cea joliot

Digital Poster Presentation
Wednesday 8th 09:15
Hall 403 - Computer 17

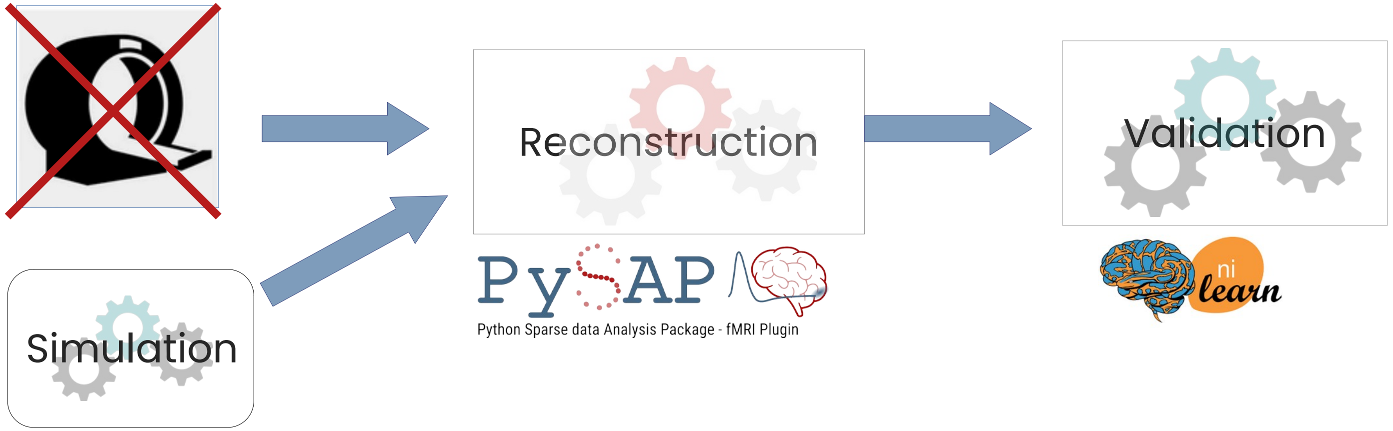
3408 SNAKE-fMRI: A Simulator from Neurovascular coupling to Acquisition of K-space data for Exploration of fMRI

Pierre-Antoine Comby
Supervisors: Philippe Ciuciu & Alexandre Vignaud

SNAKE-fMRI

NeuroSpin METRIC MIND Inria université PARIS-SACLAY

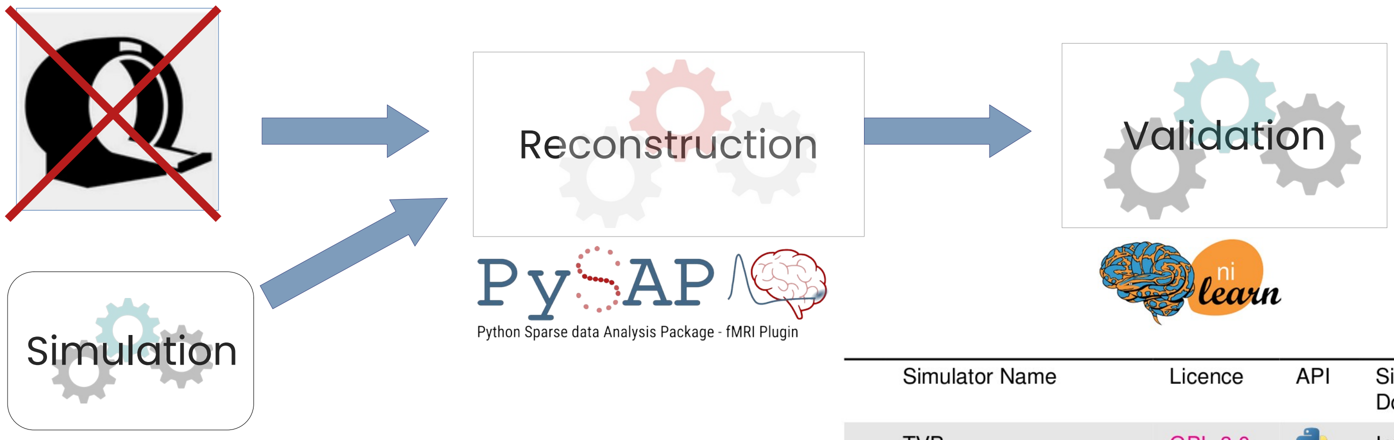
A Solution to the Reproducibility Crisis for high-res fMRI



- Push for High Res. fMRI (Space & Time)
- Develop and compare new Acquisition/Reconstruction Methods
  - Esp. for 3D Non-Cartesian Setup
- Reproducibility Issues
  - Ensure control of *all* inputs (Brain included)

↳ *Simulation setup needed*

A Solution to the Reproducibility Crisis for high-res fMRI



Reconstruction

Validation

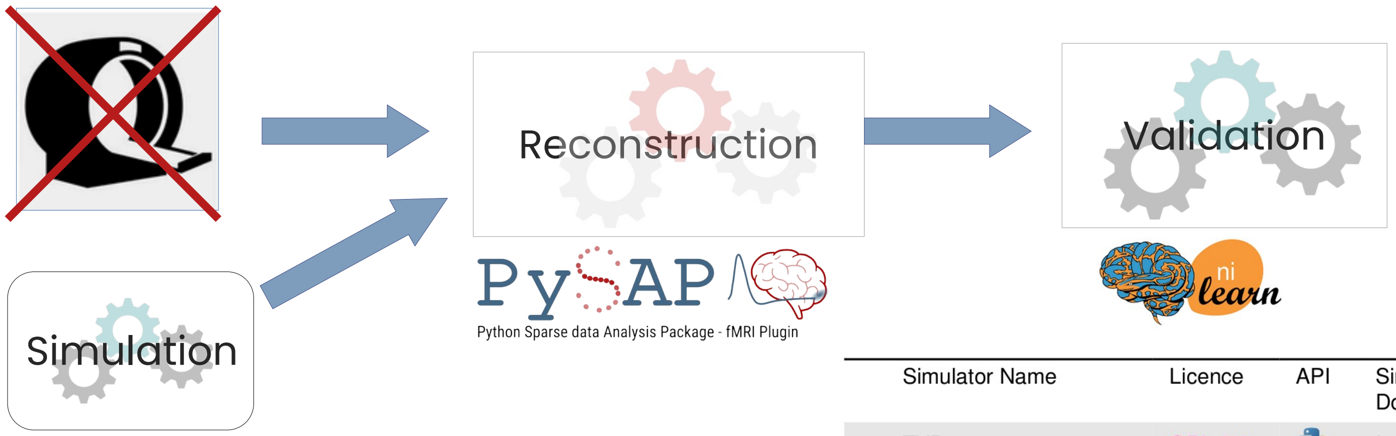
Simulation

- Push for High Res. fMRI (Space & Time)
- Develop and compare new Acquisition/Reconstruction Methods
  - Esp. for 3D Non-Cartesian Setup
- Reproducibility Issues
  - Ensure control of *all* inputs (Brain included)

↳ *Simulation setup needed*

	Simulator Name	Licence	API	Sim. Domain	Required External Data	Interface	Reconstr.
	TVB (Sanz Leon et al., 2013)	GPL-3.0	🐍	Image		GUI/script	N/A
MRI Simulator	Jemris (Stöcker et al., 2010)	GPL-2.0	▲ C++	Bloch		GUI	ISMRMD raw data
	ODIN (Jochimsen et al., 2006)	GPL-2.0	C++	Bloch	Tissue Maps, Sequence	c++/GUI	FFT
	MRILab (Liu et al., 2017)	BSD-2	▲ C++	Bloch	Preset Macros	GUI	FFT Non-Cartesian
	Bloch-Solver (Kose & Kose, 2017)	Proprietary	🐍	Bloch	Tissue Maps,	script	FFT
fMRI Simulator	POSSUM (Drobnjak et al., 2006)	FSL	▲ C++	Bloch	Tissue Maps Sequence, Events	CLI	FFT
	Neurolib (Cakan et al., 2023)	MIT	🐍	Image	Connectivity Matrices	script	N/A
	SimTB (Erhardt et al., 2012)	Open Source	▲	Image	Spatial Maps, Events	GUI	N/A
	NeuroRSim (Welvaert et al., 2011)	GPL-2.0	R	Image		script	N/A
	fmriSim (Ellis et al., 2020)	Apache-2.0	🐍	Image		script	N/A
	SNAKE-fMRI	*MIT*	🐍	*Kspace Image*	*Configuration files*	*script/CLI*	*Any (4D methods)*

A Solution to the Reproducibility Crisis for high-res fMRI



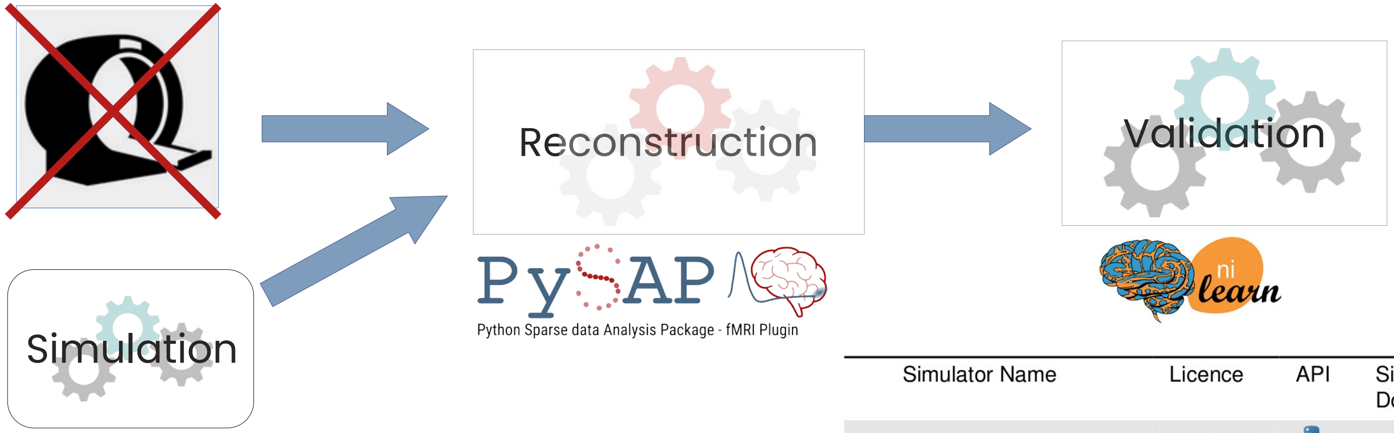
- Push for High Res. fMRI (Space & Time)
- Develop and compare new Acquisition/Reconstruction Methods
  - Esp. for 3D Non-Cartesian Setup
- Reproducibility Issues
  - Ensure control of *all* inputs (Brain included)

↳ *Simulation setup needed*

	Simulator Name	Licence	API	Sim. Domain	Required External Data	Interface	Reconstr.
	TVB (Sanz Leon et al., 2013)	GPL-3.0	🐍	Image		GUI/script	N/A
MRI Simulator	Jemris (Stöcker et al., 2010)	GPL-2.0	▲ C++	Bloch		GUI	ISMRMD raw data
	ODIN (Jochimsen et al., 2006)	GPL-2.0	C++	Bloch	Tissue Maps, Sequence	c++/GUI	FFT
	MRILab (Liu et al., 2017)	BSD-2	▲ C++	Bloch	Preset Macros	GUI	FFT Non-Cartesian
	Bloch-Solver (Kose & Kose, 2017)	Proprietary	🐍	Bloch	Tissue Maps,	script	FFT
fMRI Simulator	POSSUM (Drobnjak et al., 2006)	FSL	▲ C++	Bloch	Tissue Maps Sequence, Events	CLI	FFT
	Neurolib (Cakan et al., 2023)	MIT	🐍	Image	Connectivity Matrices	script	N/A
	SimTB (Erhardt et al., 2012)	Open Source	▲	Image	Spatial Maps, Events	GUI	N/A
	NeuroRSim (Welvaert et al., 2011)	GPL-2.0	R	Image		script	N/A
	fmriSim (Ellis et al., 2020)	Apache-2.0	🐍	Image		script	N/A
	SNAKE-fMRI	*MIT*	🐍	*Kspace Image*	*Configuration files*	*script/CLI*	*Any (4D methods)*

for Sequence Programming

A Solution to the Reproducibility Crisis for high-res fMRI



- Push for High Res. fMRI (Space & Time)
- Develop and compare new Acquisition/Reconstruction Methods
  - Esp. for 3D Non-Cartesian Setup
- Reproducibility Issues
  - Ensure control of *all* inputs (Brain included)

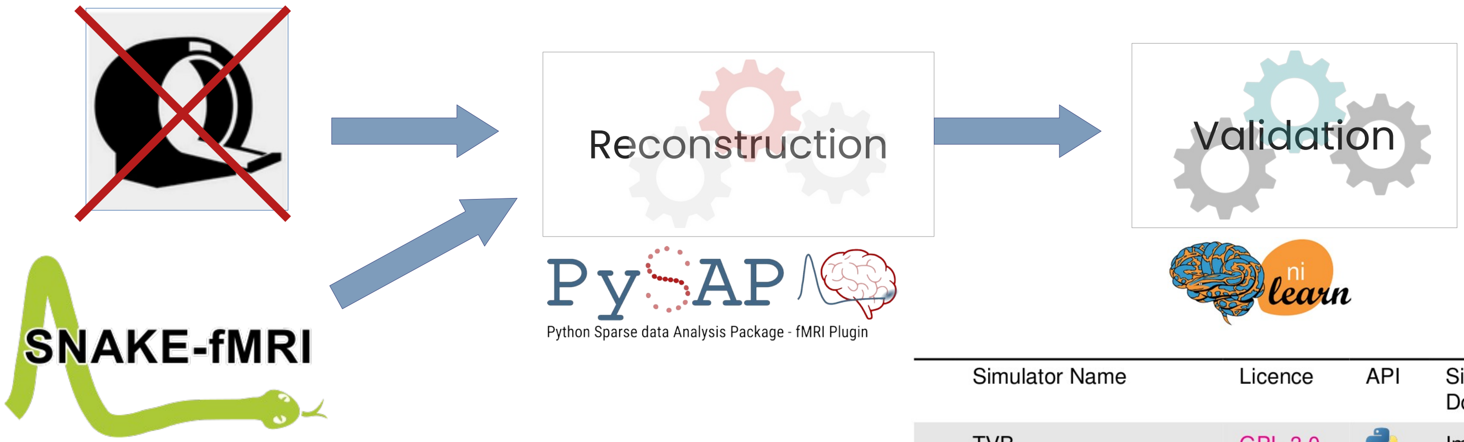
↳ *Simulation setup needed*

	Simulator Name	Licence	API	Sim. Domain	Required External Data	Interface	Reconstr.
	TVB (Sanz Leon et al., 2013)	GPL-3.0	🐍	Image		GUI/script	N/A
MRI Simulator	Jemris (Stöcker et al., 2010)	GPL-2.0	▲ C++	Bloch		GUI	ISMRMD raw data
	ODIN (Jochimsen et al., 2006)	GPL-2.0	C++	Bloch	Tissue Maps, Sequence	c++/GUI	FFT
	MRILab (Liu et al., 2017)	BSD-2	▲ C++	Bloch	Preset Macros	GUI	FFT Non-Cartesian
	Bloch-Solver (Kose & Kose, 2017)	Proprietary	🐍	Bloch	Tissue Maps,	script	FFT
fMRI Simulator	POSSUM (Drobnjak et al., 2006)	FSL	▲ C++	Bloch	Tissue Maps Sequence, Events	CLI	FFT
	Neurolib (Cakan et al., 2023)	MIT	🐍	Image	Connectivity Matrices	script	N/A
	SimTB (Erhardt et al., 2012)	Open Source	▲	Image	Spatial Maps, Events	GUI	N/A
	NeuroRSim (Welvaert et al., 2011)	GPL-2.0	R	Image		script	N/A
	fmriSim (Ellis et al., 2020)	Apache-2.0	🐍	Image		script	N/A
	SNAKE-fMRI	*MIT*	🐍	*Kspace Image*	*Configuration files*	*script/CLI*	*Any (4D methods)*

for Sequence Programming

only Image Domain

A Solution to the Reproducibility Crisis for high-res fMRI



- Push for High Res. fMRI (Space & Time)
- Develop and compare new Acquisition/Reconstruction Methods
  - Esp. for 3D Non-Cartesian Setup
- Reproducibility Issues
  - Ensure control of *all* inputs (Brain included)

↳ *Simulation setup needed*

	Simulator Name	Licence	API	Sim. Domain	Required External Data	Interface	Reconstr.
	TVB (Sanz Leon et al., 2013)	GPL-3.0	Python	Image		GUI/script	N/A
MRI Simulator	Jemris (Stöcker et al., 2010)	GPL-2.0	Matlab C++	Bloch		GUI	ISMRMD raw data
	ODIN (Jochimsen et al., 2006)	GPL-2.0	C++	Bloch	Tissue Maps, Sequence	c++/GUI	FFT
	MRILab (Liu et al., 2017)	BSD-2	Matlab C++	Bloch	Preset Macros	GUI	FFT Non-Cartesian
	Bloch-Solver (Kose & Kose, 2017)	Proprietary	Python	Bloch	Tissue Maps,	script	FFT
fMRI Simulator	POSSUM (Drobnjak et al., 2006)	FSL	Matlab C++	Bloch	Tissue Maps Sequence, Events	CLI	FFT
	Neurolib (Cakan et al., 2023)	MIT	Python	Image	Connectivity Matrices	script	N/A
	SimTB (Erhardt et al., 2012)	Open Source	Matlab	Image	Spatial Maps, Events	GUI	N/A
	NeuroRSim (Welvaert et al., 2011)	GPL-2.0	R	Image		script	N/A
	fmriSim (Ellis et al., 2020)	Apache-2.0	Python	Image		script	N/A
	SNAKE-fMRI	*MIT*	Python	*Kspace Image*	*Configuration files*	*script/CLI*	*Any (4D methods)*

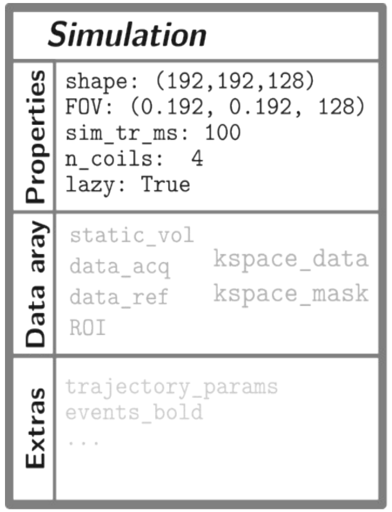
for Sequence Programming

only Image Domain

Ours

From simulated BOLD signals to K-Space

From simulated BOLD signals to K-Space



Simulation

Properties

```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```

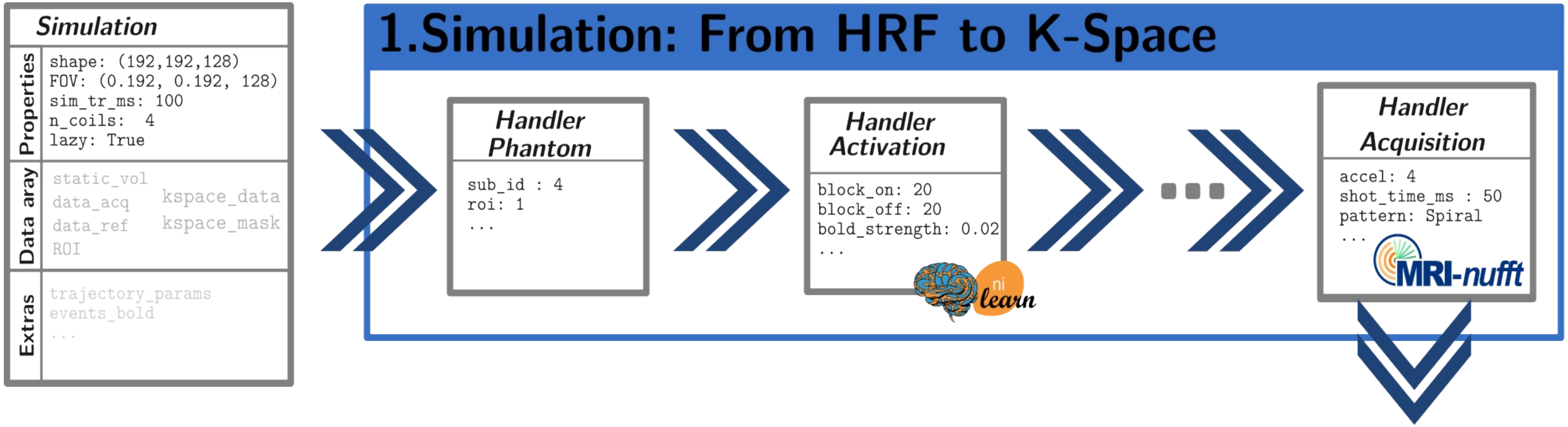
Data aray

```
static_vol
data_acq       kspace_data
data_ref       kspace_mask
ROI
```

Extras

```
trajectory_params
events_bold
...
```

From simulated BOLD signals to K-Space



Simulation

Properties

```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```

Data aray

```
static_vol
data_acq      kspace_data
data_ref      kspace_mask
ROI
```

Extras

```
trajectory_params
events_bold
...
```

1.Simulation: From HRF to K-Space

Handler Phantom

```
sub_id : 4
roi: 1
...
```

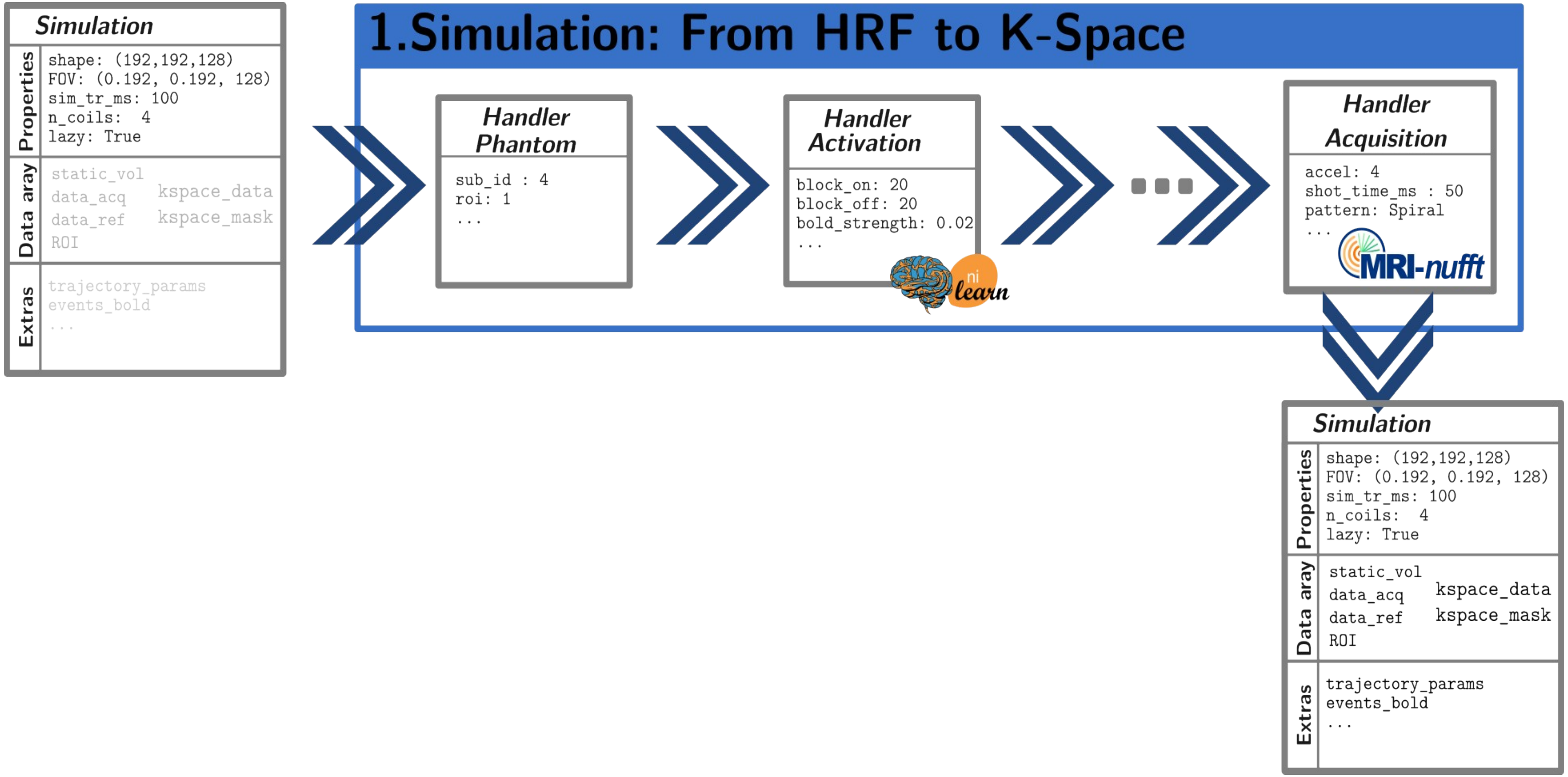
Handler Activation

```
block_on: 20
block_off: 20
bold_strength: 0.02
...
```

Handler Acquisition

```
accel: 4
shot_time_ms : 50
pattern: Spiral
...
```

From simulated BOLD signals to K-Space



Simulation

Properties

shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils: 4
lazy: True

Data aray

static_vol
data_acq
data_ref
ROI
kspace_data
kspace_mask

Extras

trajectory_params
events_bold
...

1.Simulation: From HRF to K-Space

Handler Phantom

sub_id : 4
roi: 1
...

Handler Activation

block_on: 20
block_off: 20
bold_strength: 0.02
...

Handler Acquisition

accel: 4
shot_time_ms : 50
pattern: Spiral
...

Simulation

Properties

shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils: 4
lazy: True

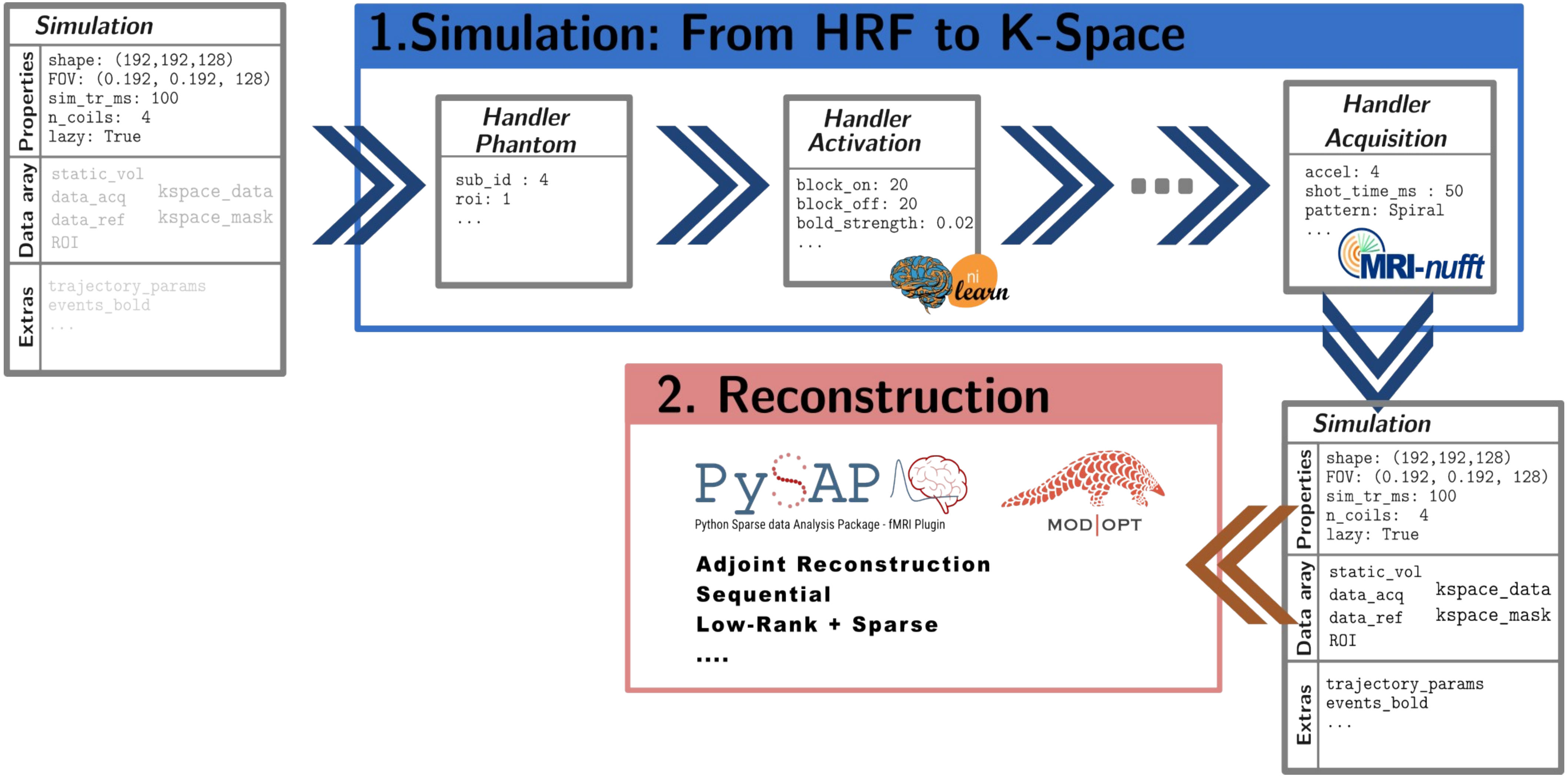
Data aray

static_vol
data_acq
data_ref
ROI
kspace_data
kspace_mask

Extras

trajectory_params
events_bold
...

From simulated BOLD signals to K-Space ... and back



Simulation

Properties

```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```

Data aray

```
static_vol
data_acq        kspace_data
data_ref        kspace_mask
ROI
```

Extras

```
trajectory_params
events_bold
...
```

1.Simulation: From HRF to K-Space

Handler Phantom

```
sub_id : 4
roi: 1
...
```

Handler Activation

```
block_on: 20
block_off: 20
bold_strength: 0.02
...
```

Handler Acquisition

```
accel: 4
shot_time_ms : 50
pattern: Spiral
...
```

Simulation

Properties

```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```

Data aray

```
static_vol
data_acq        kspace_data
data_ref        kspace_mask
ROI
```

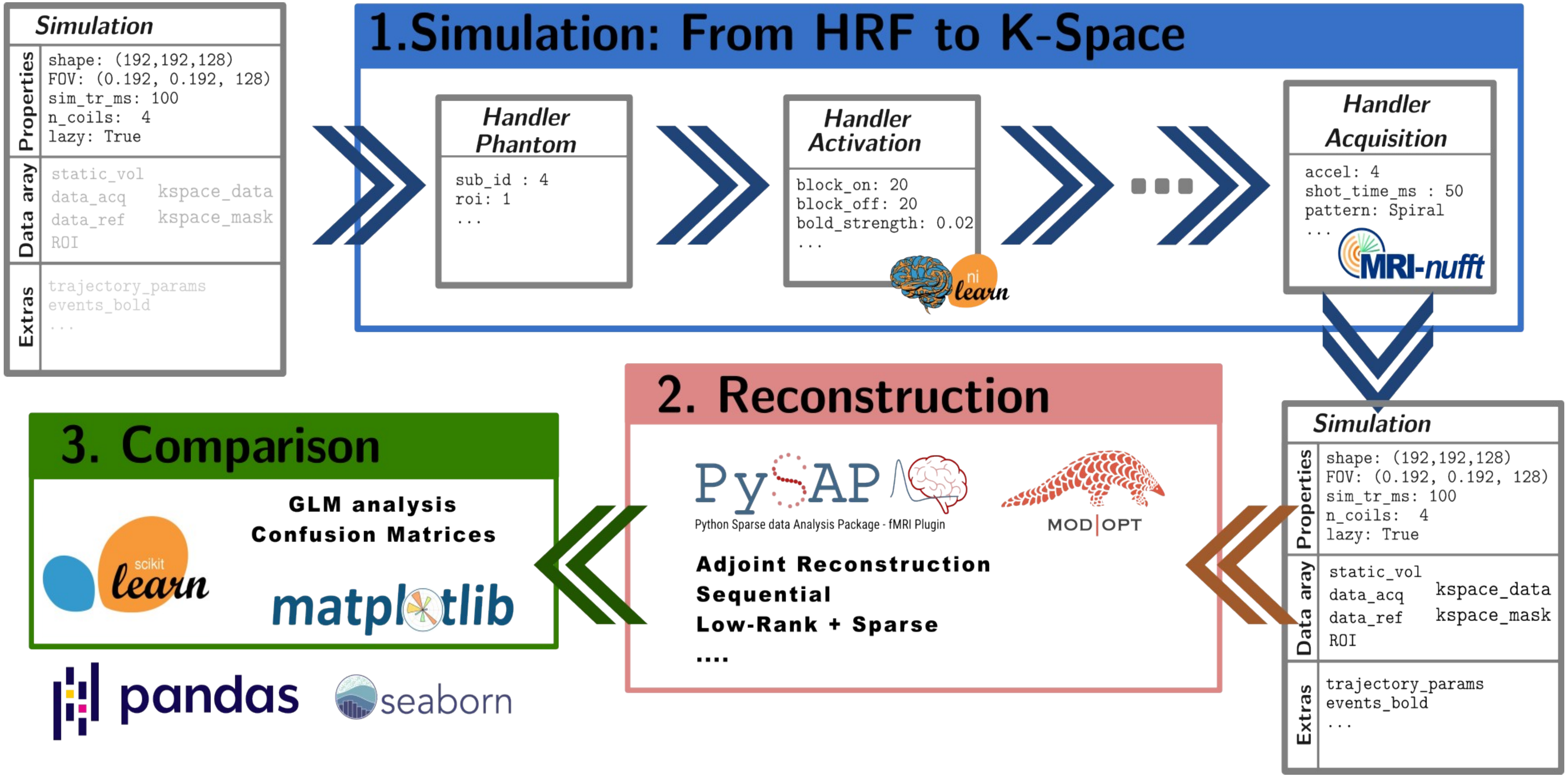
Extras

```
trajectory_params
events_bold
...
```

2. Reconstruction

Adjoint Reconstruction
Sequential
Low-Rank + Sparse
....

From simulated BOLD signals to K-Space ... and back



Simulation

Properties

```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```

Data aray

```
static_vol
data_acq      kspace_data
data_ref      kspace_mask
ROI
```

Extras

```
trajectory_params
events_bold
...
```

1.Simulation: From HRF to K-Space

Handler Phantom

```
sub_id : 4
roi: 1
...
```

Handler Activation

```
block_on: 20
block_off: 20
bold_strength: 0.02
...
```

Handler Acquisition

```
accel: 4
shot_time_ms : 50
pattern: Spiral
...
```

MRI-nufft

Simulation

Properties

```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```

Data aray

```
static_vol
data_acq      kspace_data
data_ref      kspace_mask
ROI
```

Extras

```
trajectory_params
events_bold
...
```

2. Reconstruction

PySAP

Python Sparse data Analysis Package - fMRI Plugin

MOD|OPT

Adjoint Reconstruction
Sequential
Low-Rank + Sparse
....

3. Comparison

GLM analysis
Confusion Matrices

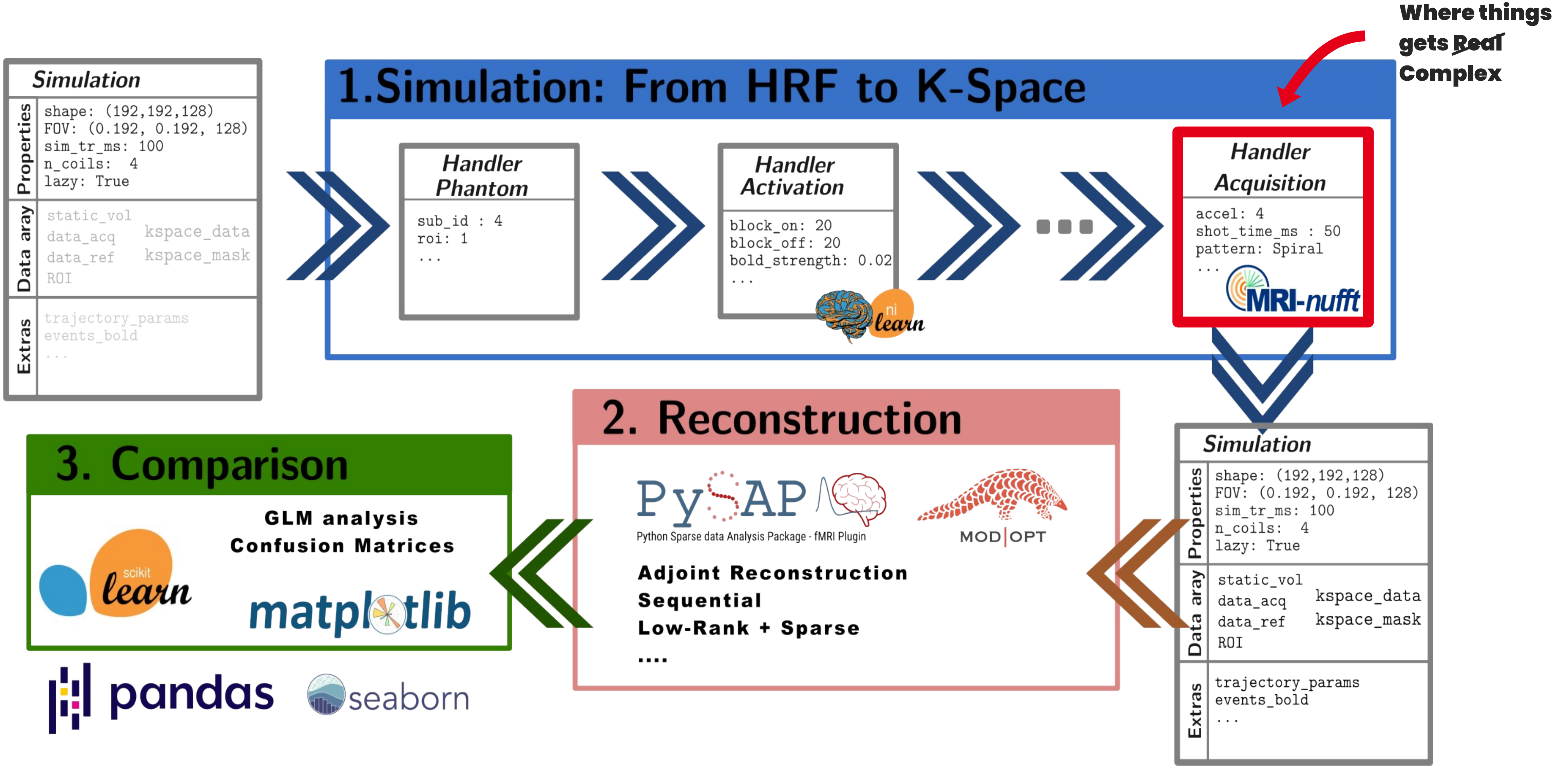
scikit learn

matplotlib

pandas

seaborn

From simulated BOLD signals to K-Space ... and back



Where things gets ~~Real~~ Complex

Simulation

Properties
```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```
Data aray
```
static_vol
data_acq      kspace_data
data_ref      kspace_mask
ROI
```
Extras
```
trajectory_params
events_bold
...
```

1.Simulation: From HRF to K-Space

Handler Phantom
```
sub_id : 4
roi: 1
...
```

Handler Activation
```
block_on: 20
block_off: 20
bold_strength: 0.02
...
```

Handler Acquisition
```
accel: 4
shot_time_ms : 50
pattern: Spiral
...
```

Simulation

Properties
```
shape: (192,192,128)
FOV: (0.192, 0.192, 128)
sim_tr_ms: 100
n_coils:  4
lazy: True
```
Data aray
```
static_vol
data_acq      kspace_data
data_ref      kspace_mask
ROI
```
Extras
```
trajectory_params
events_bold
...
```

2. Reconstruction

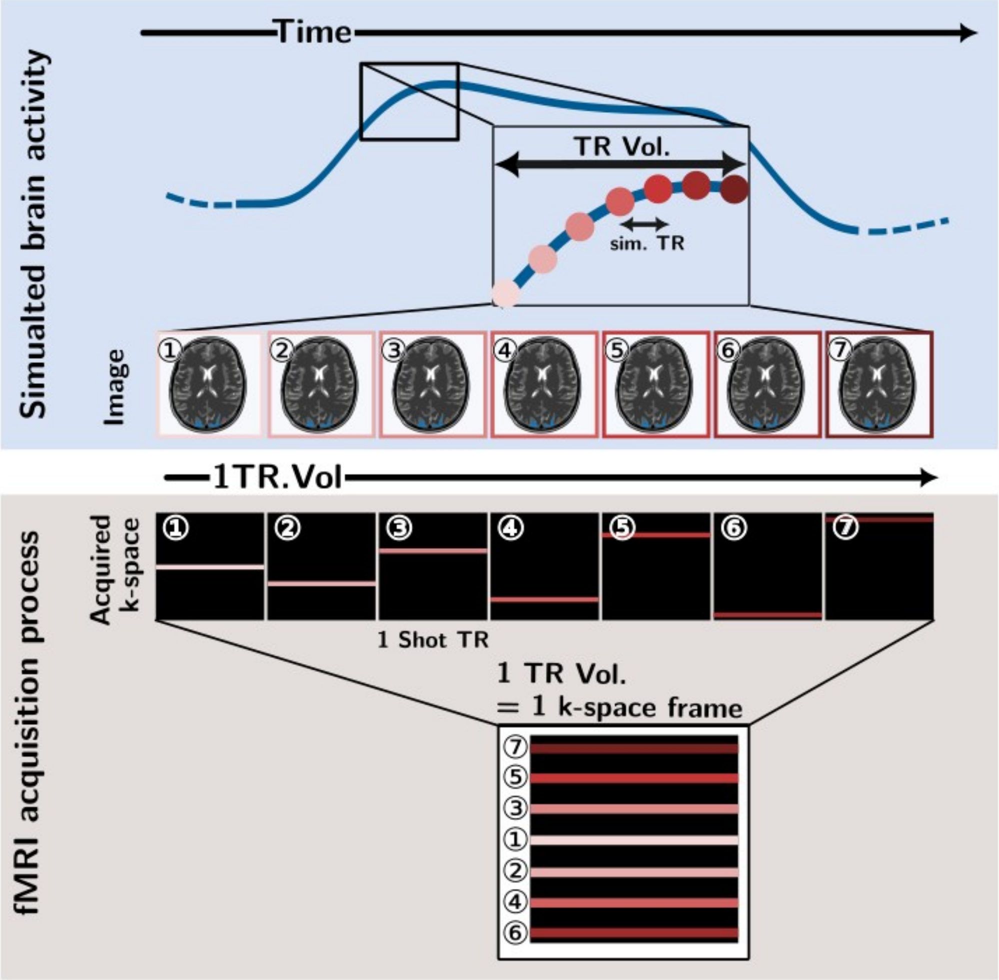
Adjoint Reconstruction
Sequential
Low-Rank + Sparse
....

3. Comparison

GLM analysis
Confusion Matrices

A focus on the Acquisition Module

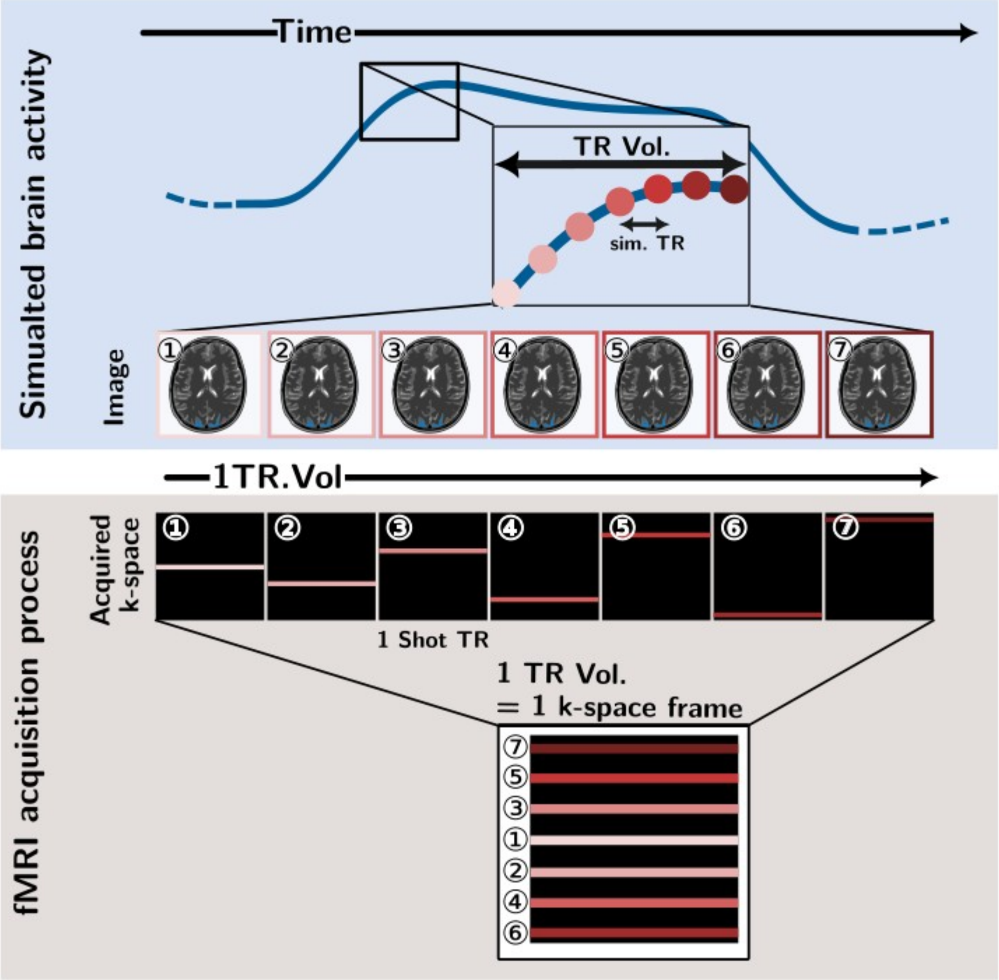
Principle of high temporal resolution for shot-wise acquisition



A focus on the Acquisition Module

- Goal: Sampling the continuous BOLD and Physiology signal at **high temporal resolution**
  - Flexibility for exploring new sampling strategies.

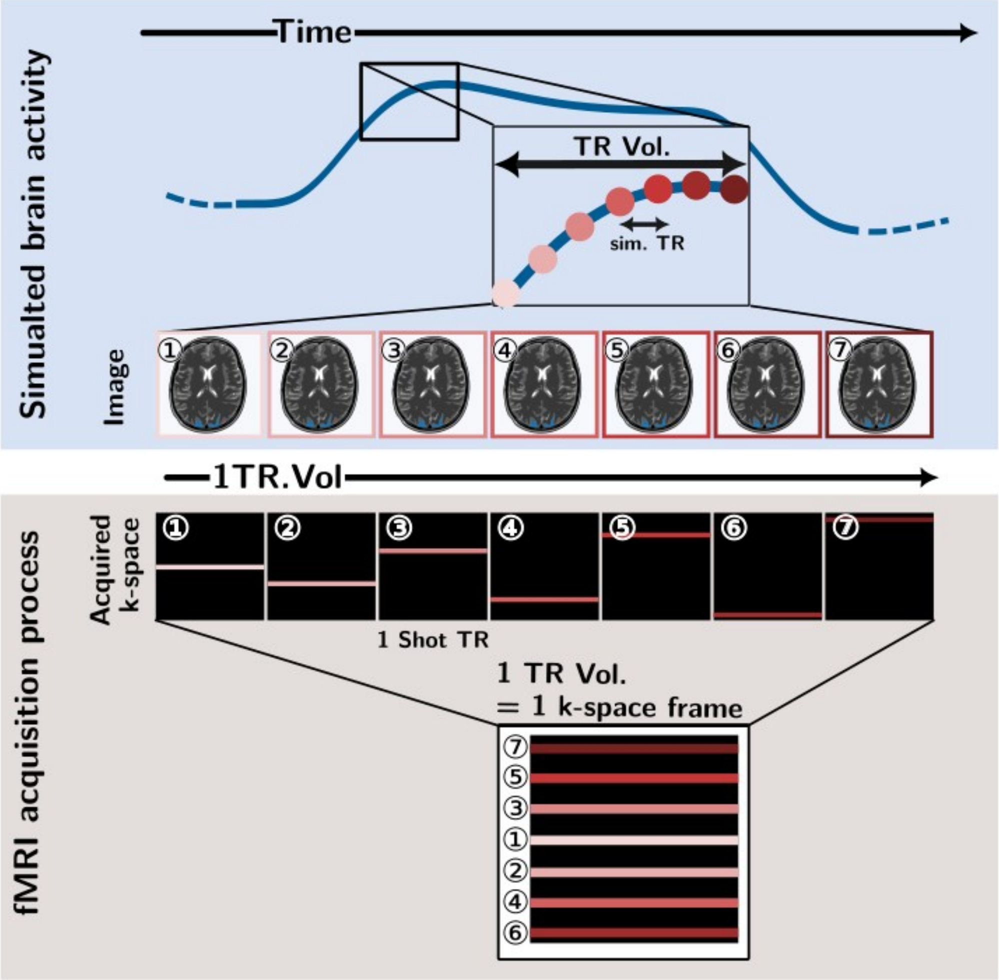
Principle of high temporal resolution for shot-wise acquisition



A focus on the Acquisition Module

- Goal: Sampling the continuous BOLD and Physiology signal at **high temporal resolution**
  - Flexibility for exploring new sampling strategies.
- Computational Efficiency
  - Generate one Volume / shot (TR=50ms)
    - on the fly simulation (low memory usage)
  - Acquisition with Fourier Model (same as reconstruction)
    - No Spin Relaxation Computation: simpler, faster

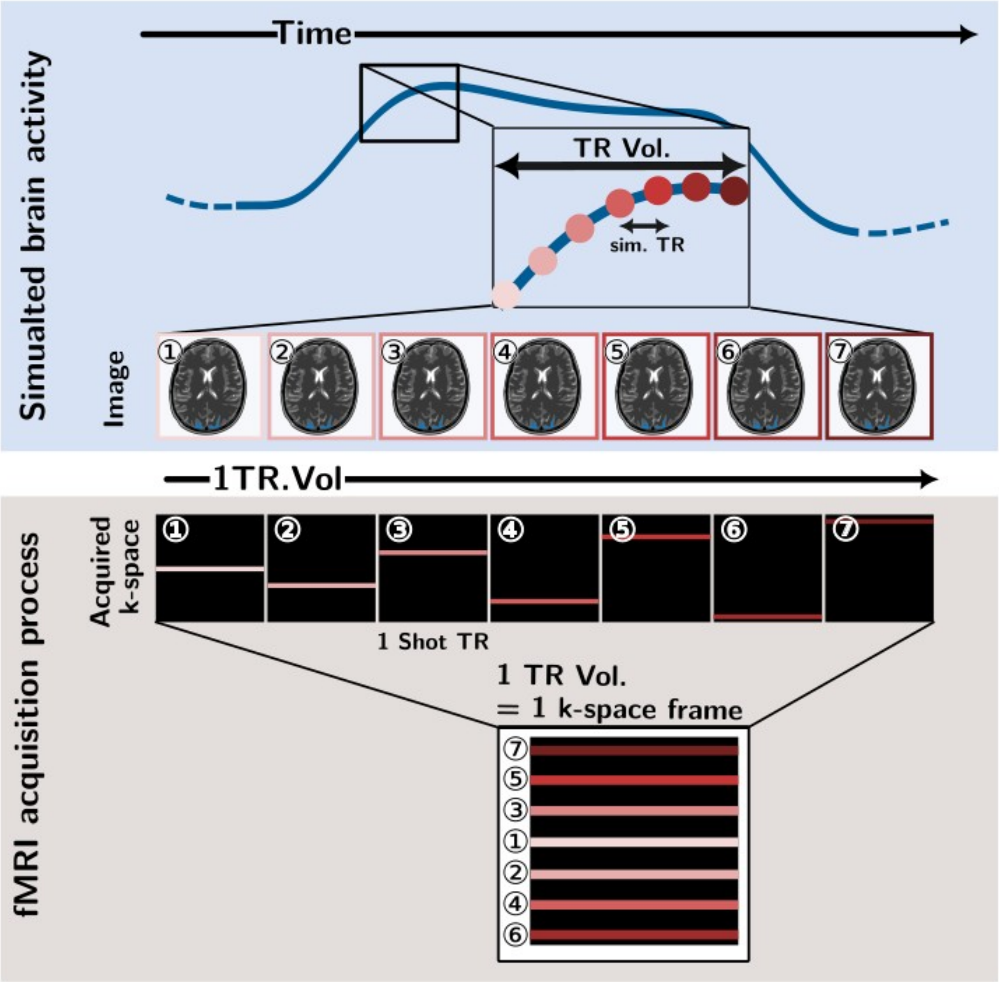
Principle of high temporal resolution for shot-wise acquisition



A focus on the Acquisition Module

- Goal: Sampling the continuous BOLD and Physiology signal at **high temporal resolution**
  - Flexibility for exploring new sampling strategies.
- Computational Efficiency
  - Generate one Volume / shot (TR=50ms)
    - on the fly simulation (low memory usage)
  - Acquisition with Fourier Model (same as reconstruction)
    - No Spin Relaxation Computation: simpler, faster
- Reconstruction of full k-space by grouping shots together
  - Exploration of new grouping strategies
    → see **#3420** (also on Wed. 8th 9:15, Hall 403 - Computer 29)

Principle of high temporal resolution for shot-wise acquisition



Example : Simulated Scenario for Benchmarking

Example : Simulated Scenario for Benchmarking

- Resolution: 3mm-iso, TR_{vol}=0,7s

Example : Simulated Scenario for Benchmarking

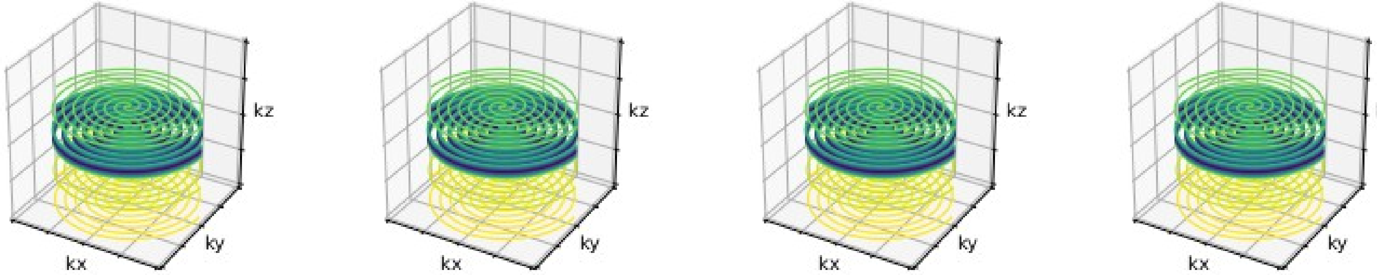
- Resolution: 3mm-iso, TR_{vol}=0,7s
- Acquisition: Stack of Spirals（AF=4）

Example : Simulated Scenario for Benchmarking

- Resolution: 3mm-iso, TR_{vol}=0,7s
- Acquisition: Stack of Spirals (AF=4)

- **Acquisition Strategies**
  - Static (Scan&Repeat) vs Dynamic
  - Gaussian Noise (SNR_{rec} ≈ 30)

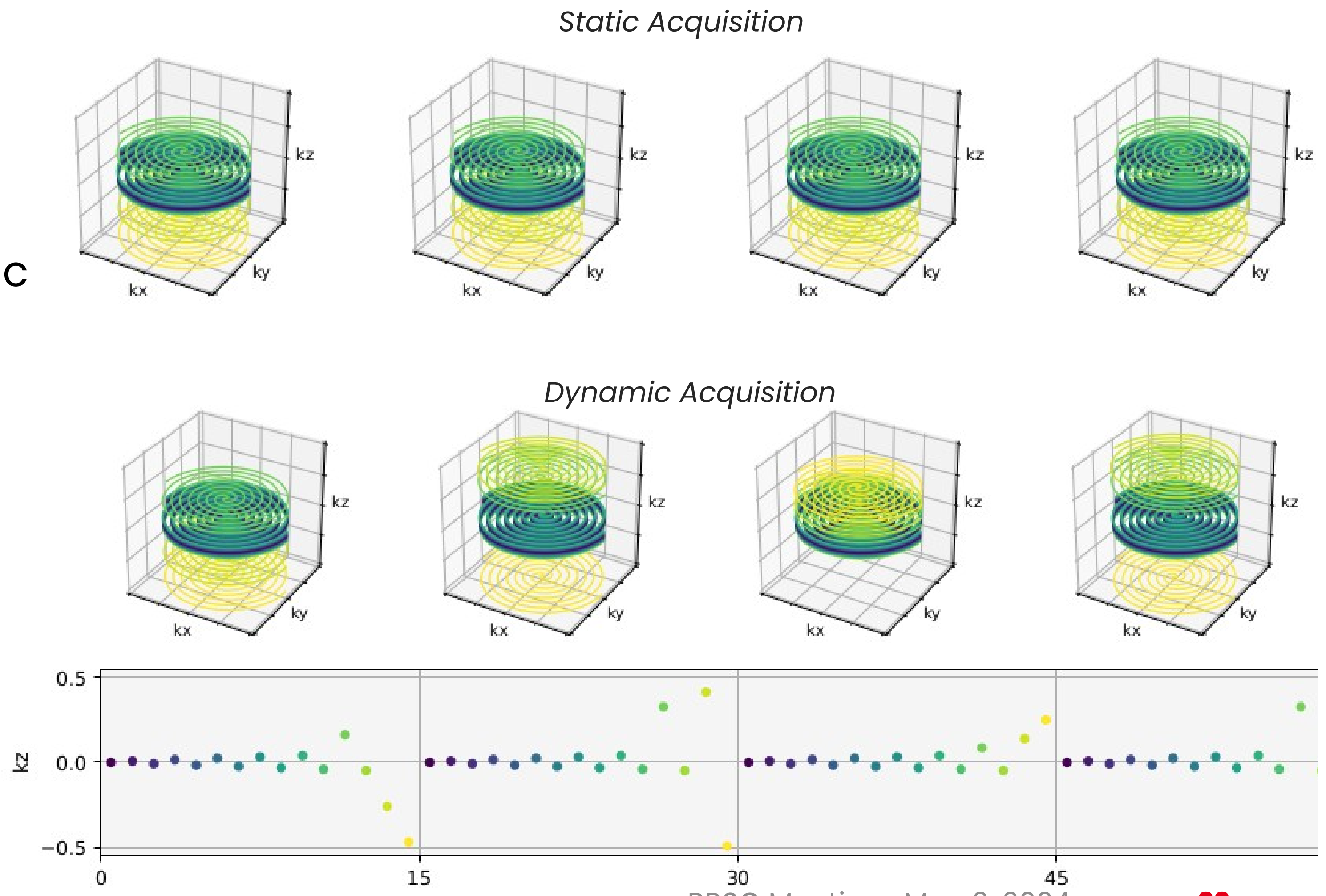
Static Acquisition



Example : Simulated Scenario for Benchmarking

- Resolution: 3mm-iso, TR_{vol}=0,7s
- Acquisition: Stack of Spirals (AF=4)

- **Acquisition Strategies**
  - Static (Scan&Repeat) vs Dynamic
  - Gaussian Noise ($SNR_{rec} \approx 30$)



Static Acquisition

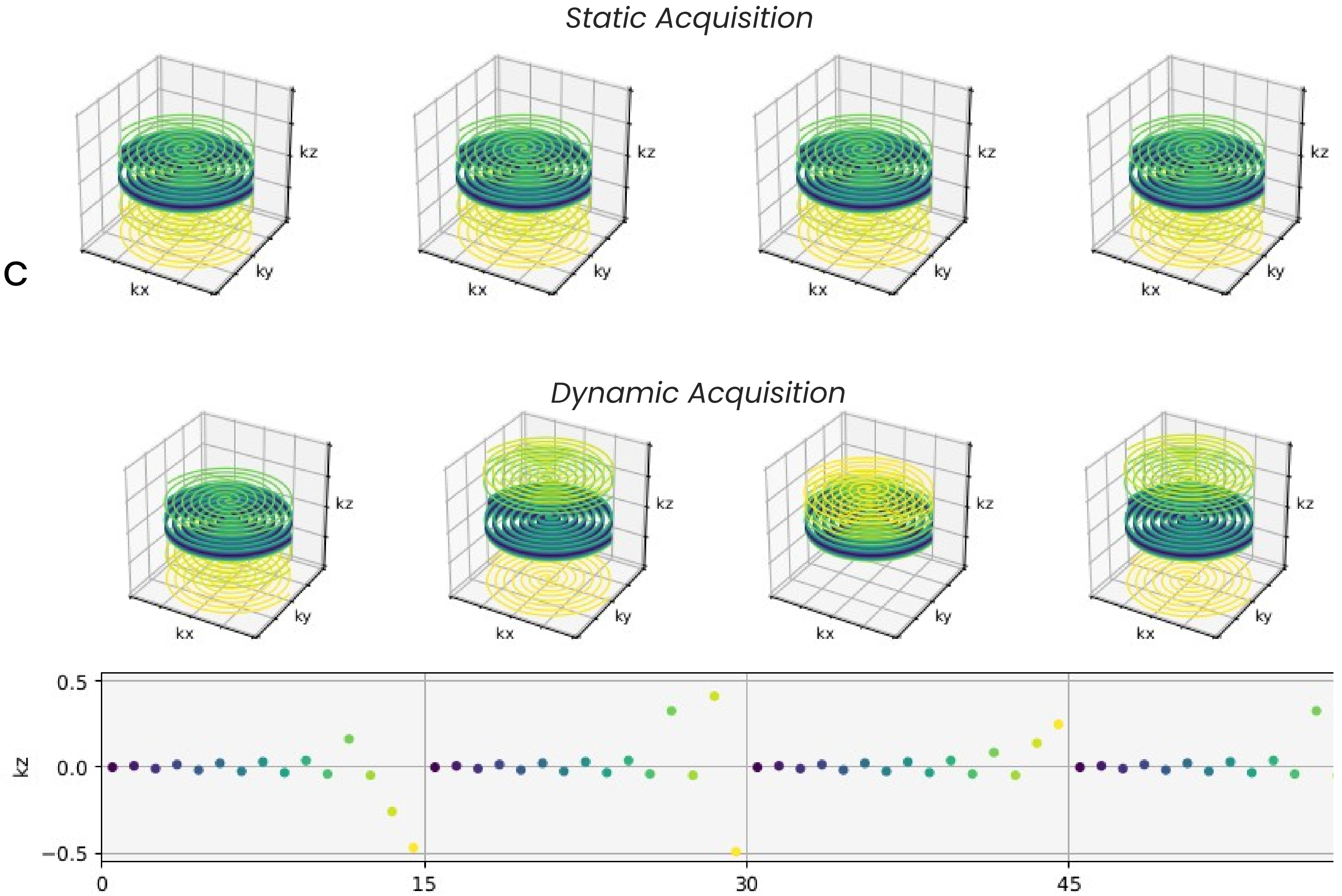
Dynamic Acquisition

Example : Simulated Scenario for Benchmarking

- Resolution: 3mm-iso, TR_{vol}=0,7s
- Acquisition: Stack of Spirals (AF=4)

- **Acquisition Strategies**
  - Static (Scan&Repeat) vs Dynamic
  - Gaussian Noise ($SNR_{rec} \approx 30$)

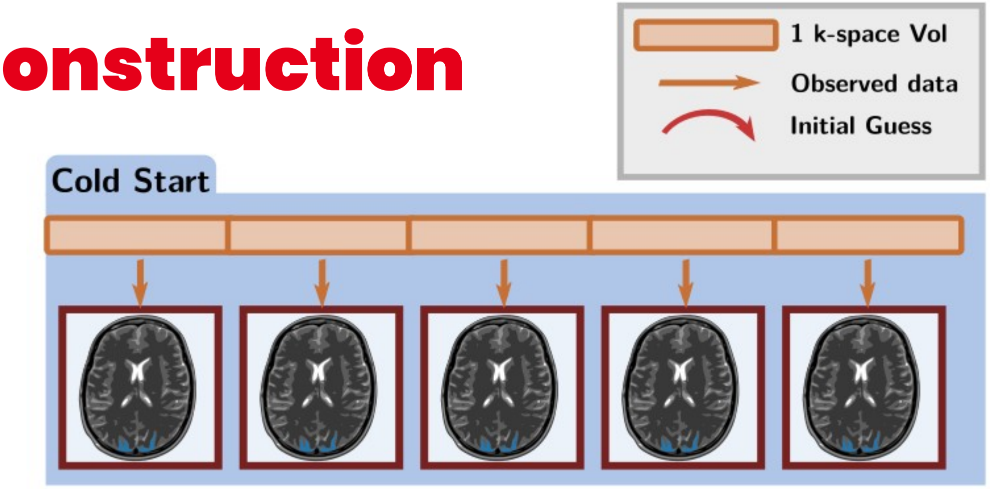
- **Reconstruction methods**
  - Adjoint NUFFT
  - Compressed Sensing
  - Restart Strategies



Static Acquisition

Dynamic Acquisition

FMRI frame-wise Reconstruction



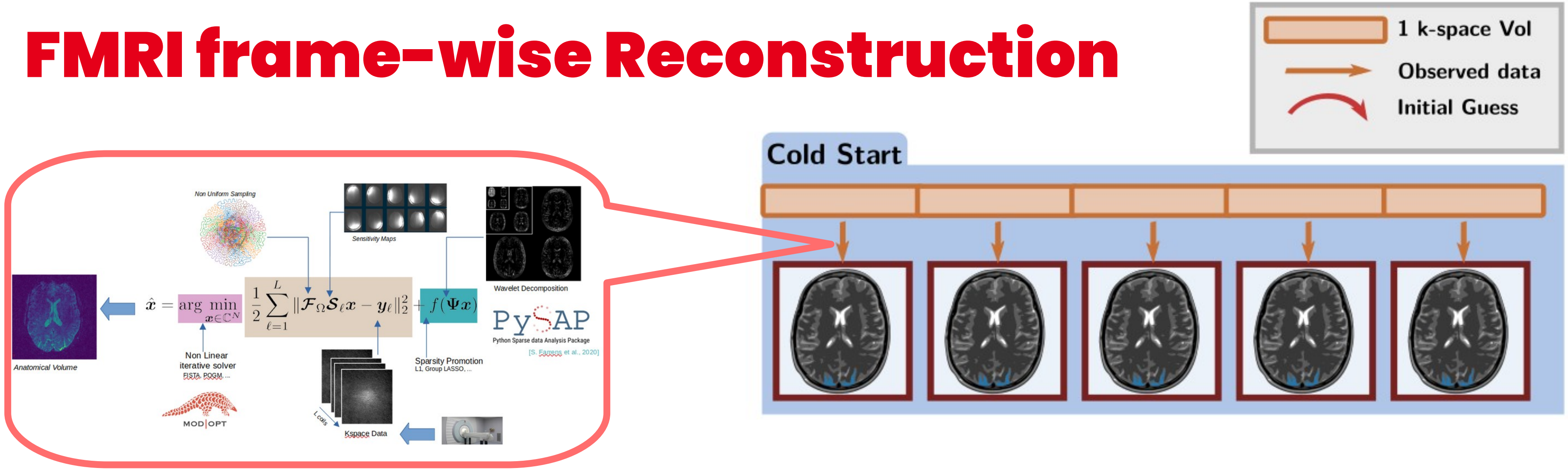
1 k-space Vol

Observed data

Initial Guess

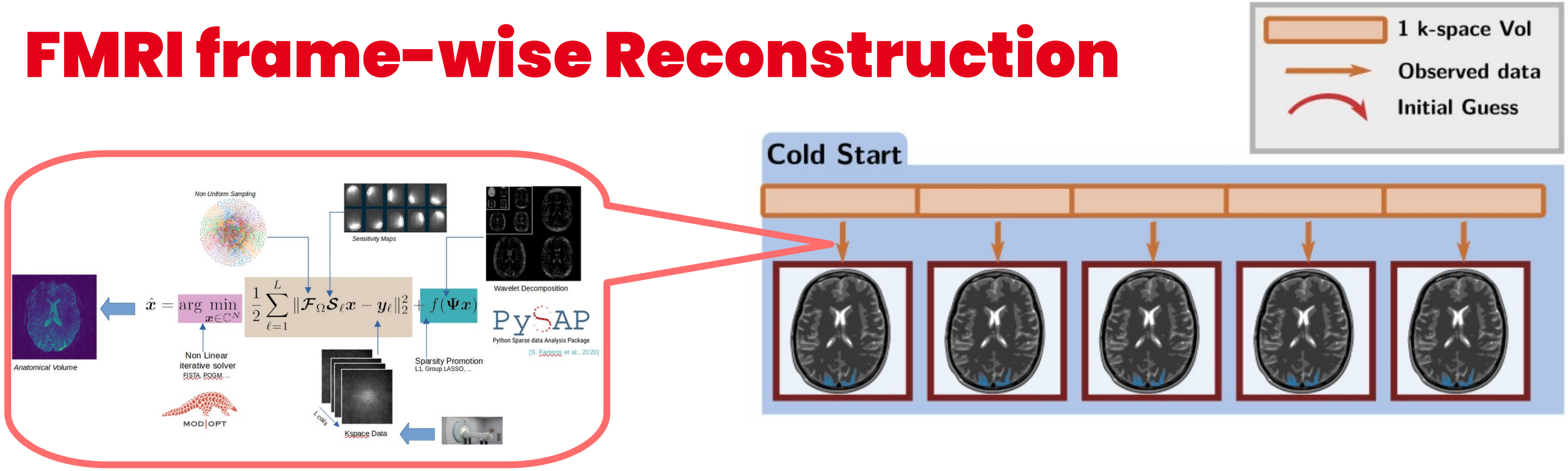
Cold Start

FMRI frame-wise Reconstruction



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

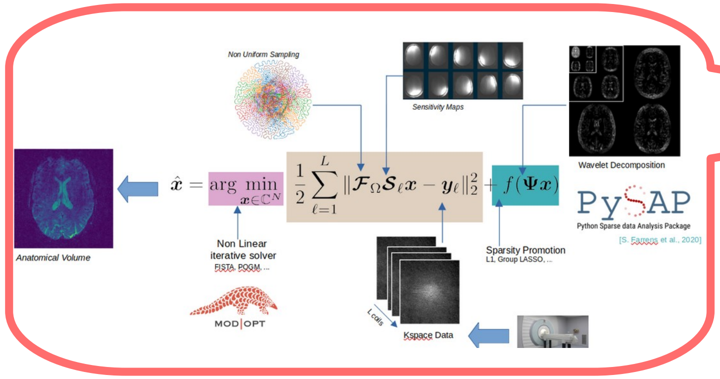
FMRI frame-wise Reconstruction



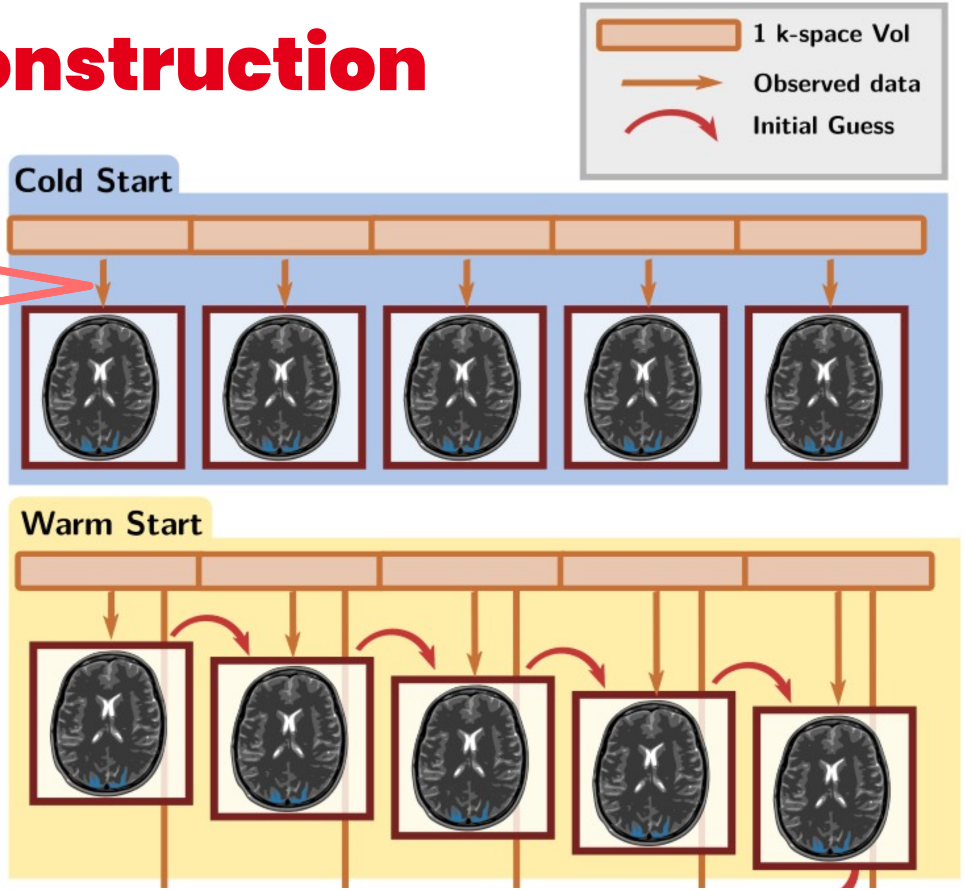
$$\hat{x} = \arg\min_{x \in \mathbb{C}^N} \frac{1}{2} \sum_{\ell=1}^{L} \| \mathcal{F}_\Omega \mathcal{S}_\ell x - y_\ell \|_2^2 + f(\Psi x)$$

@3mm-iso
6s / Volume
with full on-GPU reconstruction

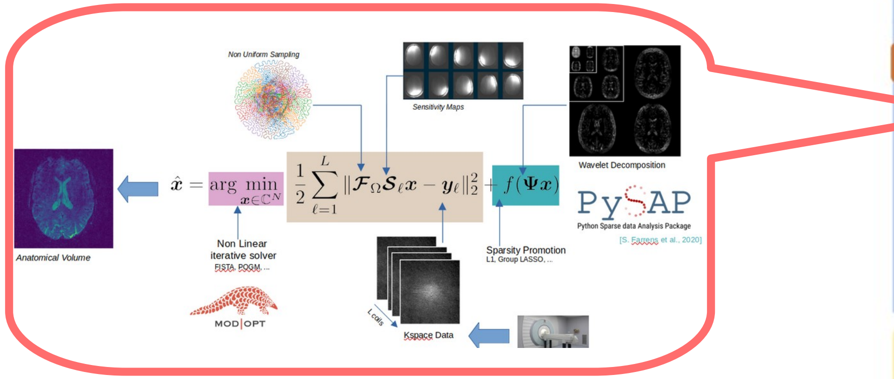
FMRI frame-wise Reconstruction


Non Uniform Sampling
Sensitivity Maps
Wavelet Decomposition
Anatomical Volume
$\hat{x} = \arg\min_{x \in \mathbb{C}^N} \frac{1}{2} \sum_{\ell=1}^{L} \|\mathcal{F}_\Omega \mathcal{S}_\ell x - y_\ell\|_2^2 + f(\Psi x)$
PySAP
Python Sparse data Analysis Package
[S. Farrens et al., 2020]
Non Linear iterative solver
FISTA, POGM, ...
Sparsity Promotion
L1, Group LASSO, ...
MOD|OPT
L coils
Kspace Data

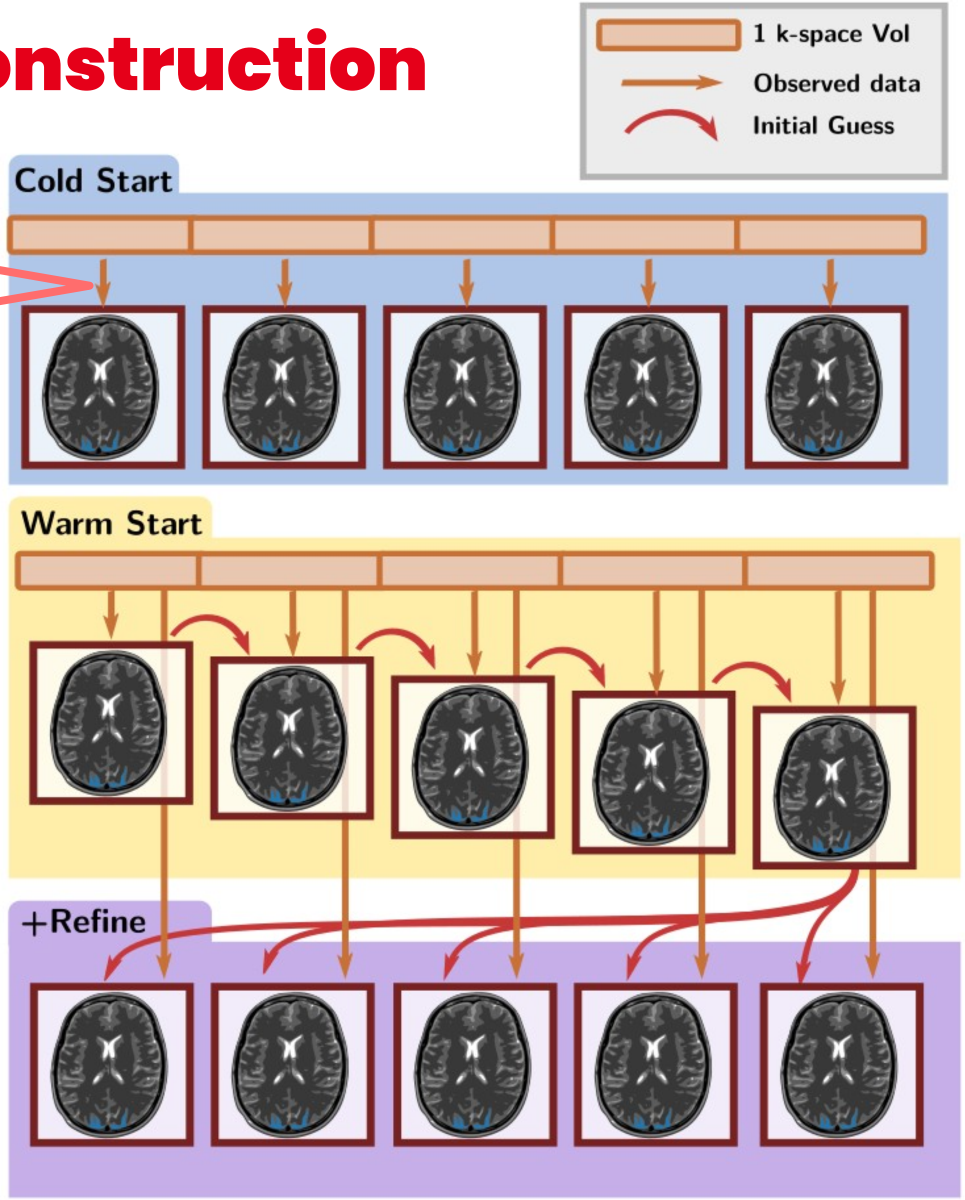
@3mm-iso
6s / Volume
with full on-GPU reconstruction



FMRI frame-wise Reconstruction

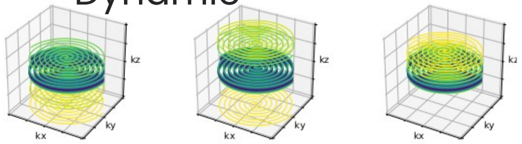

Non Uniform Sampling
Sensitivity Maps
Wavelet Decomposition
Anatomical Volume
$\hat{x} = \arg\min_{x \in \mathbb{C}^N} \frac{1}{2} \sum_{\ell=1}^{L} \|\mathcal{F}_\Omega \mathcal{S}_\ell x - y_\ell\|_2^2 + f(\Psi x)$
PySAP
Python Sparse data Analysis Package
[S. Farrens et al., 2020]
Non Linear iterative solver
FISTA, POGM, ...
MOD|OPT
Sparsity Promotion
L1, Group LASSO, ...
L coils
Kspace Data

@3mm-iso
6s / Volume
with full on-GPU reconstruction

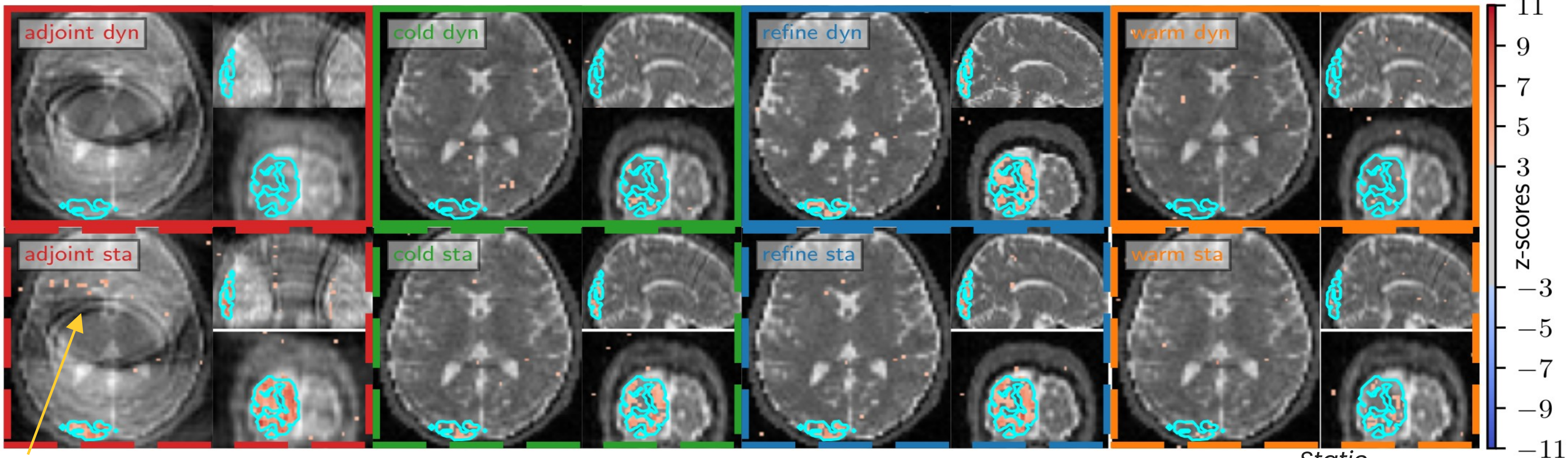


Scenario : Stack of Spirals
0.7s @ 3mm^3

Dynamic

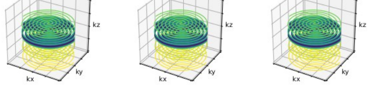


Background Volume: First Reconstructed frame

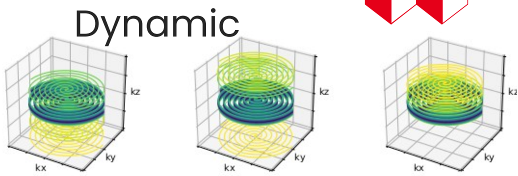


Aliased Activation

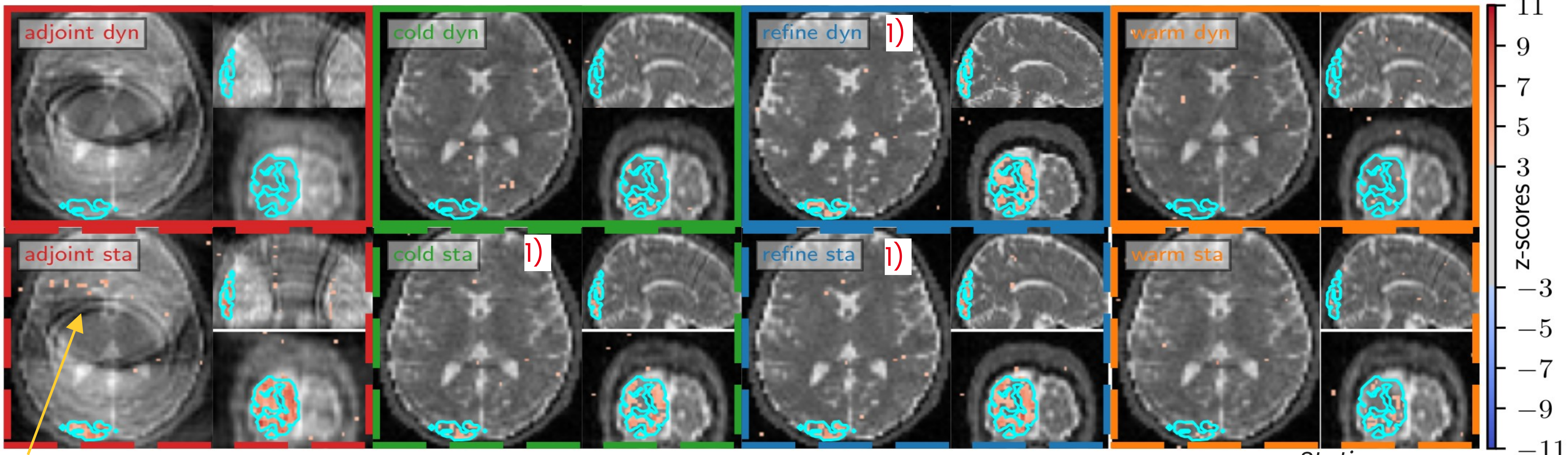
Static



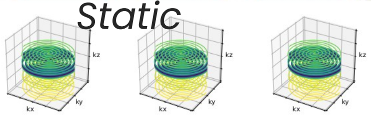
Scenario : Stack of Spirals 0.7s @ 3mm³



Background Volume: First Reconstructed frame



Aliased Activation

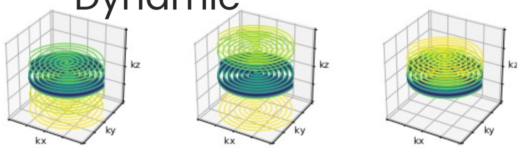


1) Warm start (+Refine) reconstruction is key in dynamic acquisition

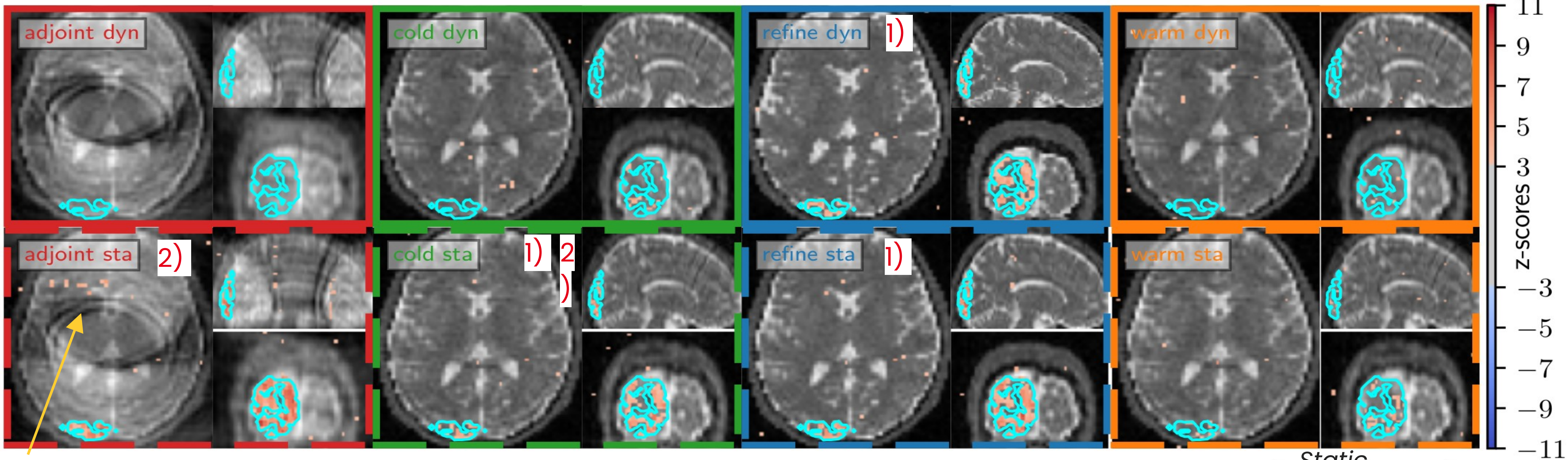
Scenario : Stack of Spirals

0.7s @ 3mm^3

Dynamic

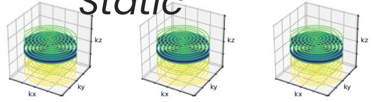


Background Volume: First Reconstructed frame



Aliased Activation

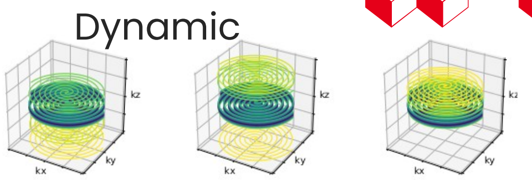
Static



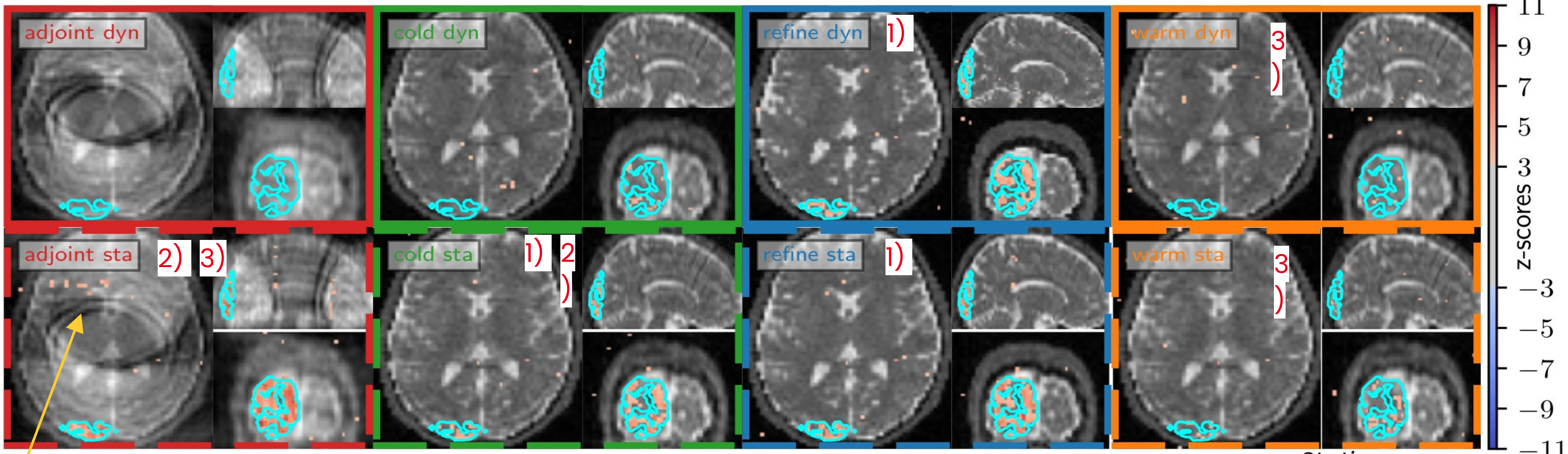
1) Warm start (+Refine) reconstruction is key in dynamic acquisition

2) Static acquisition boosts sensitivity at the cost of specificity and image quality

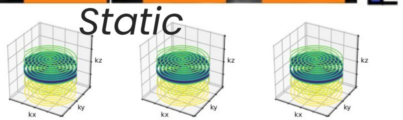
Scenario : Stack of Spirals 0.7s @ 3mm³



Background Volume: First Reconstructed frame



Aliased Activation



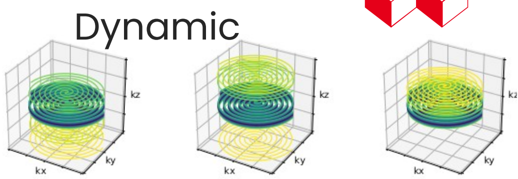
1) Warm start (+Refine) reconstruction is key in dynamic acquisition

2) Static acquisition boosts sensitivity at the cost of specificity and image quality

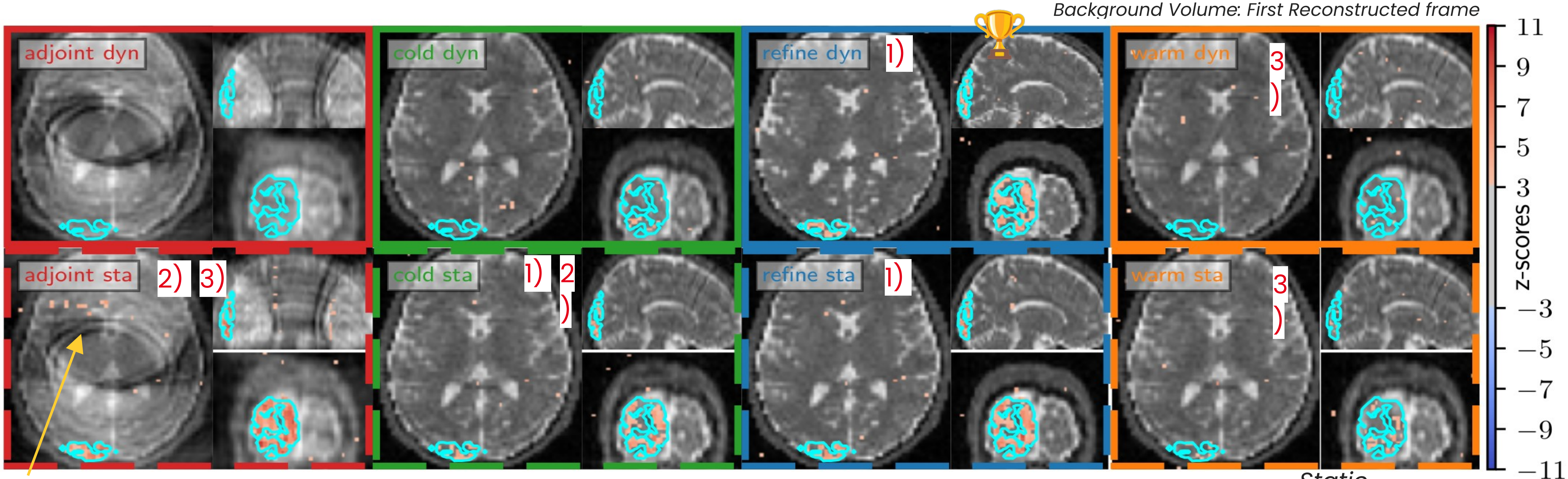
3) Image Quality is not a proxy for good statistical performances.

Scenario : Stack of Spirals

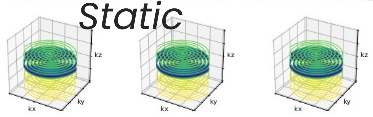
0.7s @ $3mm^3$



Background Volume: First Reconstructed frame



Aliased Activation



1) Warm start (+Refine) reconstruction is key in dynamic acquisition

2) Static acquisition boosts sensitivity at the cost of specificity and image quality

3) Image Quality is not a proxy for good statistical performances.

🏆 Best Strategy is Dynamic + Refine

Extensibility and Usability of SNAKE-fMRI

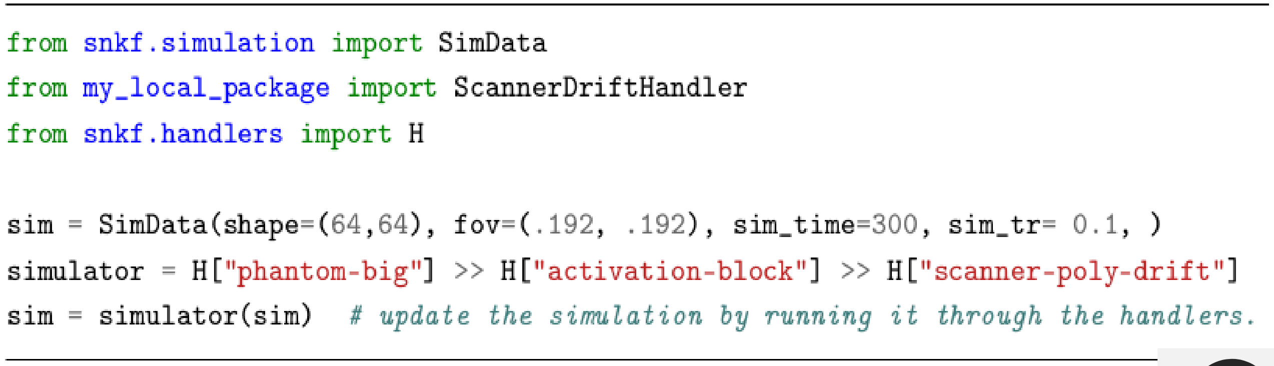
https://github.com/paquiteau/snake-fmri ⭐

https://hal.science/hal-04533862v1/document

Thank you for your attention

Extensibility and Usability of SNAKE-fMRI

- Functional API for prototyping



```
from snkf.simulation import SimData
from my_local_package import ScannerDriftHandler
from snkf.handlers import H

sim = SimData(shape=(64,64), fov=(.192, .192), sim_time=300, sim_tr= 0.1, )
simulator = H["phantom-big"] >> H["activation-block"] >> H["scanner-poly-drift"]
sim = simulator(sim)  # update the simulation by running it through the handlers.
```

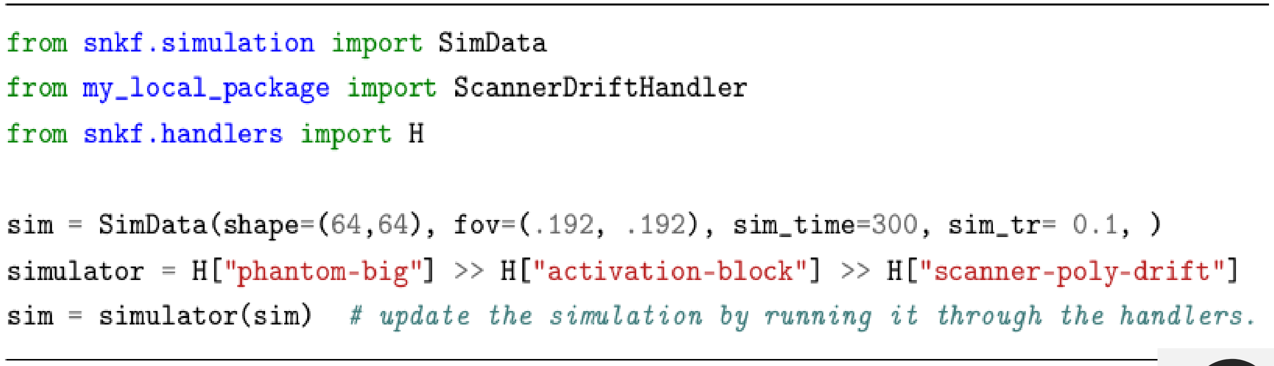
https://github.com/paquiteau/snake-fmri ⭐

https://hal.science/hal-04533862v1/document

Thank you for your attention

Extensibility and Usability of SNAKE-fMRI

- Functional API for prototyping



```
from snkf.simulation import SimData
from my_local_package import ScannerDriftHandler
from snkf.handlers import H

sim = SimData(shape=(64,64), fov=(.192, .192), sim_time=300, sim_tr= 0.1, )
simulator = H["phantom-big"] >> H["activation-block"] >> H["scanner-poly-drift"]
sim = simulator(sim)  # update the simulation by running it through the handlers.
```

- Configuration file and CLI

https://github.com/paquiteau/snake-fmri ⭐

https://hal.science/hal-04533862v1/document

Thank you for your attention

Extensibility and Usability of SNAKE-fMRI

- Functional API for prototyping

```
from snkf.simulation import SimData
from my_local_package import ScannerDriftHandler
from snkf.handlers import H

sim = SimData(shape=(64,64), fov=(.192, .192), sim_time=300, sim_tr= 0.1, )
simulator = H["phantom-big"] >> H["activation-block"] >> H["scanner-poly-drift"]
sim = simulator(sim)  # update the simulation by running it through the handlers.
```

https://github.com/paquiteau/snake-fmri ⭐

https://hal.science/hal-04533862v1/document

- Configuration file and CLI

```
$ pip install snake-fmri
$ snkf-main --config-name="scenario1"
# Using Hydra, parameters can be modified and run over a grid of parameter.
$ snkf-main --config-name="scenario2" -m ++reconstructors.sequential.restart_strategy=cold,warm,refine
```

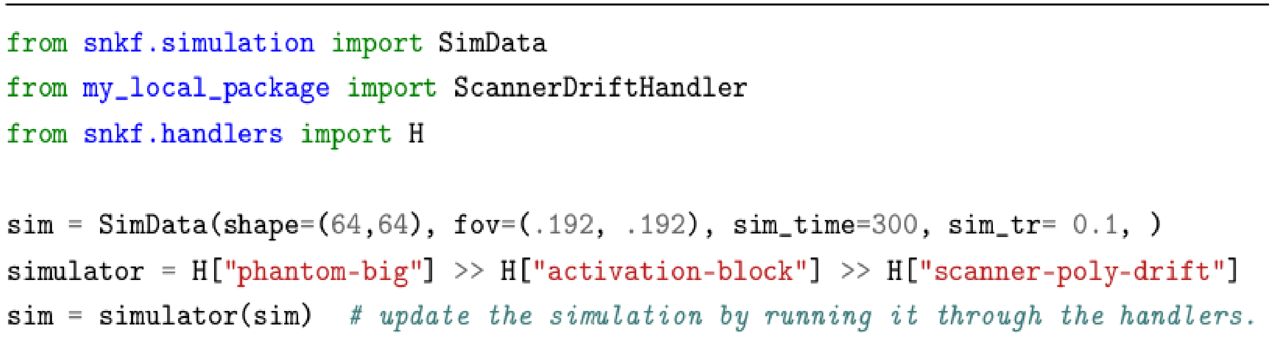
To reproduce data of the previous slide

Thank you for your attention

Extensibility and Usability of SNAKE-fMRI

Digital Poster Presentation
Wednesday 8th 09:15
Hall 403 - Computer 17

- Functional API for prototyping



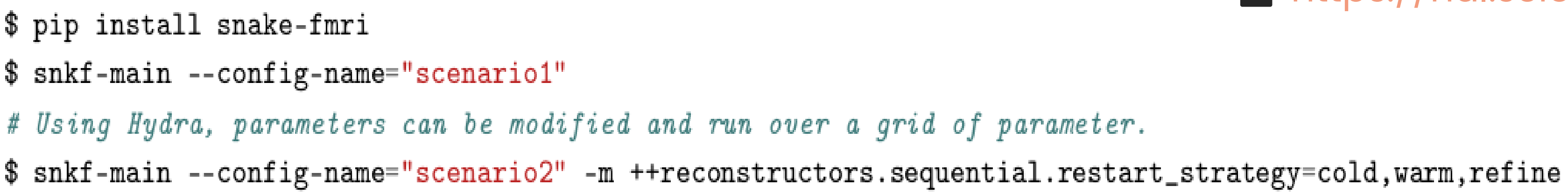
```
from snkf.simulation import SimData
from my_local_package import ScannerDriftHandler
from snkf.handlers import H

sim = SimData(shape=(64,64), fov=(.192, .192), sim_time=300, sim_tr= 0.1, )
simulator = H["phantom-big"] >> H["activation-block"] >> H["scanner-poly-drift"]
sim = simulator(sim)  # update the simulation by running it through the handlers.
```

https://github.com/paquiteau/snake-fmri ⭐

https://hal.science/hal-04533862v1/document

- Configuration file and CLI



```
$ pip install snake-fmri
$ snkf-main --config-name="scenario1"
# Using Hydra, parameters can be modified and run over a grid of parameter.
$ snkf-main --config-name="scenario2" -m ++reconstructors.sequential.restart_strategy=cold,warm,refine
```

To reproduce data of the previous slide

Thank you for your attention