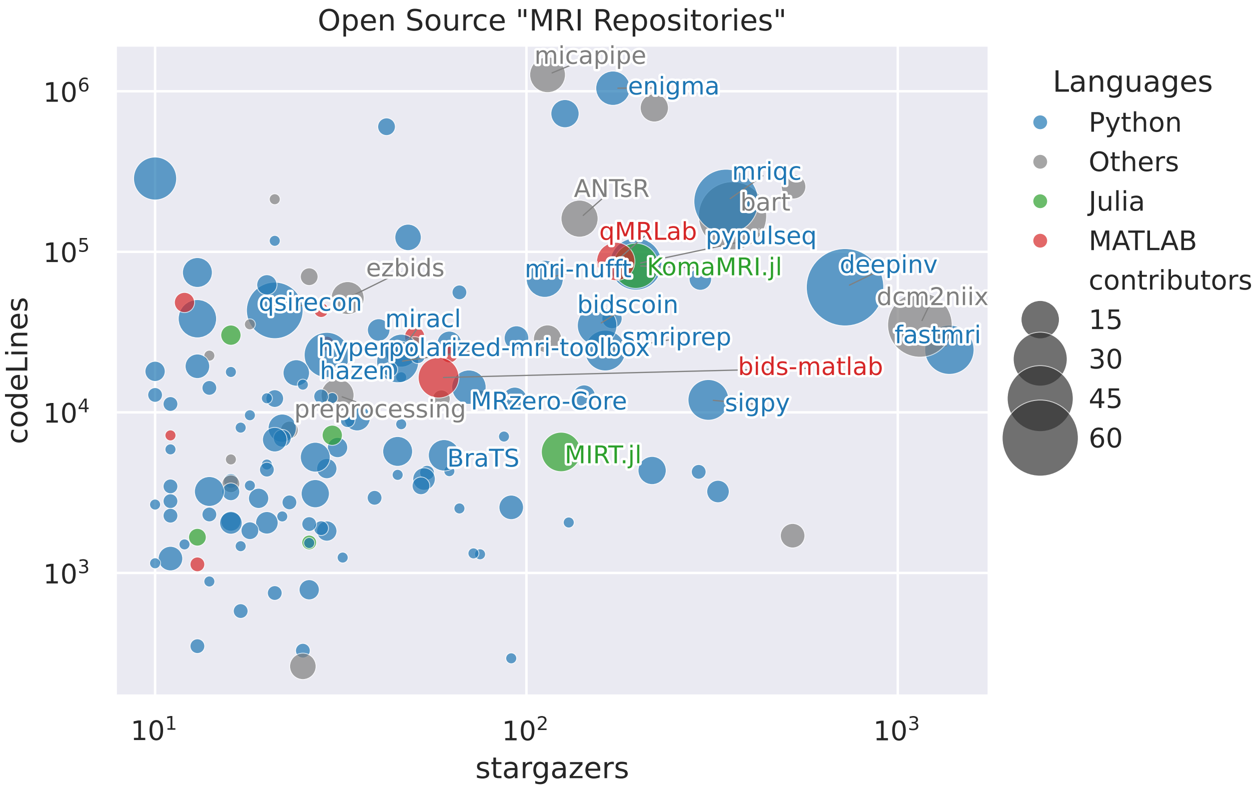


Figure 2: Main GitHub Repositories with 'mri' topic

Catch them all !

Catch them all !

Catch them all !

Deepinv ⭐726

Deep Inverse

BART ⭐363

BART

Catch them all !

Deepinv ⭐726

Deep Inverse

BART ⭐363

BART

SigPy ⭐337

SigPy

Catch them all !

Deepinv ⭐726

BART ⭐363

SigPy ⭐337

DIRECT ⭐301

MIRT ⭐204

SCICO ⭐162

Catch them all !

Deepinv ⭐726

BART ⭐363

SigPy ⭐337

DIRECT ⭐301

MIRT ⭐204

SCICO ⭐162

MIRT.jl ⭐124

MRIReco.jl ⭐98

PySAP ⭐48

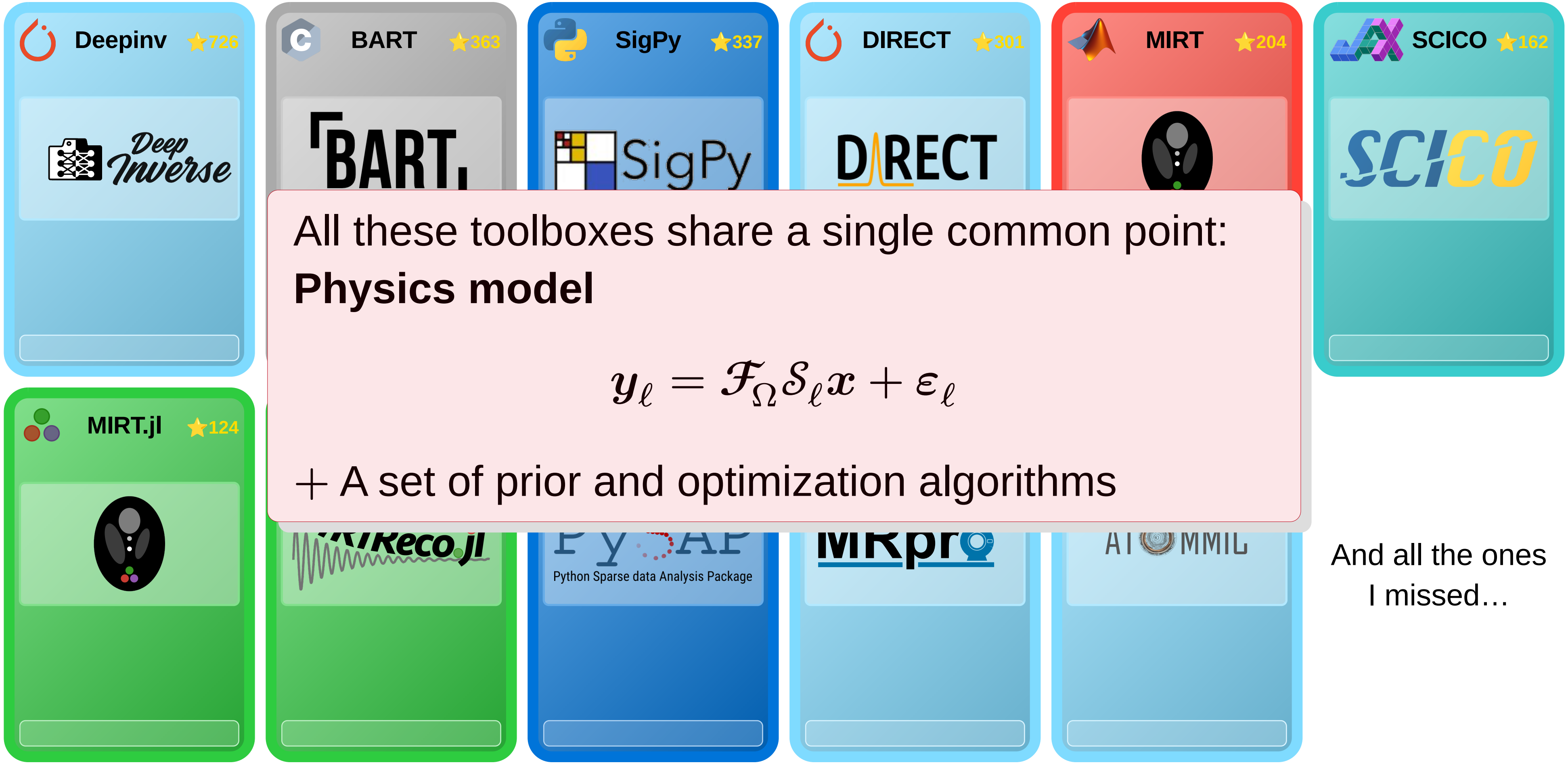
Python Sparse data Analysis Package

MRPro ⭐48

Atommic ⭐23

And all the ones I missed…

Catch them all !



All these toolboxes share a single common point:
Physics model

$$y_\ell = \mathcal{F}_\Omega \mathcal{S}_\ell x + \varepsilon_\ell$$

$+$ A set of prior and optimization algorithms

And all the ones I missed…

2. MRI-NUFFT: Case study

Introducing MRI-NUFFT

- Most MR Toolboxes focuses on Cartesian sampling

Introducing MRI-NUFFT

- Most MR Toolboxes focuses on Cartesian sampling
- Non-Cartesian Reconstruction has great potential
  - ✓ Freedom of sampling locations
  - ✓ Increased sampling efficacy
  - ✓ Robustness to Motion

Introducing MRI-NUFFT

- Most MR Toolboxes focuses on Cartesian sampling
- Non-Cartesian Reconstruction has great potential
  - ✓ Freedom of sampling locations
  - ✓ Increased sampling efficacy
  - ✓ Robustness to Motion
- But …
  - ✗ Require Expensive Non-Uniform Fourier Transform
  - ✗ Off-Resonance effects
  - ✗ (Re)-design non-Cartesian trajectories for sequences
  - ✗ MR System constraints

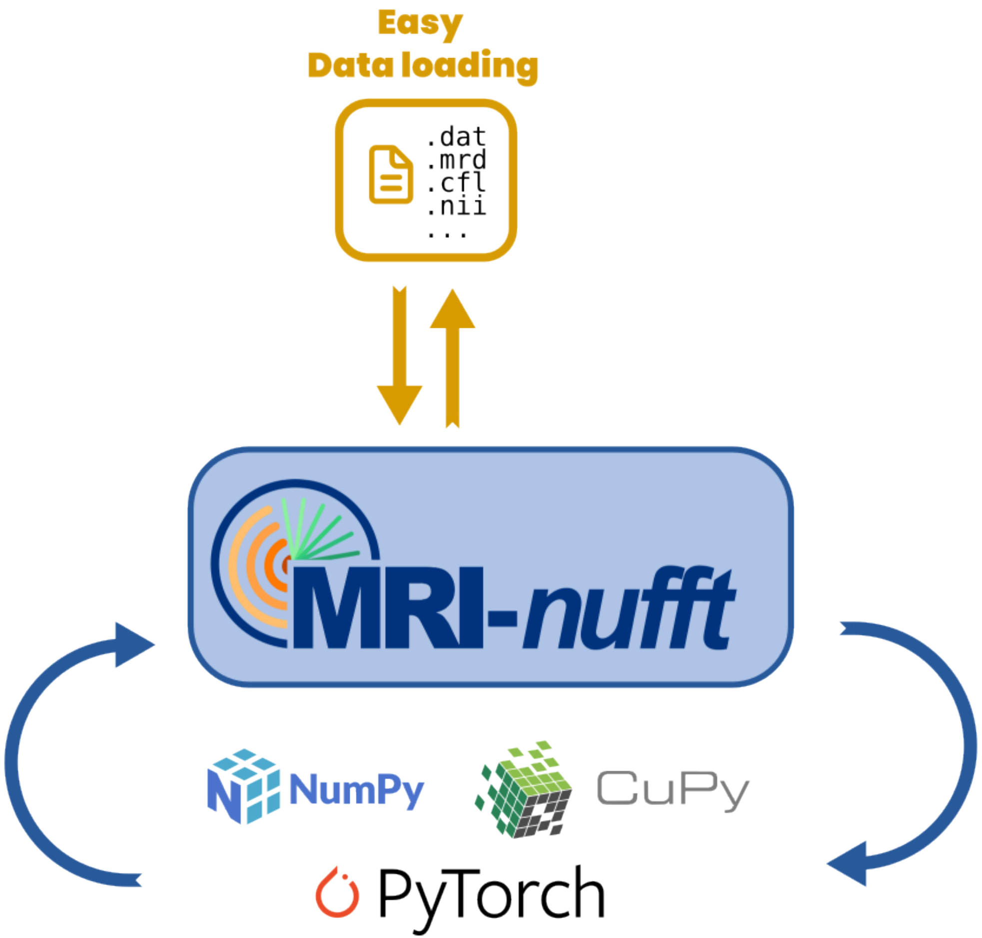
Introducing MRI-NUFFT

- Most MR Toolboxes focuses on Cartesian sampling
- Non-Cartesian Reconstruction has great potential
  - ✓ Freedom of sampling locations
  - ✓ Increased sampling efficacy
  - ✓ Robustness to Motion
- But …
  - ✗ Require Expensive Non-Uniform Fourier Transform
  - ✗ Off-Resonance effects
  - ✗ (Re)-design non-Cartesian trajectories for sequences
  - ✗ MR System constraints

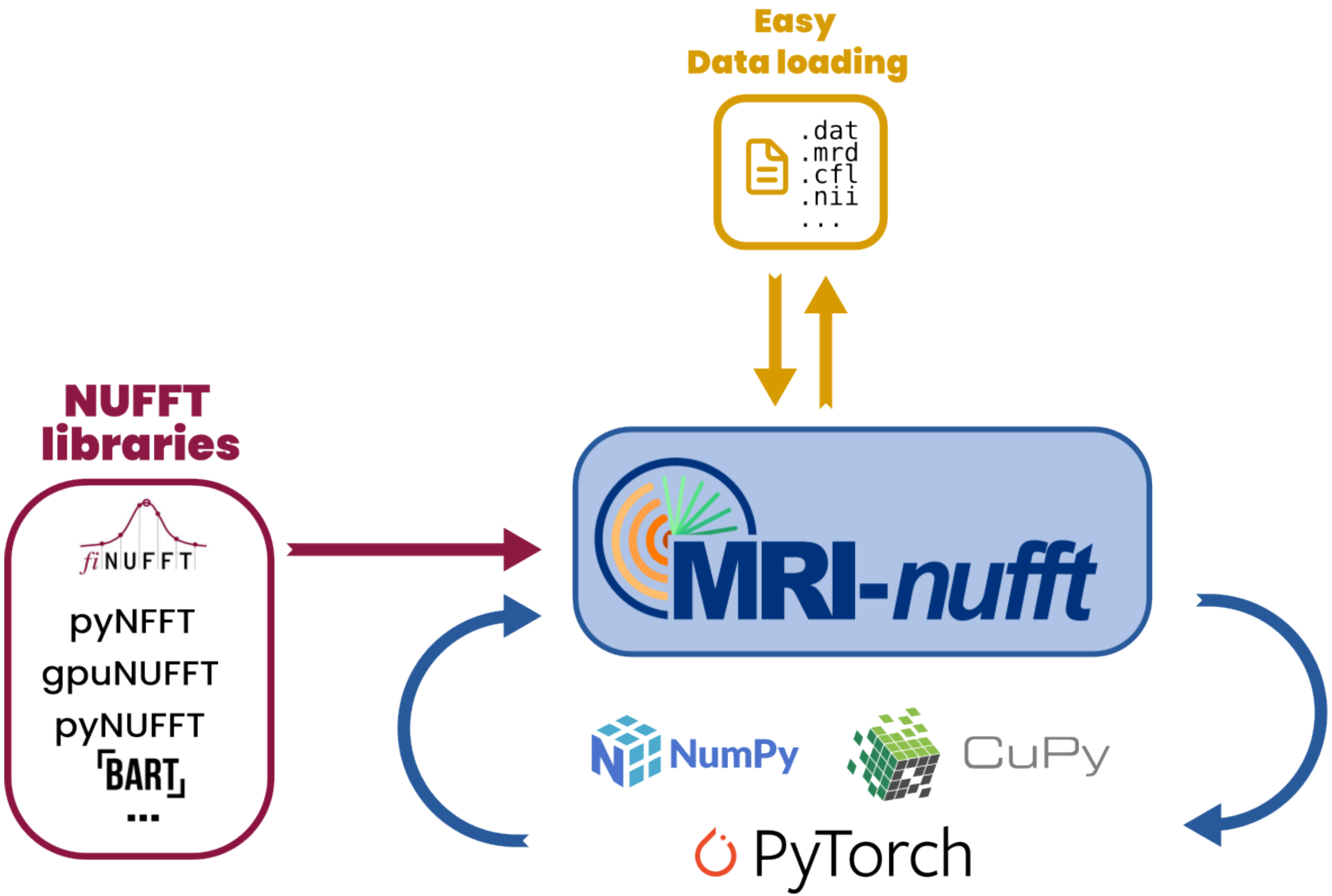
Let's fix this with:

MRI-nufft

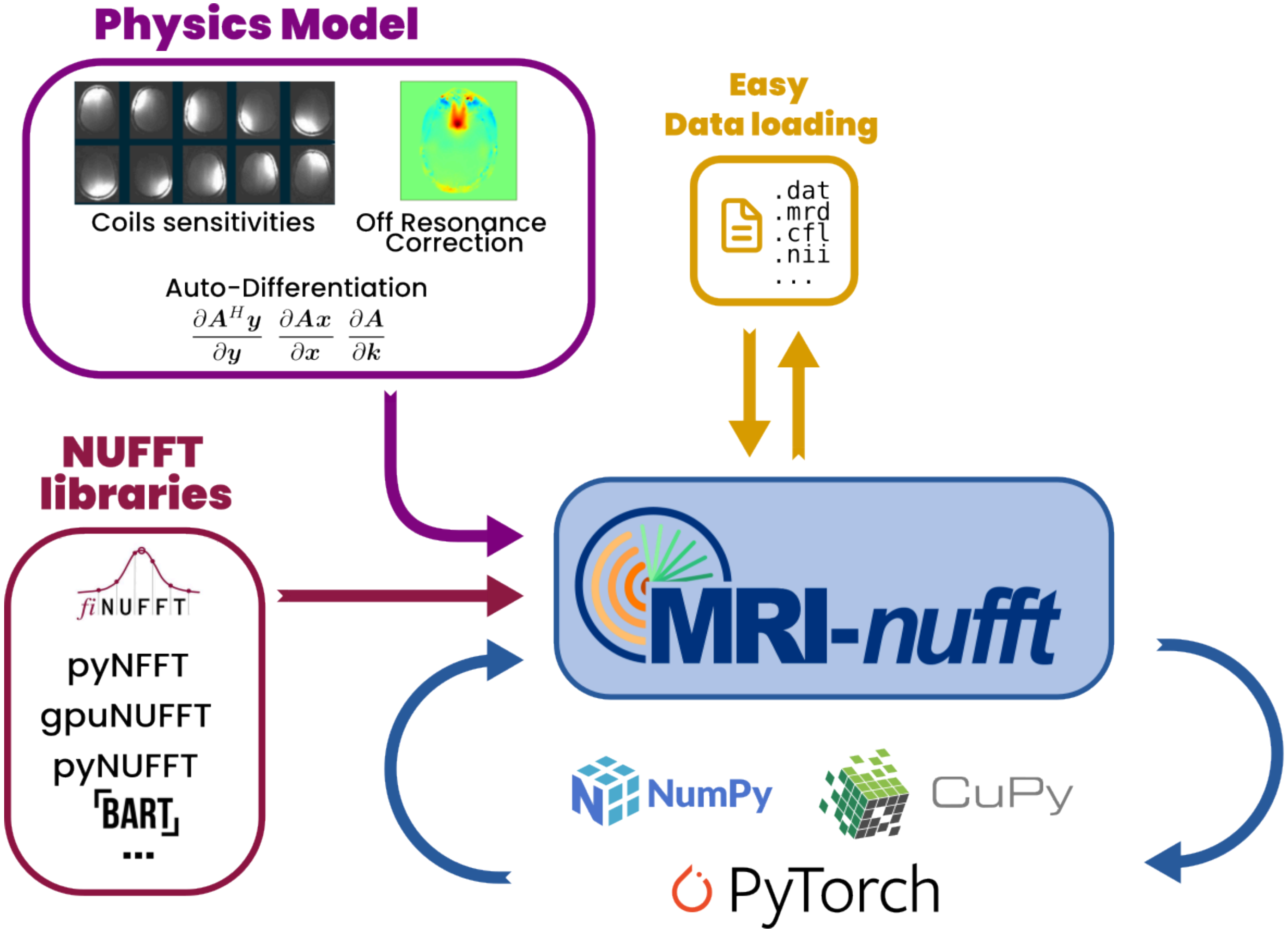
What is MRI-NUFFT ?



What is MRI-NUFFT ?



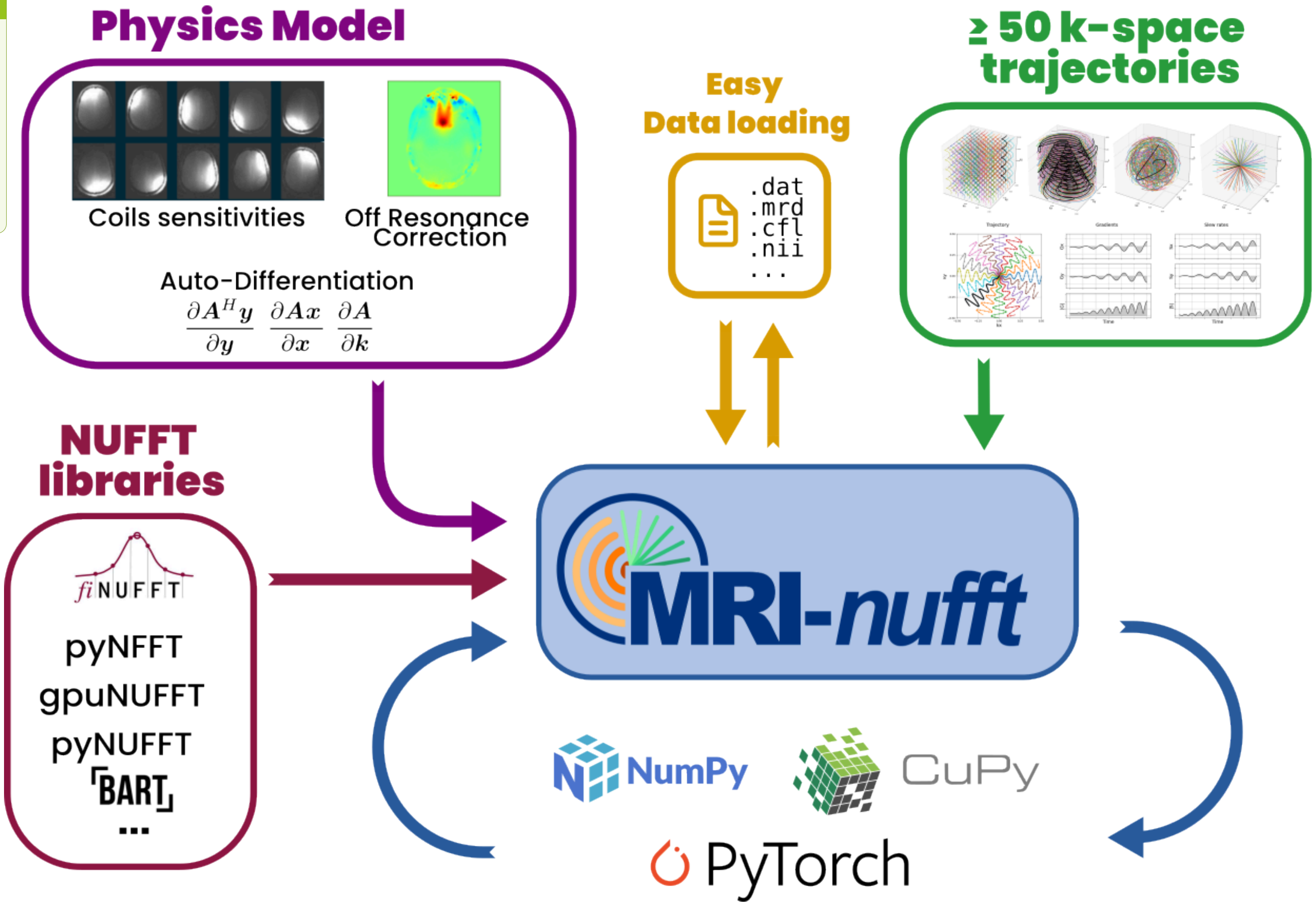
What is MRI-NUFFT ?



What is MRI-NUFFT ?

Learn more

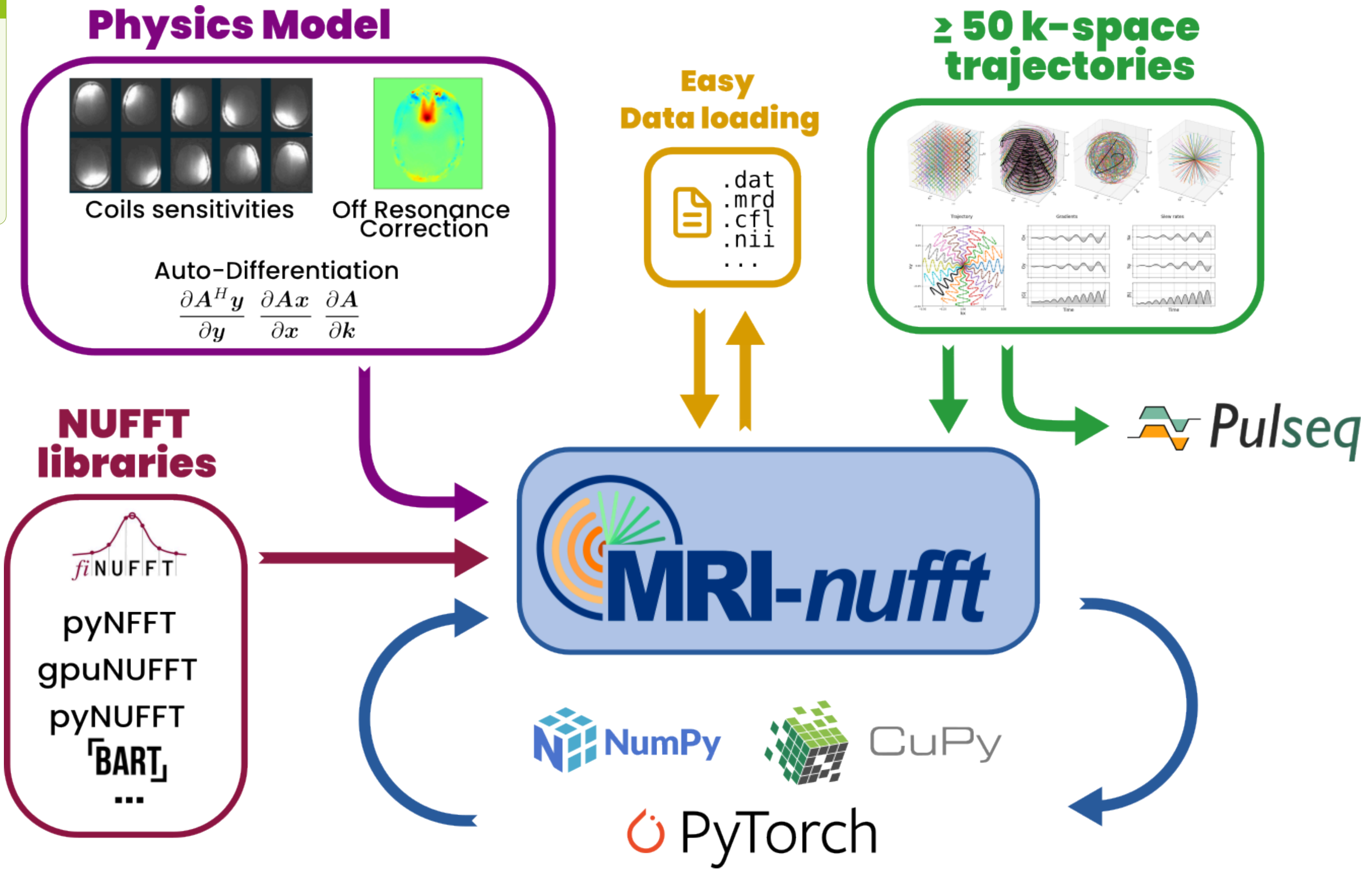
AI Driven k-space Sampling
Mon. 11/05 7:30-8 AM
Chaithya GR.



What is MRI-NUFFT ?

Learn more

AI Driven k-space Sampling
Mon. 11/05 7:30-8 AM
Chaithya GR.



NUFFT for MRI Reconstruction

- Inverse problem formulation for MRI

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{x} \in \mathbb{C}^N} \frac{1}{2} \sum_{\ell=1}^{L} \|\mathcal{F}_\Omega \mathcal{S}_\ell \boldsymbol{x} - \boldsymbol{y}_\ell\|_2^2 + \lambda g(\boldsymbol{x})$$

NUFFT for MRI Reconstruction

- Inverse problem formulation for MRI

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\sum_{\ell=1}^{L}\|\mathcal{F}_\Omega\mathcal{S}_\ell\boldsymbol{x} - \boldsymbol{y}_\ell\|_2^2 + \lambda g(\boldsymbol{x})$$

> *"Priors and Networks are temporary, …*
>
> *… Forward model is for ever"*

NUFFT for MRI Reconstruction

- Inverse problem formulation for MRI

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{C}^N} \boxed{\frac{1}{2}\sum_{\ell=1}^{L}\|\mathcal{F}_\Omega\mathcal{S}_\ell\boldsymbol{x} - \boldsymbol{y}_\ell\|_2^2} + \lambda g(\boldsymbol{x})$$

“Priors and Networks are temporary, …

… Forward model is for ever“

NUFFT for MRI Reconstruction

- Inverse problem formulation for MRI

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\sum_{\ell=1}^{L} \|\mathcal{F}_\Omega \mathcal{S}_\ell \boldsymbol{x} - \boldsymbol{y}_\ell\|_2^2 + \lambda g(\boldsymbol{x})$$

"Priors and Networks are temporary, …

… Forward model is for ever"

MRI-NUFFT provides *efficient* implementation for the forward model operator:

- based on the NUFFT backend of your choice.
- Handles CPU/GPU transfers and array conversion (`torch/numpy/cupy`)
- `autograd` support $\frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{A}^H\boldsymbol{y}}{\partial \boldsymbol{y}}, \frac{\partial \boldsymbol{A}_\theta \boldsymbol{x}}{\partial \theta}, \ldots$

NUFFT for MRI Reconstruction

- Inverse problem formulation for MRI

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{C}^N} \boxed{\frac{1}{2}\sum_{\ell=1}^{L} \|\mathcal{F}_\Omega \mathcal{S}_\ell \boldsymbol{x} - \boldsymbol{y}_\ell\|_2^2} + \lambda g(\boldsymbol{x})$$

> *“Priors and Networks are temporary, …*
>
> *… Forward model is for ever“*

MRI-NUFFT provides *efficient* implementation for the forward model operator:

- based on the NUFFT backend of your choice.
- Handles CPU/GPU transfers and array conversion (`torch/numpy/cupy`)
- `autograd` support $\frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{A}^H\boldsymbol{y}}{\partial \boldsymbol{y}}, \frac{\partial \boldsymbol{A}_\theta \boldsymbol{x}}{\partial \theta}, \ldots$

```
from mrinufft import get_operator
nufft = get_operator("cufinufft", samples, shape, smaps=smaps,
density="pipe")
ksp = nufft.op(image)          # y = Ax
adjoint = nufft.adj_op(ksp)    # x = A^H y
pinv = nufft.pinv_solver(ksp) # x = A^† y, using CG
```

Extended Fourier Model

- Multi-coil, subspace representations, …
-

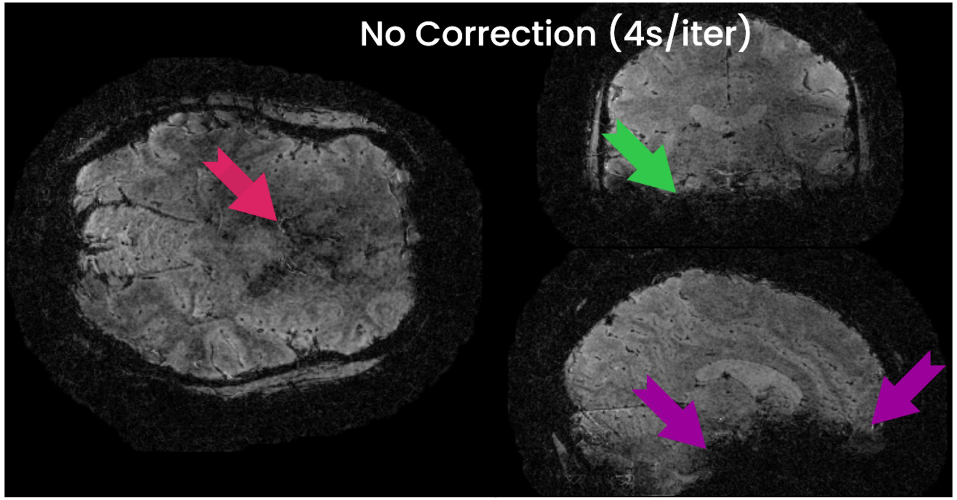
Extended Fourier Model

- Multi-coil, subspace representations, …
- Density Compensation (`voronoi`, `pipe`, …)
-

Extended Fourier Model

- Multi-coil, subspace representations, …
- Density Compensation (`voronoi`, `pipe`, …)
- Off-Resonance

$$\boldsymbol{y}_\ell(t) = \int_{FOV} \boldsymbol{x}(\boldsymbol{r})\mathcal{S}_\ell(\boldsymbol{r})e^{-\Delta\omega(\boldsymbol{r})t}e^{-2i\pi\boldsymbol{k}(t)\cdot\boldsymbol{r}}\mathrm{d}\boldsymbol{r}$$

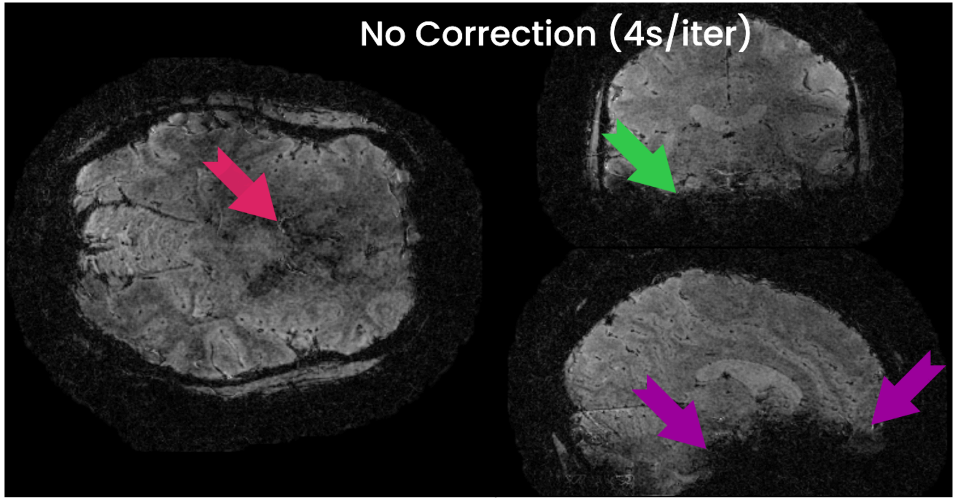


Extended Fourier Model

- Multi-coil, subspace representations, ...
- Density Compensation (`voronoi`, `pipe`, ...)
- Off-Resonance correction via bilinearization (MTI, MFI)

$$\boldsymbol{y}_\ell(t) = \int_{FOV} \boldsymbol{x}(\boldsymbol{r}) \mathcal{S}_\ell(\boldsymbol{r}) e^{-\Delta\omega(\boldsymbol{r})t} e^{-2i\pi \boldsymbol{k}(t)\cdot\boldsymbol{r}} \mathrm{d}\boldsymbol{r}$$

$$\boldsymbol{y}_\ell \approx \sum_{p=1}^{P} \boldsymbol{b}_p \sum_{\ell=1}^{L} \mathcal{F}_\Omega(\boldsymbol{c}_p \mathcal{S}_\ell \boldsymbol{x})$$

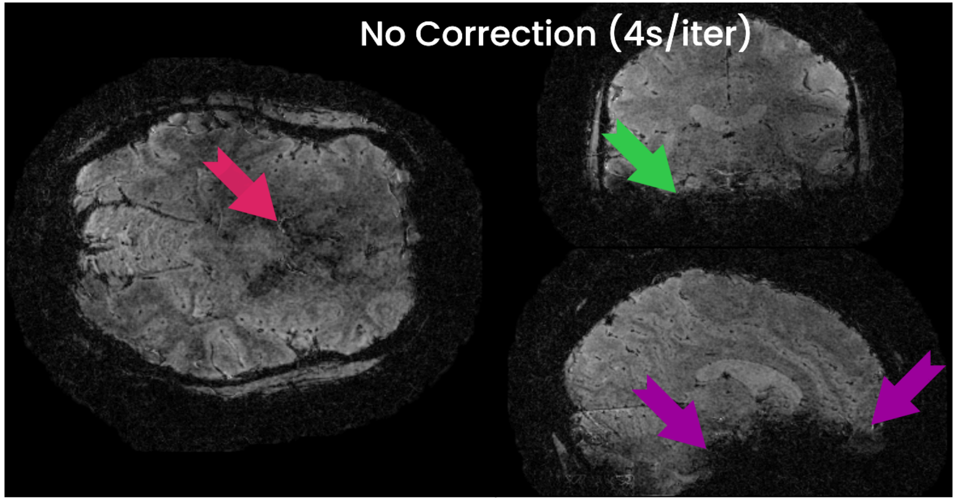


Extended Fourier Model

- Multi-coil, subspace representations, …
- Density Compensation (`voronoi`, `pipe`, …)
- Off-Resonance correction via bilinearization (MTI, MFI) via bilinearization (MTI, MFI, **SVD**)

$$\boldsymbol{y}_\ell(t) = \int_{FOV} \boldsymbol{x}(\boldsymbol{r})\mathcal{S}_\ell(\boldsymbol{r}) e^{-\Delta\omega(\boldsymbol{r})t} e^{-2i\pi \boldsymbol{k}(t)\cdot\boldsymbol{r}} \mathrm{d}\boldsymbol{r}$$

$$\min_{\substack{B\in\mathbb{C}^{M\times P}\\ C\in\mathbb{C}^{P\times N}}} \|e^{-\Delta\omega(\boldsymbol{r})t} - BC\|_2^2 \implies \boldsymbol{y}_\ell \approx \sum_{p=1}^{P} \boldsymbol{b}_p \sum_{\ell=1}^{L} \mathcal{F}_\Omega(\boldsymbol{c}_p \mathcal{S}_\ell \boldsymbol{x})$$



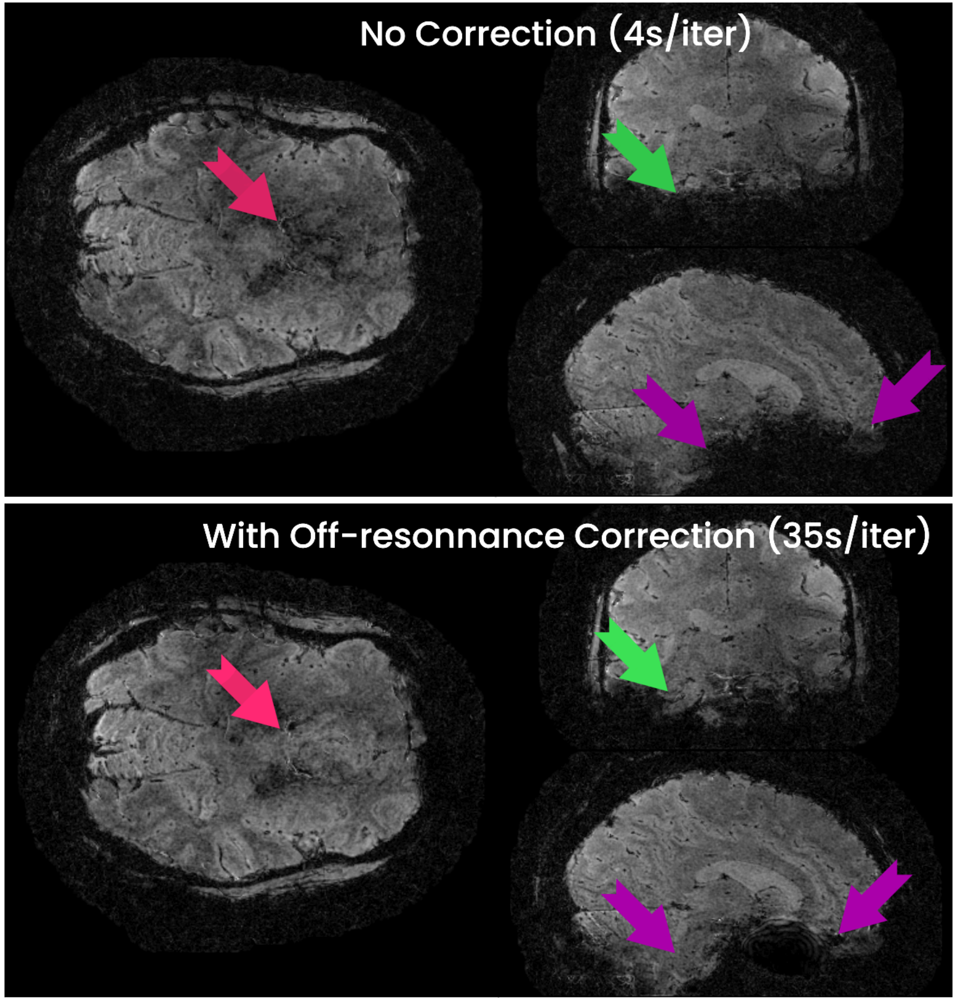
Extended Fourier Model

- Multi-coil, subspace representations, …
- Density Compensation (`voronoi`, `pipe`, …)
- Off-Resonance correction via bilinearization (MTI, MFI) via bilinearization (MTI, MFI, **SVD**)

$$\boldsymbol{y}_\ell(t) = \int_{FOV} \boldsymbol{x}(\boldsymbol{r})\mathcal{S}_\ell(\boldsymbol{r})e^{-\Delta\omega(\boldsymbol{r})t}e^{-2i\pi\boldsymbol{k}(t)\cdot\boldsymbol{r}}\mathrm{d}\boldsymbol{r}$$

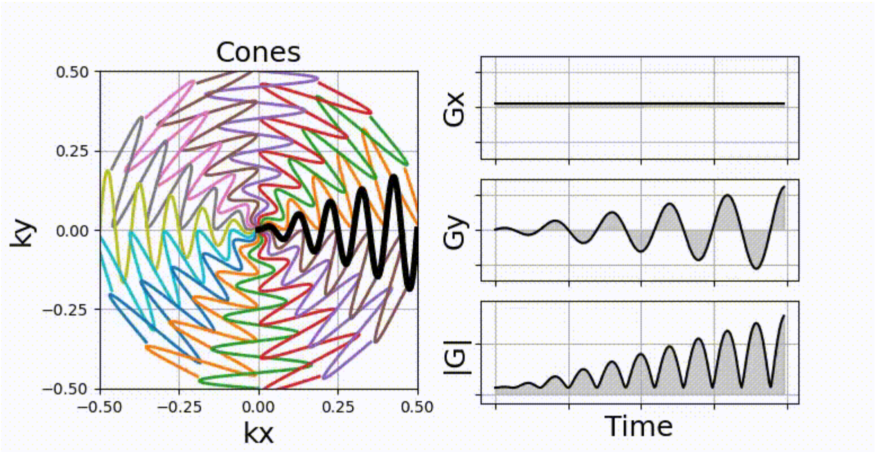
$$\min_{\substack{B\in\mathbb{C}^{M\times P}\\ C\in\mathbb{C}^{P\times N}}} \|e^{-\Delta\omega(\boldsymbol{r})t} - BC\|_2^2 \Longrightarrow \boldsymbol{y}_\ell \approx \sum_{p=1}^{P} b_p \sum_{\ell=1}^{L} \mathcal{F}_\Omega(c_p \mathcal{S}_\ell \boldsymbol{x})$$

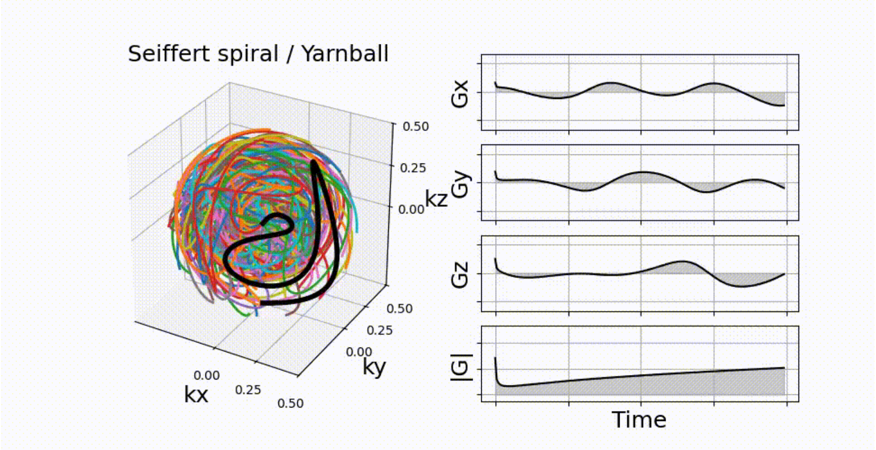
```
nufft_orc = nufft.with_off_resonance_correction(
    field_map=field_map,
    readout_time=readout_time,
    interpolators=20)
adj_orc = nufft_orc.adj_op(ksp) # x = A^H_{Δω} y
pinv_orc = nufft_orc.pinv_solver(ksp) # x = A^†_{Δω} y
```



Trajectories & Gradients

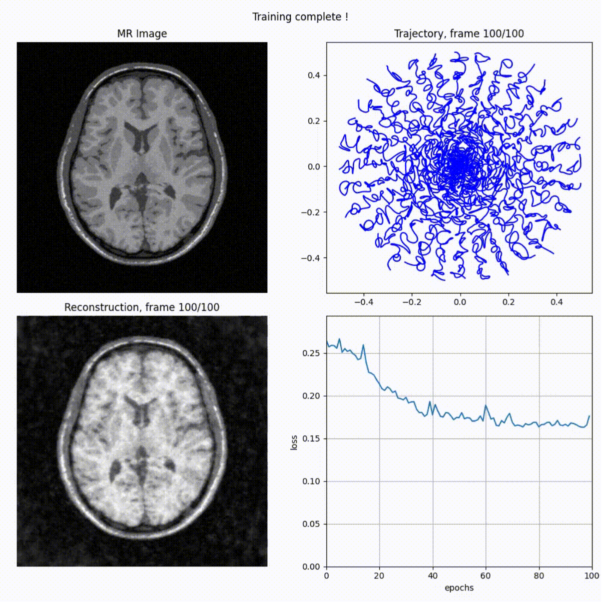
- A large collection of non-Cartesian trajectories
  - 2D and 3D, fully parameterized

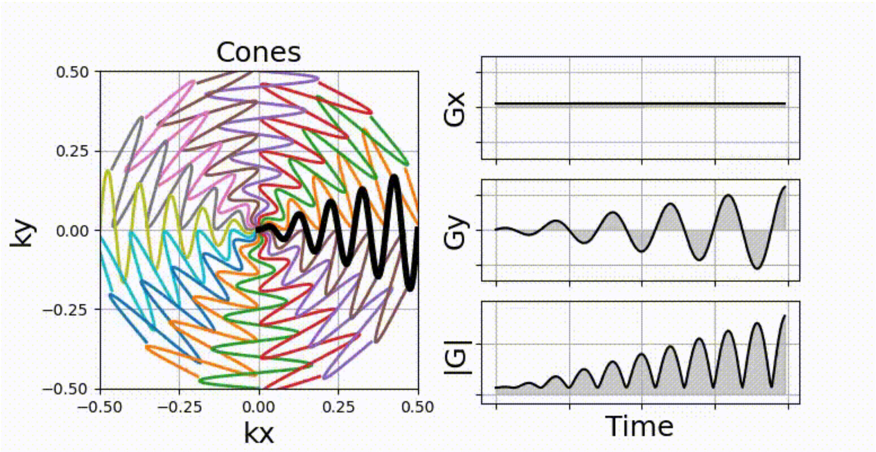


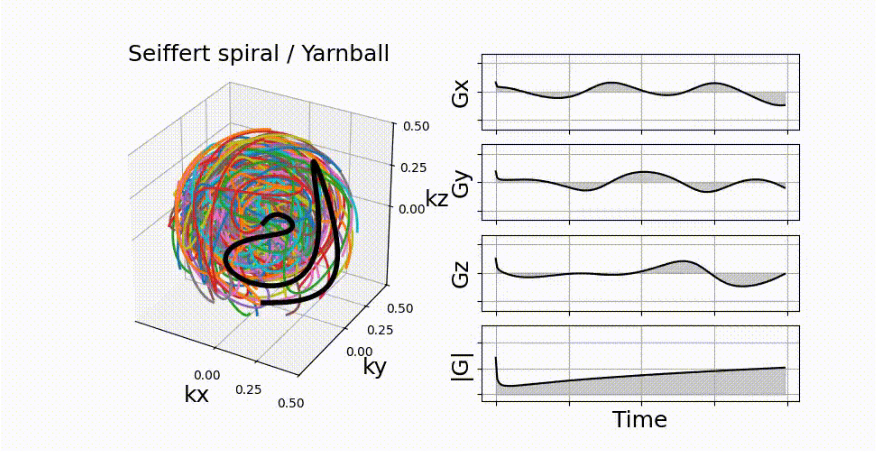


Trajectories & Gradients

- A large collection of non-Cartesian trajectories
  - 2D and 3D, fully parameterized
  - Hardware compliance enforcement.
  - You can even learn trajectories in a deep Learning setting with autodifferentiation !

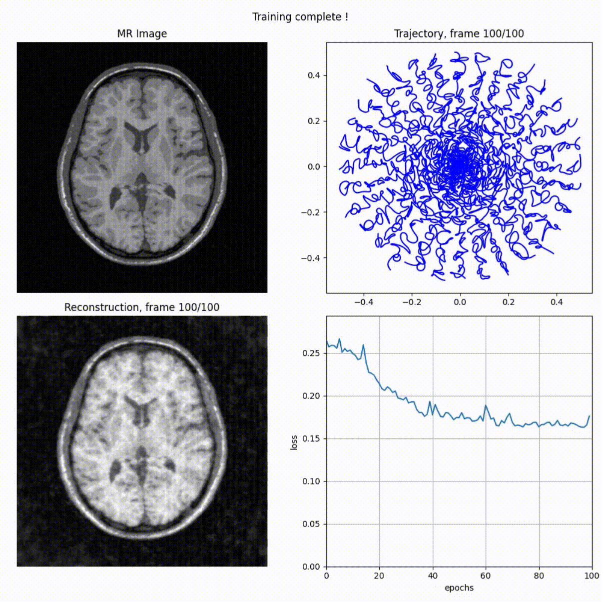






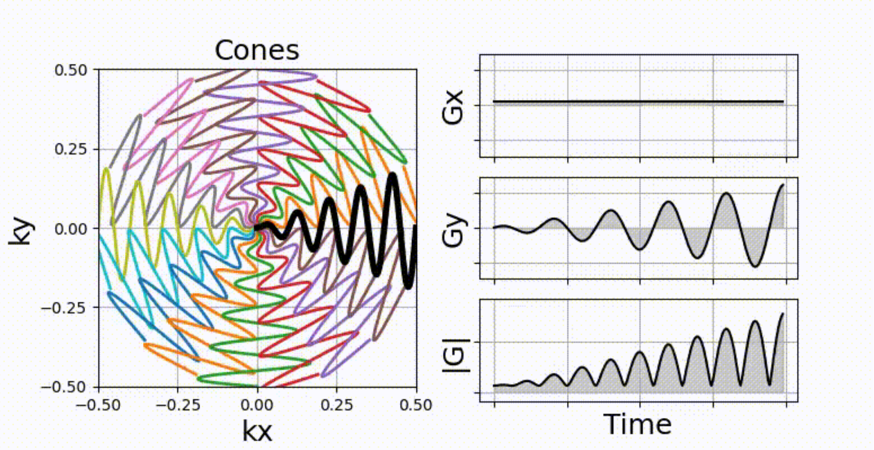
Trajectories & Gradients

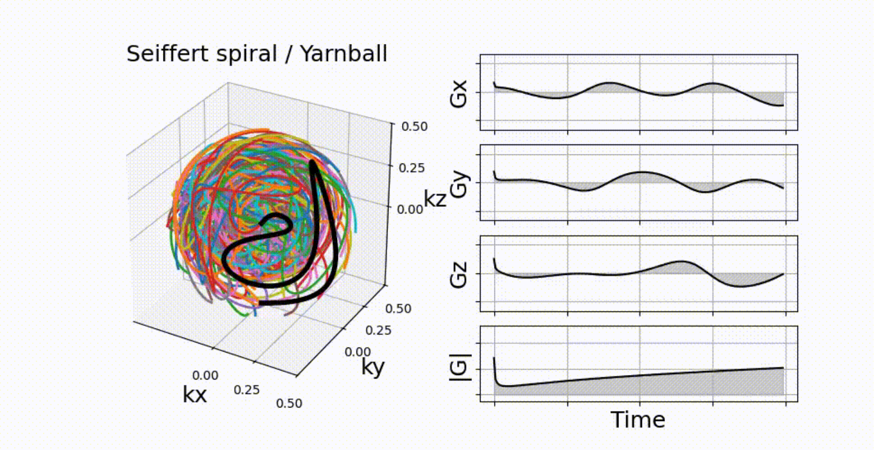
- A large collection of non-Cartesian trajectories
  - 2D and 3D, fully parameterized
  - Hardware compliance enforcement.
  - You can even learn trajectories in a deep Learning setting with autodifferentiation !



Live Demo !

Open in Colab





Integration with other libraries

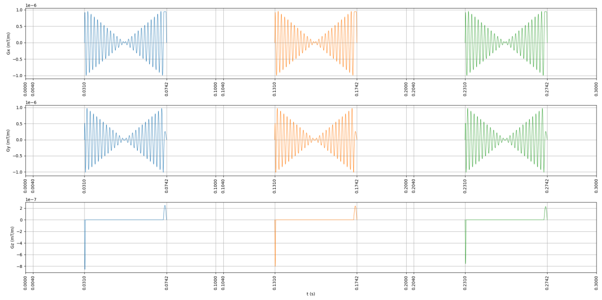
- $\geq$ 20 Projects depends on MRI-NUFFT[1]
- Integration with external packages

[1] https://github.com/mind-inria/mri-nufft/network/dependents

Integration with other libraries

- $\geq$ 20 Projects depends on MRI-NUFFT[1]
- Integration with external packages
  - (Py)Pulseq

```python
from mrinufft.trajectories import (pulseq_gre,
    read_pulseq_traj)
traj = ...
seq = pulseq_gre(traj, acq=acq)
seq.write("spiral_gre.seq")
traj_read = read_pulseq_traj(seq)
```

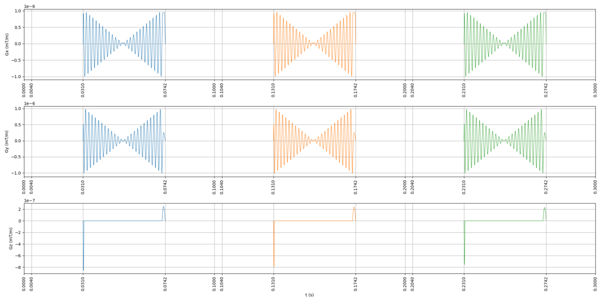


[1] https://github.com/mind-inria/mri-nufft/network/dependents

Integration with other libraries

- $\geq$ 20 Projects depends on MRI-NUFFT[1]
- Integration with external packages
  - (Py)Pulseq
  - DeepInverse

```python
from mrinufft.trajectories import (pulseq_gre,
    read_pulseq_traj)
traj = ...
seq = pulseq_gre(traj, acq=acq)
seq.write("spiral_gre.seq")
traj_read = read_pulseq_traj(seq)
```



Example for integration with deepinv

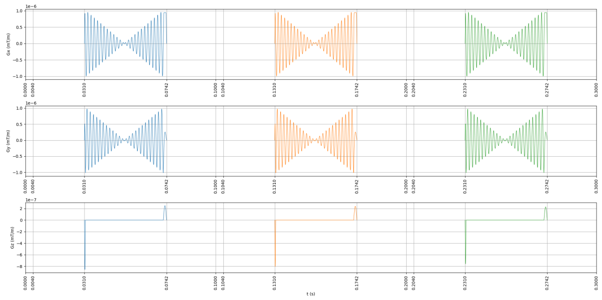
```python
from mrinufft import get_operator
from deepinv.optim import WaveletPrior, L2, optim_builder
# MRI-NUFFT setup
fourier_op = get_operator("cufinufft",samples_loc,shape=shape)
y = fourier_op.op(ground_truth) + noise  # y = Ax + ε
```

[1] https://github.com/mind-inria/mri-nufft/network/dependents

Integration with other libraries

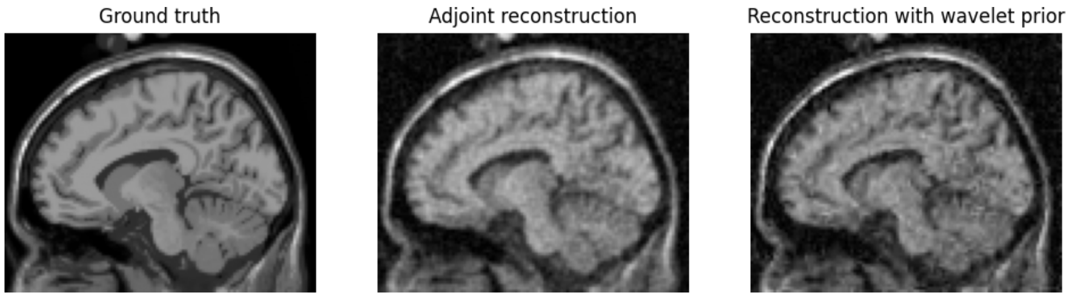
- $\geq$ 20 Projects depends on MRI-NUFFT[1]
- Integration with external packages
  - (Py)Pulseq
  - DeepInverse

```python
from mrinufft.trajectories import (pulseq_gre,
    read_pulseq_traj)
traj = ...
seq = pulseq_gre(traj, acq=acq)
seq.write("spiral_gre.seq")
traj_read = read_pulseq_traj(seq)
```



Example for integration with deepinv

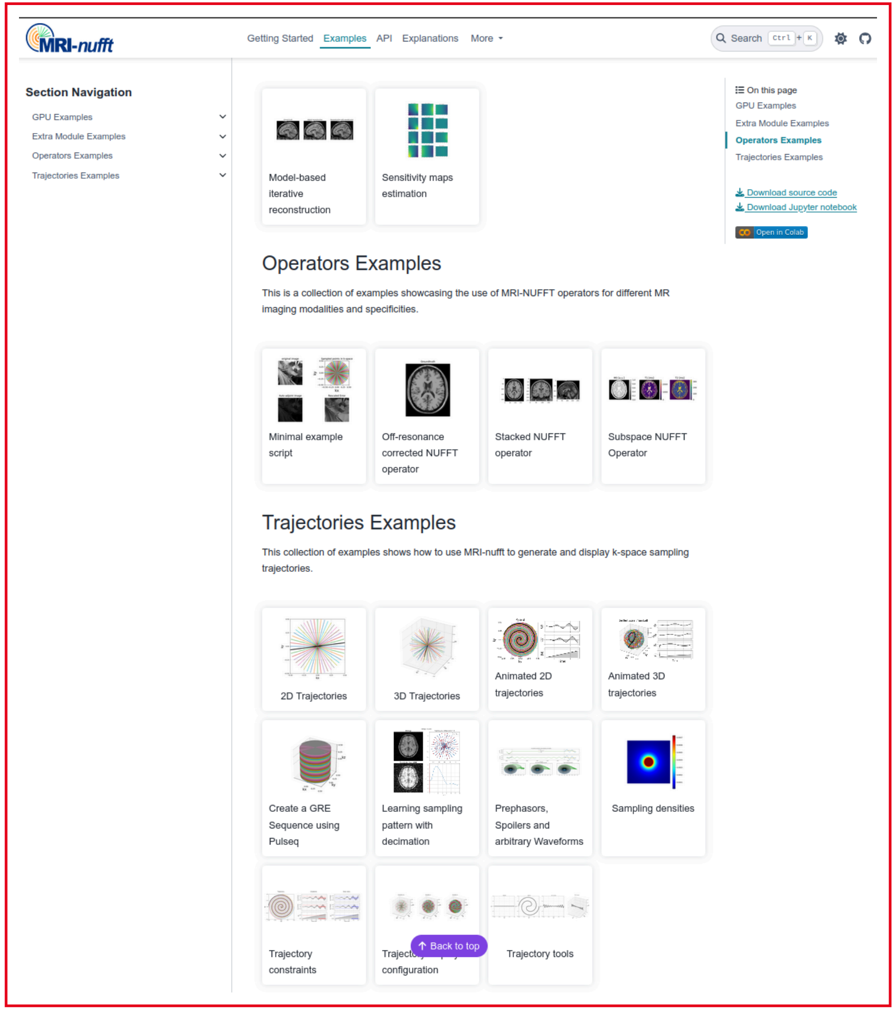
```python
from mrinufft import get_operator
from deepinv.optim import WaveletPrior, L2, optim_builder
# MRI-NUFFT setup
fourier_op = get_operator("cufinufft",samples_loc,shape=shape)
y = fourier_op.op(ground_truth) + noise  # y = Ax + ε

# DeepInverse setup
physics = fourier_op.make_deepinv_phy()
wavelet = WaveletPrior(wv="sym8", is_complex=True, p=1) # ‖Ψ·‖_1
wavelet_fista = optim_builder(
    iteration="FISTA",
    prior=wavelet, # swap with ANY prior !
    data_fidelity=L2(),
    params_algo={"lambda": 0.1}
)

# Reconstruction
x_pinv   = physics.A_dagger(y) # argmin_x‖ Ax − y ‖_2^2
x_wavelet = wavelet_fista(y, physics) # argmin_x‖ Ax − y ‖_2^2 + λ‖Ψx‖_1^1
```



[1] https://github.com/mind-inria/mri-nufft/network/dependents

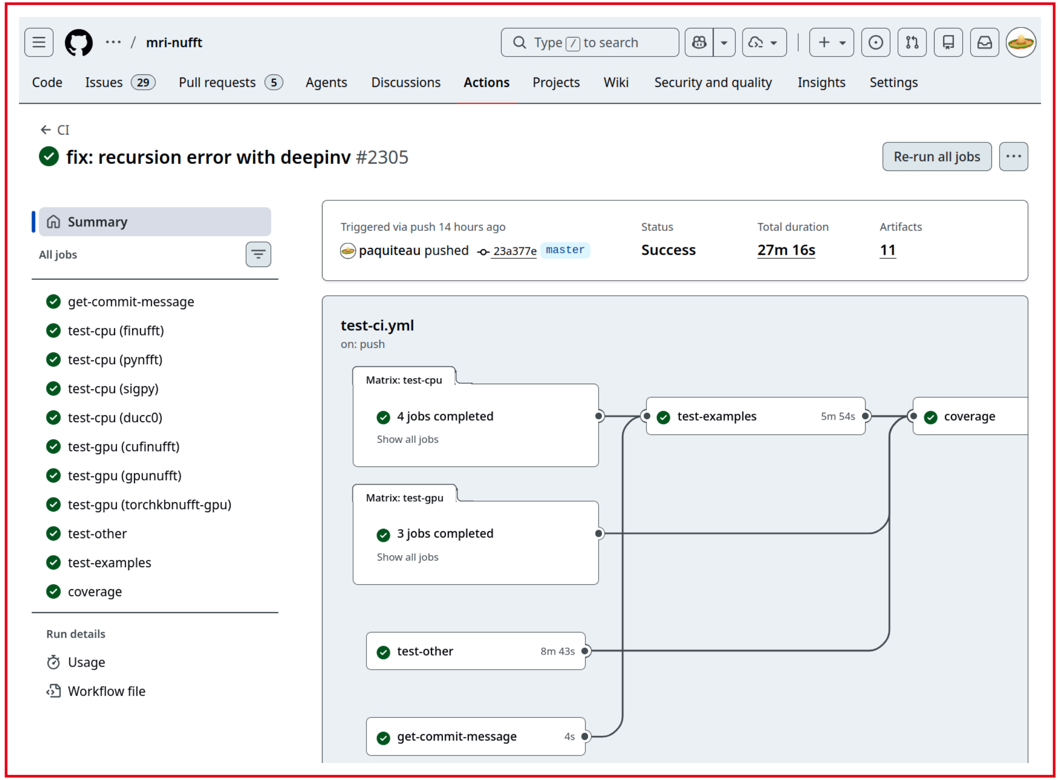
Packaging an Open Source Software

- 📖 Documentation
  - No documentation, no utilization
- ❓ Examples, tutorials
  - Onboarding new-comers Open in Colab



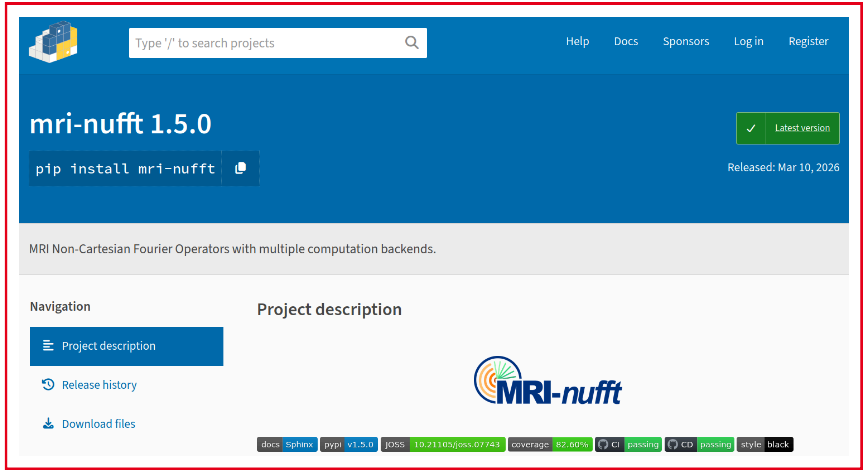
Packaging an Open Source Software

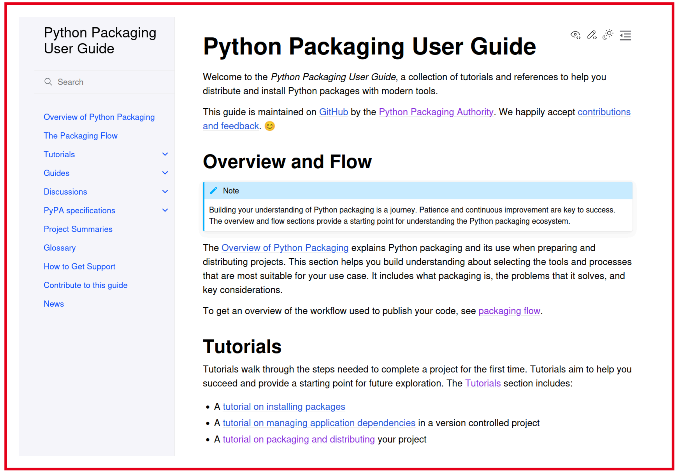
- 📖 Documentation
  - No documentation, no utilization
- ❓ Examples, tutorials
  - Onboarding new-comers Open in Colab
- 🧪 Unit Tests
  - Catch bug if you can
- 🪄 Continuous integration and quality linters



Packaging an Open Source Software

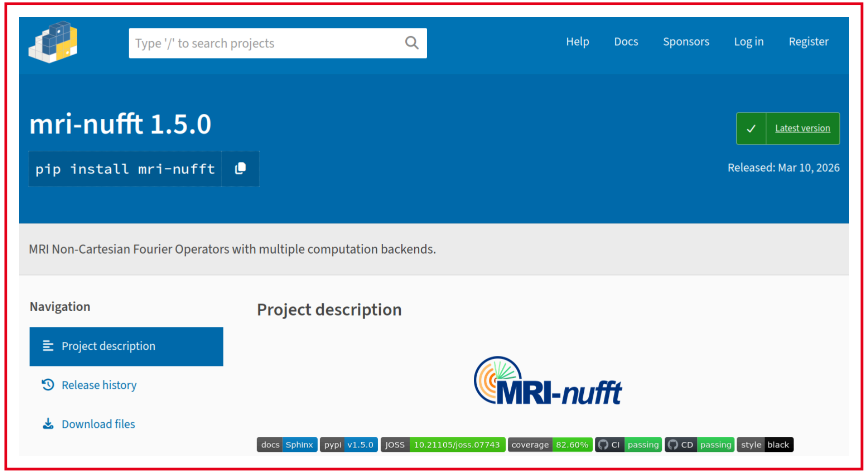
- 📖 Documentation
  - No documentation, no utilization
- ❓ Examples, tutorials
  - Onboarding new-comers Open in Colab
- 🧪 Unit Tests
  - Catch bug if you can
- 🪄 Continuous integration and quality linters
  - `ruff`, `black`, `ty`, `pyright`, `pyrefly`, …
- 🎁 Python Package:
  - `pip install mri-nufft`

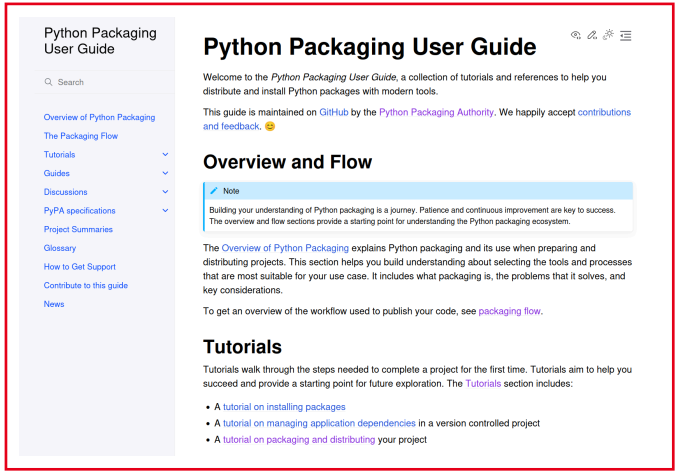




Packaging an Open Source Software

- Documentation
  - No documentation, no utilization
- Examples, tutorials
  - Onboarding new-comers Open in Colab
- Unit Tests
  - Catch bug if you can
- Continuous integration and quality linters
  - `ruff`, `black`, `ty`, `pyright`, `pyrefly`, ...
- Python Package:
  - `pip install mri-nufft`
- Collaborators (and stickers !)





3. Beyond Toolboxes and Packages

Open data for Reproducibility

Dataset for training and validation of reconstruction methods

FastMRI

- Brain, Knee, Prostate, Breast
- stack of 2D images
- https://fastmri.med.nyu.edu/

Open data for Reproducibility

Dataset for training and validation of reconstruction methods

FastMRI

- Brain, Knee, Prostate, Breast
- stack of 2D images
- https://fastmri.med.nyu.edu/

Calgary-Campinas

- T1w Brain images
- Stack of 2D
- https://www.ccdataset.com/

Open data for Reproducibility

Dataset for training and validation of reconstruction methods

FastMRI

- Brain, Knee, Prostate, Breast
- stack of 2D images
- https://fastmri.med.nyu.edu/

Calgary-Campinas

- T1w Brain images
- Stack of 2D
- https://www.ccdataset.com/

CMRxRecon

- Cardiac Imaging 2D+T, 3D+T
- https://cmrxrecon.github.io/2025/Home.html

Open data for Reproducibility

Dataset for training and validation of reconstruction methods

FastMRI

- Brain, Knee, Prostate, Breast
- stack of 2D images
- https://fastmri.med.nyu.edu/

Calgary-Campinas

- T1w Brain images
- Stack of 2D
- https://www.ccdataset.com/

CMRxRecon

- Cardiac Imaging 2D+T, 3D+T
- https://cmrxrecon.github.io/2025/Home.html

Open Format and API

- BIDS: Dataset file structure for MRI (and Neuro Imaging)
  - No k-space standard (yet ?)
- ISMRMRD: Standard raw k-space data format (Instead of `.dat`, `.raw`, “P-files”, …)

Open data for Reproducibility

Dataset for training and validation of reconstruction methods

FastMRI

- Brain, Knee, Prostate, Breast
- stack of 2D images
- https://fastmri.med.nyu.edu/

Calgary-Campinas

- T1w Brain images
- Stack of 2D
- https://www.ccdataset.com/

CMRxRecon

- Cardiac Imaging 2D+T, 3D+T
- https://cmrxrecon.github.io/2025/Home.html

Open Format and API

- BIDS: Dataset file structure for MRI (and Neuro Imaging)
  - No k-space standard (yet ?)
- ISMRMRD: Standard raw k-space data format (Instead of `.dat`, `.raw`, “P-files”, …)
- Gadgetron: Deploy Reconstruction at the Scanner
  - See Also: OpenRecon (Siemens), Tyger (Microsoft)

Open data for Reproducibility

Dataset for training and validation of reconstruction methods

FastMRI

- Brain, Knee, Prostate, Breast
- stack of 2D images
- https://fastmri.med.nyu.edu/

Calgary-Campinas

- T1w Brain images
- Stack of 2D
- https://www.ccdataset.com/

CMRxRecon

- Cardiac Imaging 2D+T, 3D+T
- https://cmrxrecon.github.io/2025/Home.html

Open Format and API

- BIDS: Dataset file structure for MRI (and Neuro Imaging)
  - No k-space standard (yet ?)
- ISMRMRD: Standard raw k-space data format (Instead of `.dat`, `.raw`, “P-files”, …)
- Gadgetron: Deploy Reconstruction at the Scanner
  - See Also: OpenRecon (Siemens), Tyger (Microsoft)

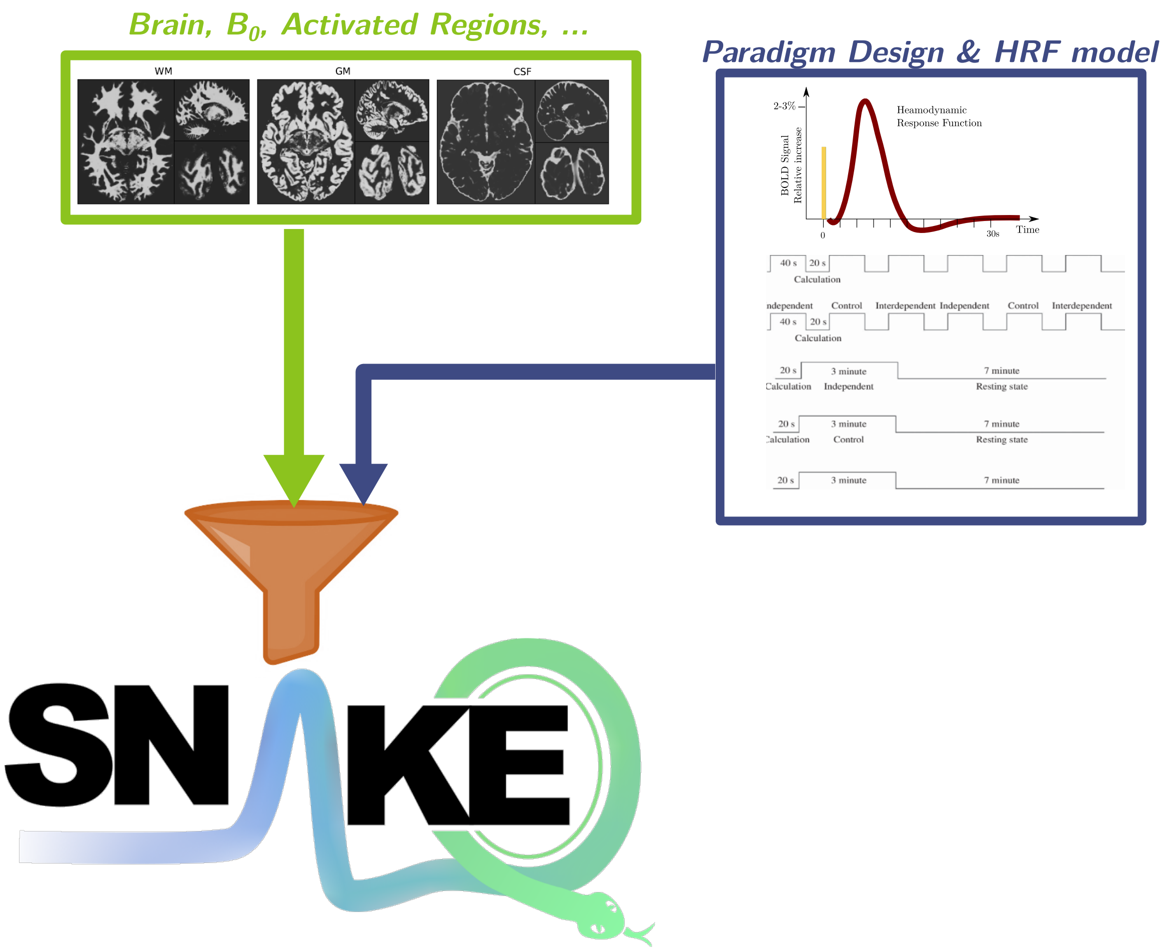
No dataset or Ground-Truth available ?

fMRI, space/time resolution tradeoff

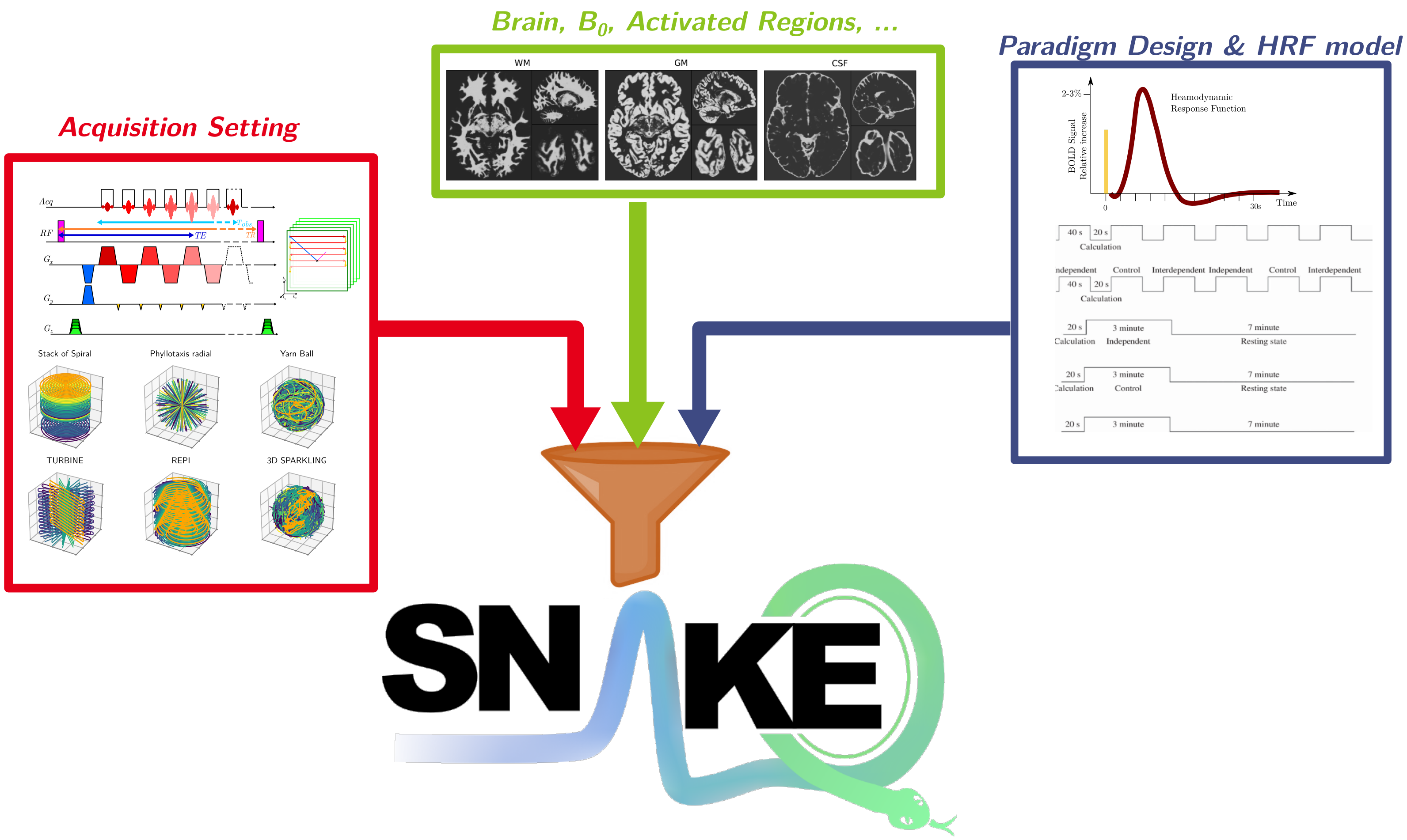
SNAKE fMRI k-space Simulator

Brain, B_0, Activated Regions, ...

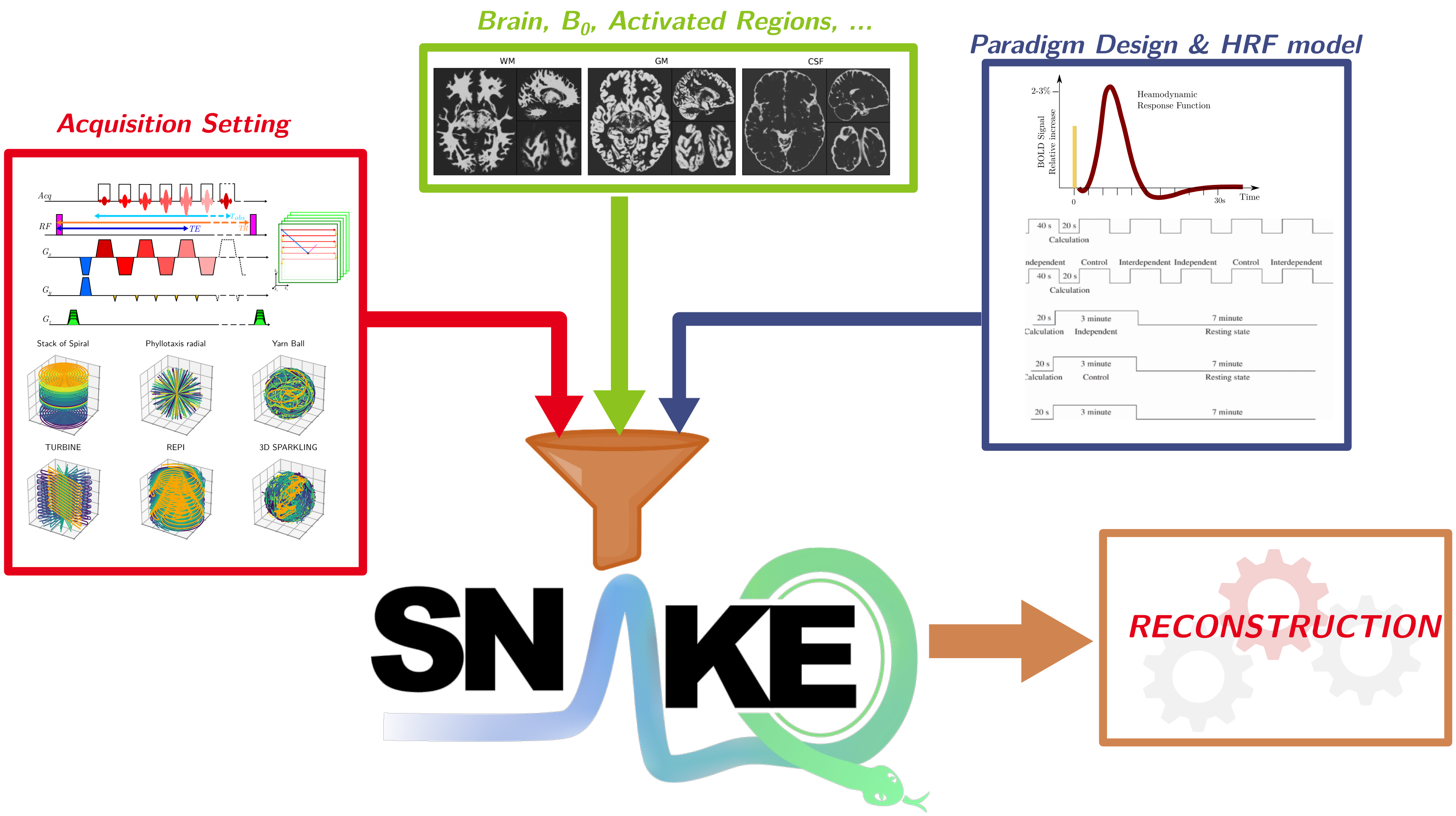
SNAKE fMRI k-space Simulator



SNAKE fMRI k-space Simulator



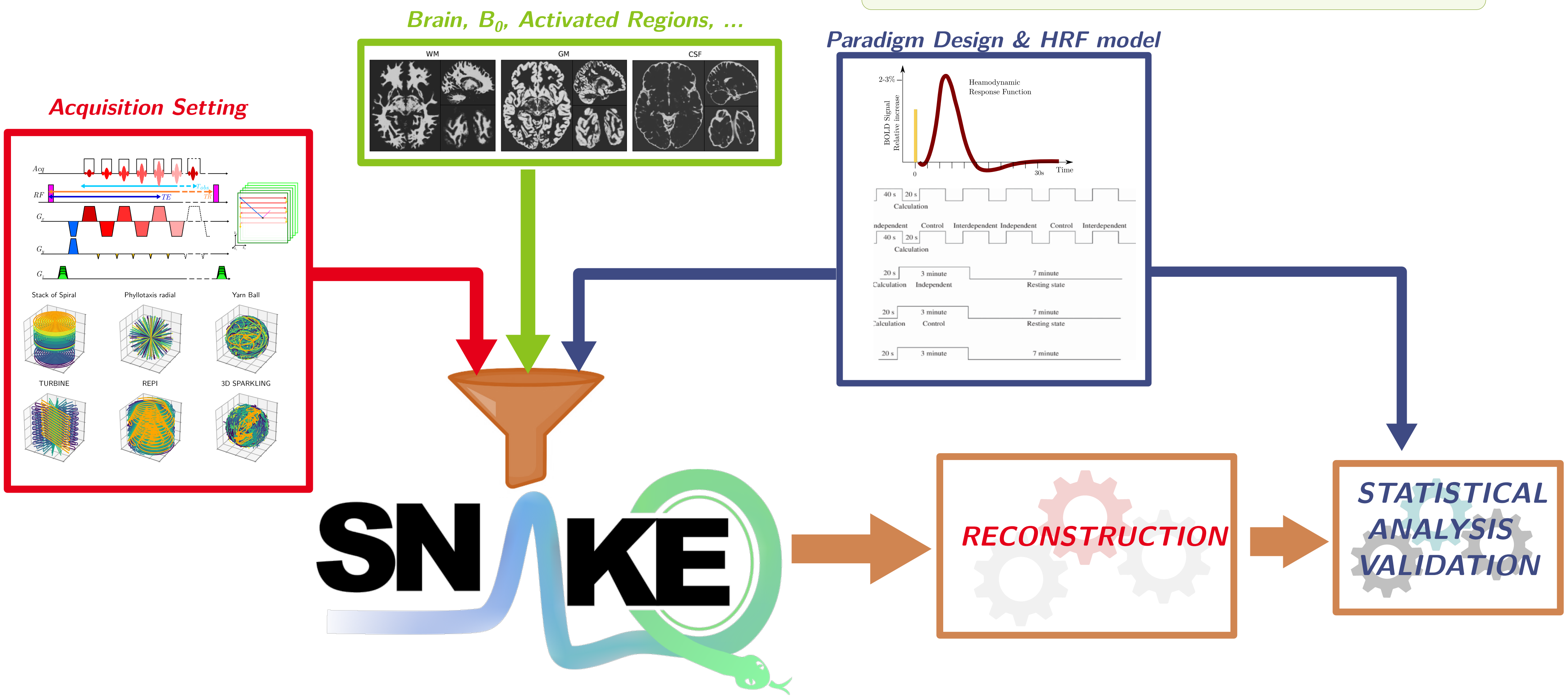
SNAKE fMRI k-space Simulator



SNAKE fMRI k-space Simulator

Learn more

Comby et al., Imaging Neuroscience, 2025.



A word about benchmarks

MRM Editorial

MAGNETIC RESONANCE IN MEDICINE | ISMRM | ONE COMMUNITY FOR CLINICIANS AND SCIENTISTS

EDITORIAL | Free Access

The “State of the Art” in MR Image Reconstruction? Knowledge, Culture, and What We Leave Behind in An Era of Big Data and Machine Learning

Justin P. Haldar

First published: 10 April 2026 | https://doi.org/10.1002/mrm.70377 | VIEW METRICS

- Is PSNR and SSIM really what we should optimize for ?
- Leaderboards vs clinical impact

A word about benchmarks

MRM Editorial

MAGNETIC RESONANCE IN MEDICINE | ISMRM | ONE COMMUNITY FOR CLINICIANS AND SCIENTISTS

EDITORIAL | Free Access

The "State of the Art" in MR Image Reconstruction? Knowledge, Culture, and What We Leave Behind in An Era of Big Data and Machine Learning

Justin P. Haldar

First published: 10 April 2026 | https://doi.org/10.1002/mrm.70377 | VIEW METRICS

- Is PSNR and SSIM really what we should optimize for ?
- Leaderboards vs clinical impact

State of the art $\neq$ State of the practice

- Make the toolbox usable by non-expert of the k-space.

Community initiatives

- 🏜️ Workshop on MRI Acquisition and Reconstruction in Sedona, AZ.

Community initiatives

- 🏜️ Workshop on MRI Acquisition and Reconstruction in Sedona, AZ.
- 🔬 Reproducible research group
  - MRM Code Review
  - Join the mailing list !

Group Meeting

Sunday 9:45-11:45
Ballroom East

Community initiatives

- 🏜️ Workshop on MRI Acquisition and Reconstruction in Sedona, AZ.
- 🔬 Reproducible research group
  - MRM Code Review
  - Join the mailing list !
- 🌐 Websites
  - https://ismrm.github.io/mrhub/
  - https://www.opensourceimaging.org/projects

Group Meeting

Sunday 9:45-11:45
Ballroom East

Community initiatives

- 🏜️ Workshop on MRI Acquisition and Reconstruction in Sedona, AZ.
- 🔬 Reproducible research group
  - MRM Code Review
  - Join the mailing list !
- 🌐 Websites
  - https://ismrm.github.io/mrhub/
  - https://www.opensourceimaging.org/projects
- 📄 Commits to h-index convertors:
  - Journal of Open-Source Software (JOSS)
  - Zenodo
  - MRM Tooolbox Articles (be the first ?)

Group Meeting

Sunday 9:45-11:45
Ballroom East

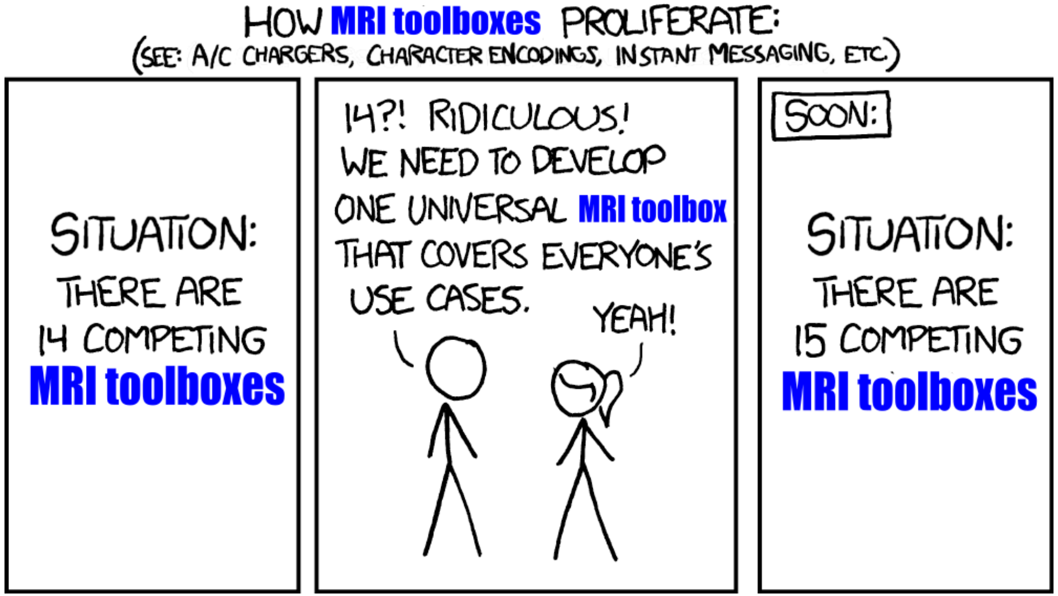
Take home messages

- MRI Reconstruction is a challenging but *fun* problem
  - Interface between Math, Physics and Computer Science

[1] MRI-NUFFT + DeepInv goes a long way: https://mind-inria.github.io/mri-nufft/generated/autoexamples/extras/example_recon.html

Take home messages

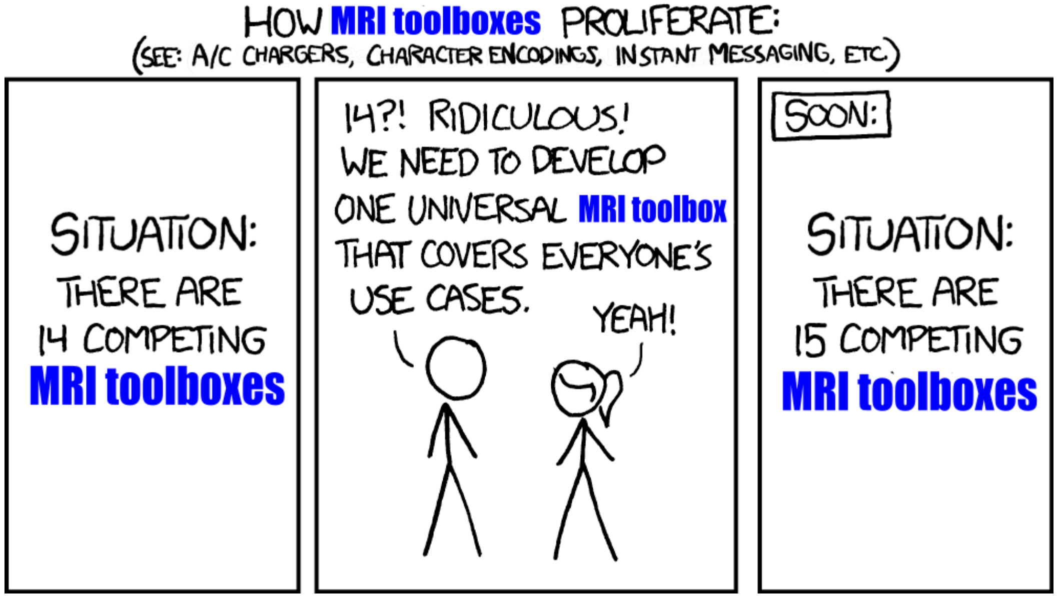
- MRI Reconstruction is a challenging but *fun* problem
  - Interface between Math, Physics and Computer Science
- It's okay to reinvent the wheel, but most of it is on-the-shelf[1]



[1] MRI-NUFFT + DeepInv goes a long way: https://mind-inria.github.io/mri-nufft/generated/autoexamples/extras/example_recon.html

Take home messages

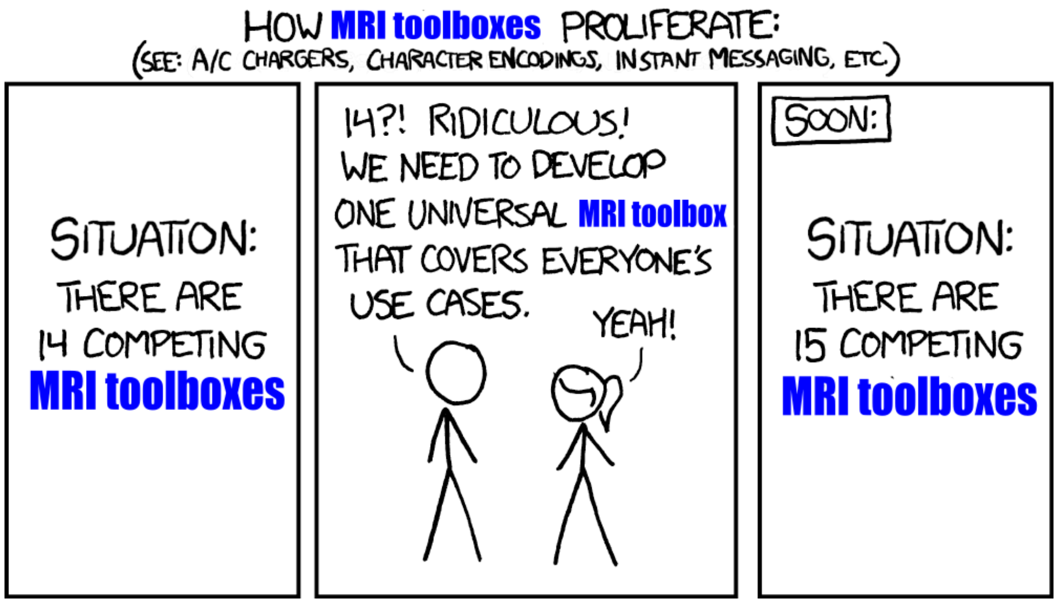
- MRI Reconstruction is a challenging but *fun* problem
  - Interface between Math, Physics and Computer Science
- It's okay to reinvent the wheel, but most of it is on-the-shelf[1]
- Provide code and figures plotting script with paper !
  - don't forget the instructions to run it.
  - NO HARDCODED ABSOLUTE FILEPATH!



[1] MRI-NUFFT + DeepInv goes a long way: https://mind-inria.github.io/mri-nufft/generated/autoexamples/extras/example_recon.html

Take home messages

- MRI Reconstruction is a challenging but *fun* problem
  - Interface between Math, Physics and Computer Science
- It's okay to reinvent the wheel, but most of it is on-the-shelf[1]
- Provide code and figures plotting script with paper !
  - don't forget the instructions to run it.
  - NO HARDCODED ABSOLUTE FILEPATH!
- Contributions are *always* welcomes



[1] MRI-NUFFT + DeepInv goes a long way: https://mind-inria.github.io/mri-nufft/generated/autoexamples/extras/example_recon.html

Check out Our *awesome* **open-source** software:

MRI-nufft

SNAKE

Thank you for your attention

pierre-antoine.comby@cea.fr

Neurospin, CEA, Paris-Saclay University