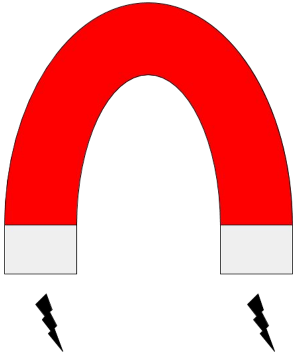
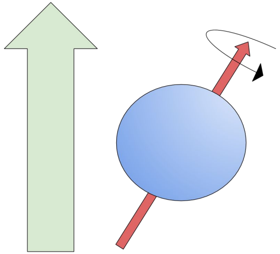
XPDNet for brain multi-coil MRI reconstruction

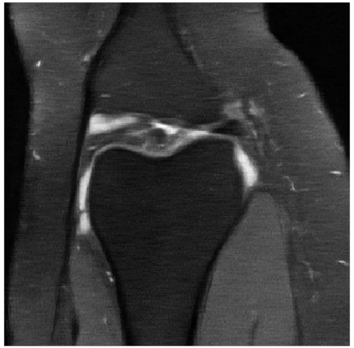
Zaccharie Ramzi
(zaccharie.ramzi@inria.fr)
supervised by Philippe Ciuciu and
Jean-Luc Starck

What is MRI?

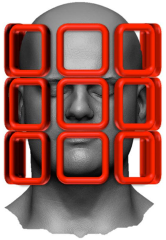
MRI = **M**agnetic **R**esonance **I**maging

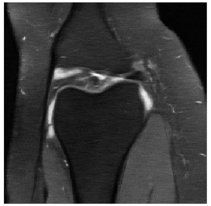


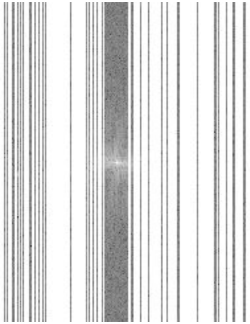




The (idealized) inverse problem







$$\forall j, \; F_{\Omega} S_j x = y_j$$

Masked 2D Fourier Transform, **well-defined** in the fastMRI case

The “periodic” mask, **known** but **specific to each acquisition**, part of the **input**

The sensitivity maps, **unknown** and **specific to each acquisition**

The anatomical image, the **output** of our reconstruction

The k-space measurements for all coils, part of the **input** to our reconstruction (complex-valued)

Where is the room for learning?

$$\underset{x \in \mathbb{C}^n}{\arg\min} \sum_j \frac{1}{2} \| y_j - F_\Omega S_j x \|_2^2 + R(x)$$

Optimization algorithm parameters, such as **step sizes** or **iterates merging**, can be learned

Noise model data consistency distance, usually set to be AWGN, we can learn the equivalent of its **gradient**

Sensitivity maps can be computed using l1-SPIRiT, but we can learn how to **refine them**

Prior term usually set by hand, we can learn the equivalent of its **proximity operator**

Outline

1. Cross domain learning
2. Measurement operator refinement
3. Proximity operator learning
4. Data consistency learning

Outline

1. Cross domain learning
2. Measurement operator refinement
3. Proximity operator learning
4. Data consistency learning

Cross-domain learning

Key intuitive idea: Alternate the corrections between image domain and k-space, "comparing" the original k-space measurements with the current iterate's k-space measurements.

Tool for that: unrolling optimisation algorithms.

Correction in the **k-space**

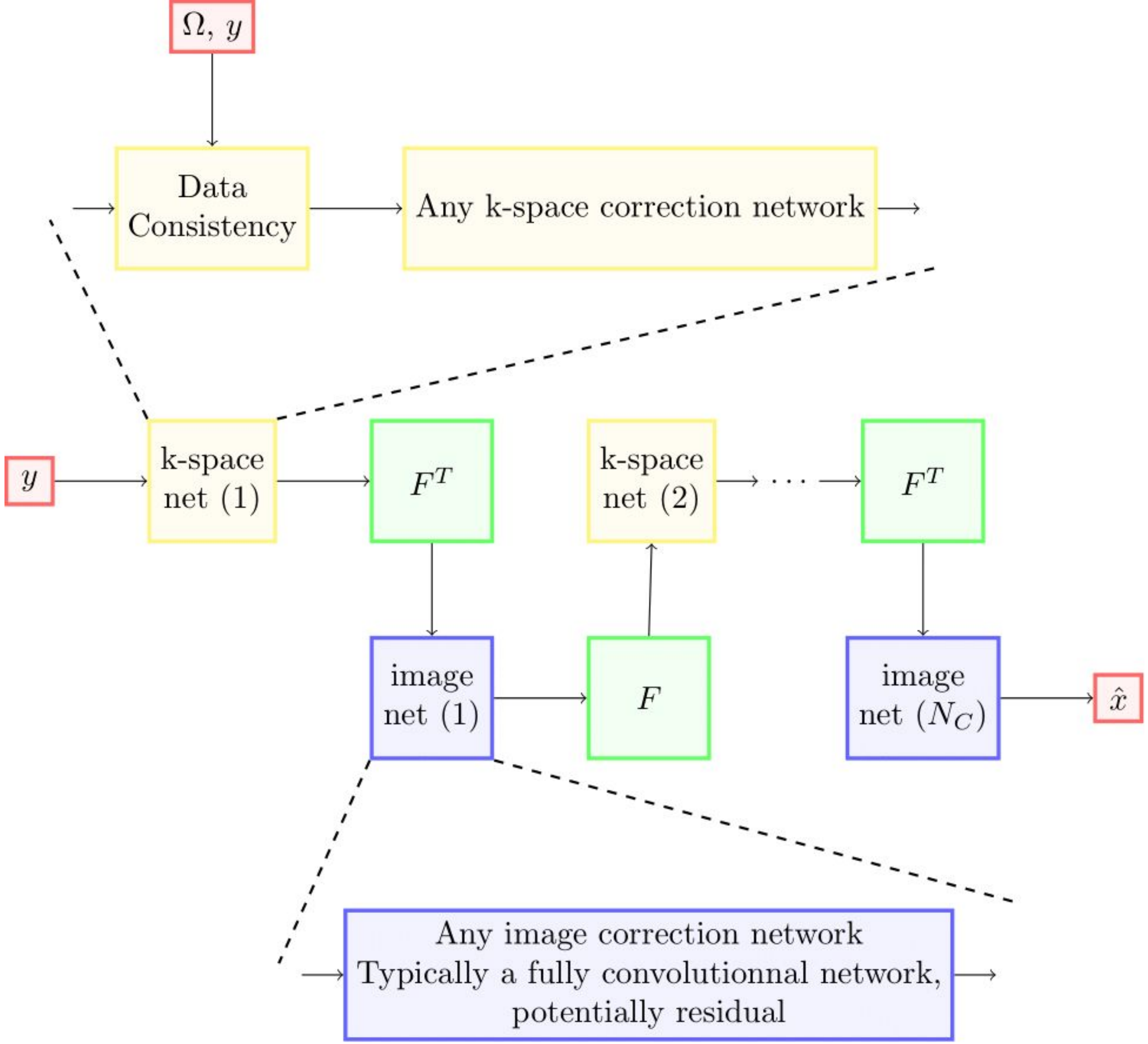
Current k-space measurements

$$x_{n+1} = x_n + \epsilon_n \, (F_\Omega S)^H (F_\Omega S \, x_n - y)$$

$$x_{n+1} = \mathrm{prox}_{\epsilon_n R}\,(x_{n+1})$$

Correction in the **image domain**

Original k-space measurements



Ω, y
Data Consistency
Any k-space correction network
y
k-space net (1)
F^T
k-space net (2)
F^T
image net (1)
F
image net (N_C)
$\hat{x}$
Any image correction network
Typically a fully convolutionnal network,
potentially residual

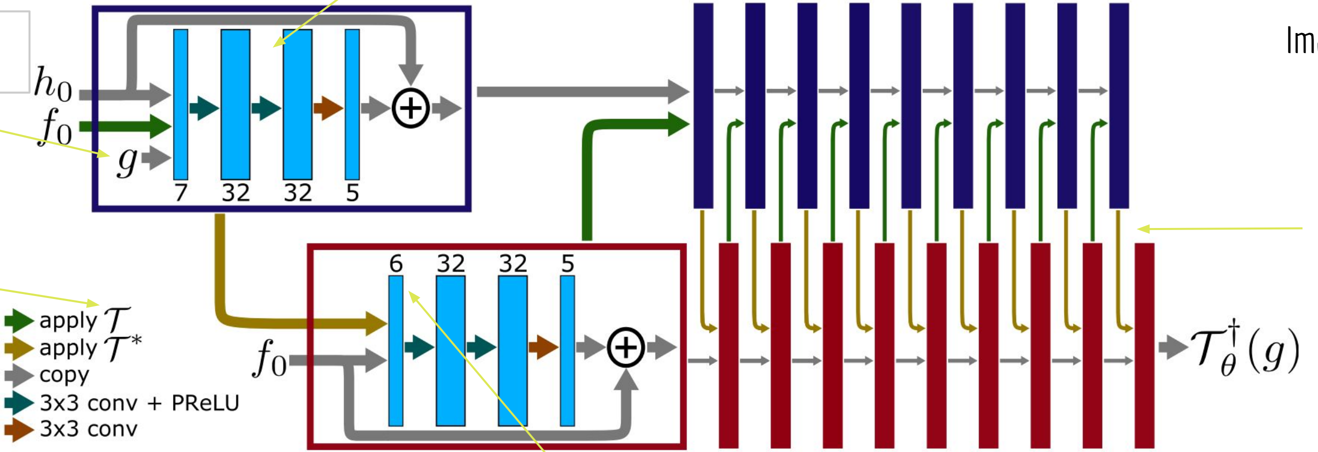
Cross-domain learning - unrolling PDHG [1]

PD-net unrolls the PDHG, where the data consistency is a (learned) residual

We don't use a k-space correction network, and just use a residual

Original measurements

Image from [1]



Masked, multicoil Fourier transform

We unroll **25** iterations

Buffers are used to learn the merging of consecutive iterates

[1] Learned Primal-Dual Reconstruction, Adler et al. 2018

Outline

1. Cross domain learning
2. Measurement operator refinement
3. Proximity operator learning
4. Data consistency learning

Sensitivity maps refinement [2]

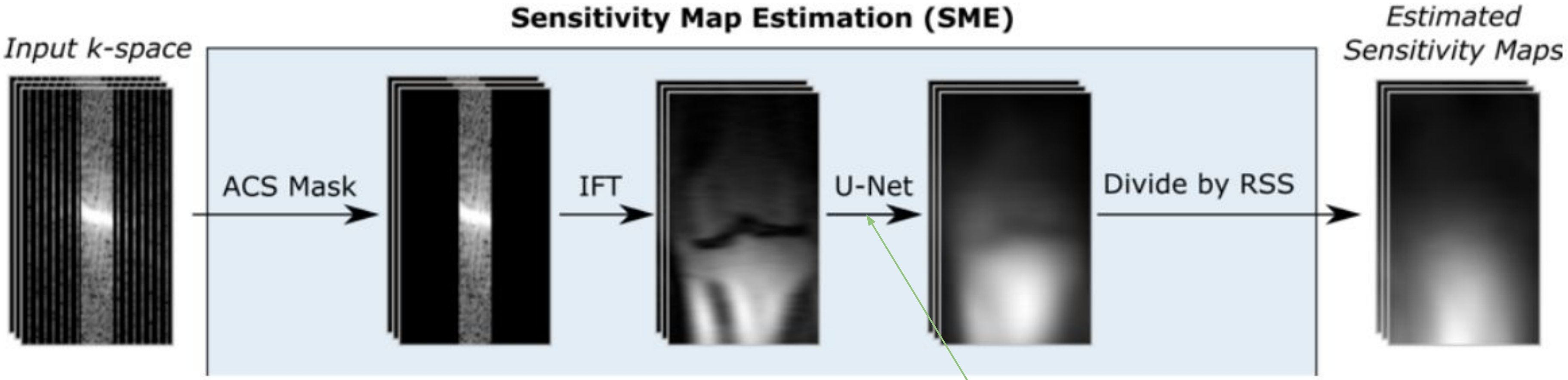


Image from [2]

The U-net is applied to all the coarse sensitivity maps with the same weights.

[2] End-to-End Variational Networks for Accelerated MRI Reconstruction, Sriram et al. 2020

Outline

1. Cross domain learning
2. Measurement operator refinement
3. Proximity operator learning
4. Data consistency learning

Proximity operator architecture [3]

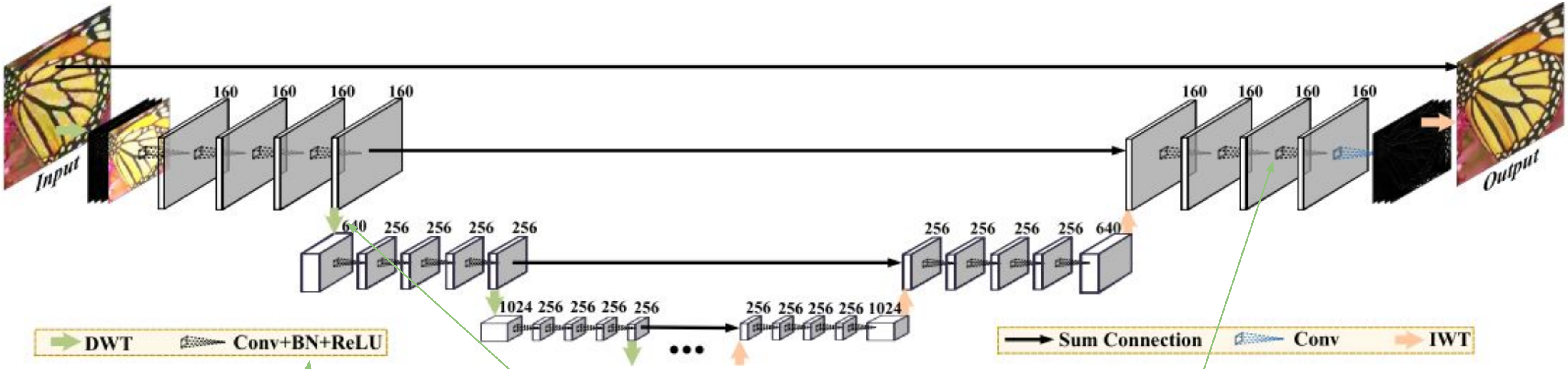


Image from [3]

Pooling is done using a simple wavelet operator (same for unpooling)

We **omit** Batch Normalisation in the architecture given the batch size is 1

To fit on a single GPU, we use **less feature channels** (64, 128, 256)

[3] Multi-level Wavelet-CNN for Image Restoration, Liu et al. 2018

Outline

1. Cross domain learning
2. Measurement operator refinement
3. Proximity operator learning
4. Data consistency learning

Data consistency learning

I didn't use it during the challenge... No good reason (I thought memory was going to be an issue).

Adding a CNN here

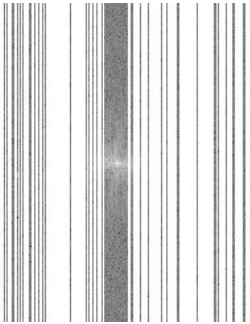
$$x_{n+1} = x_n + \epsilon_n (F_\Omega S)^H (F_\Omega S\, x_n - y)$$

$$x_{n+1} = \mathrm{prox}_{\epsilon_n R}(x_{n+1})$$

Data

We use the fastMRI dataset [4], which is composed of:

- 4469 train volumes (1.2 Tb), 1378 volumes, 558 test volumes (unknown ground truth)
- Input data is generated retrospectively



[4] fastMRI: An Open Dataset and Benchmarks for Accelerated MRI, J. Zbontar et al. 2018.

Training details

- Data scaling by 10^6 (not grid-searched)
- RAdam optimizer (with default params.), learning rate of 10^{-4} [5]
- Total number of steps 446.9k (= 100 epochs x 4469 volumes) => 1 week on V100 for 155.5M params.
- Batch size of 1 (we select one slice at random during each epoch)
- Training is done separately for Accel. factor 4 and 8, then the network is fine-tuned per-contrast
- Loss is a compound L1-MSSIM as advised in [6]

[5] On the Variance of the Adaptive Learning Rate and Beyond, Liu et al. 2020
[6] An Adaptive Intelligence Algorithm for Undersampled Knee MRI Reconstruction: Application to the 2019 fastMRI Challenge, Pezzotti et al. 2020

2020 fastMRI challenge - AF4 quantitative podium

AIRS-Net 10/15/2020	4x	0.0028	0.9640	42.2	✓
Joint-ICNet 10/15/2020	4x	0.0035	0.9599	41.2	✓
xpdnet_v2 10/15/2020	4x	0.0034	0.9589	41.3	✓

2020 fastMRI challenge - AF8 quantitative podium

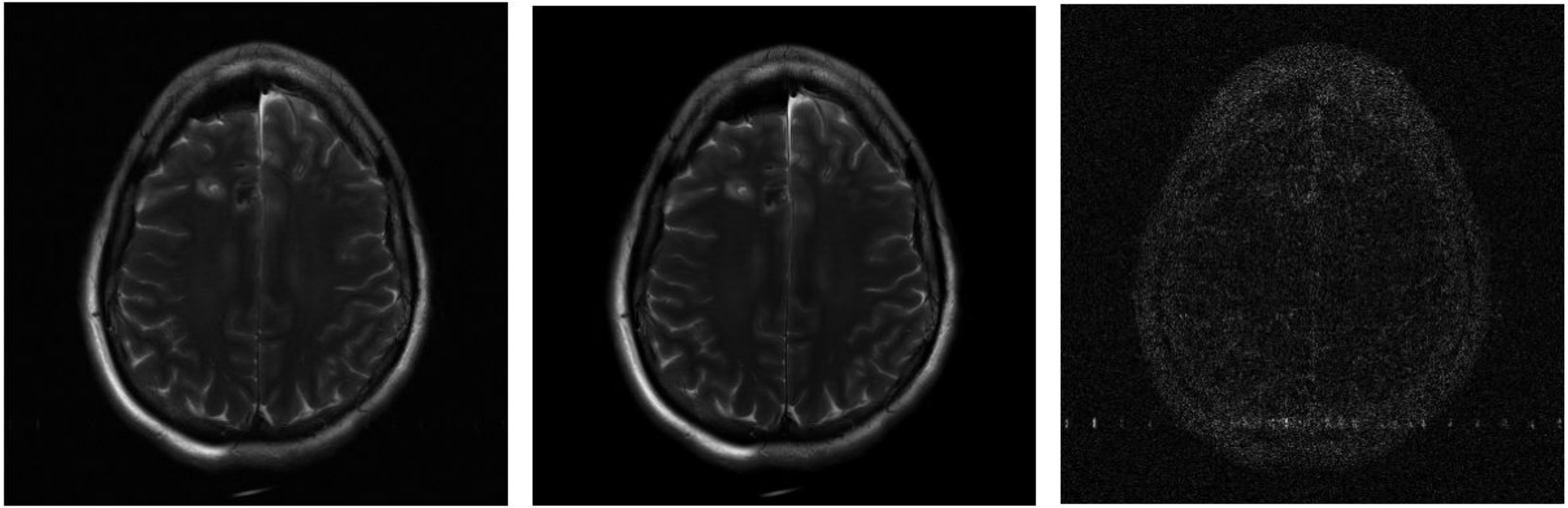
AIRS-Net 10/15/2020	8x	0.0050	0.9524	39.6	✓
Joint-ICNet 10/15/2020	8x	0.0073	0.9437	37.9	✓
xpdnet_v2 10/15/2020	8x	0.0072	0.9420	38.0	✓

2020 fastMRI challenge - qualitative podium [7]

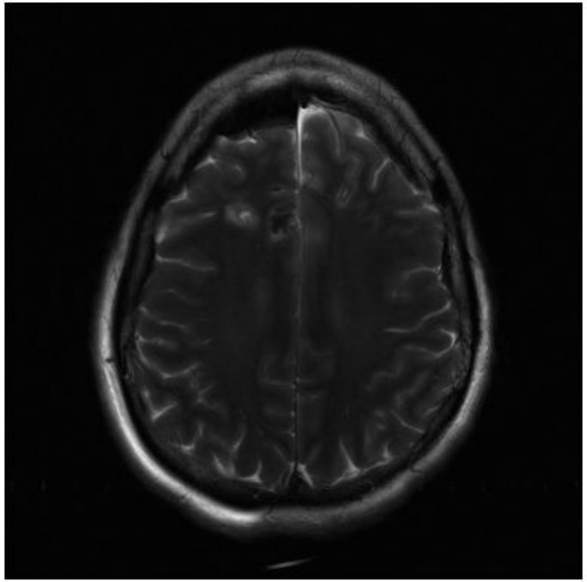
Team	Rank	Artifacts	Sharpness	CNR
4X Track				
AIRS Medical	**1.36 ± 0.64**	**1.53 ± 0.70**	**1.53 ± 0.51**	**1.53 ± 0.51**
Neurospin	1.94 ± 0.86	1.81 ± 1.01	1.72 ± 0.66	1.75 ± 0.84
ATB	2.22 ± 0.87	1.75 ± 0.97	1.97 ± 0.65	1.86 ± 0.80
8X Track				
AIRS Medical	**1.28 ± 0.64**	**1.67 ± 0.68**	**1.89 ± 0.75**	**1.94 ± 0.75**
Neurospin	2.25 ± 0.77	1.86 ± 0.83	2.72 ± 0.81	2.28 ± 0.81
ATB	2.28 ± 0.70	1.92 ± 0.94	2.56 ± 0.77	2.42 ± 0.84

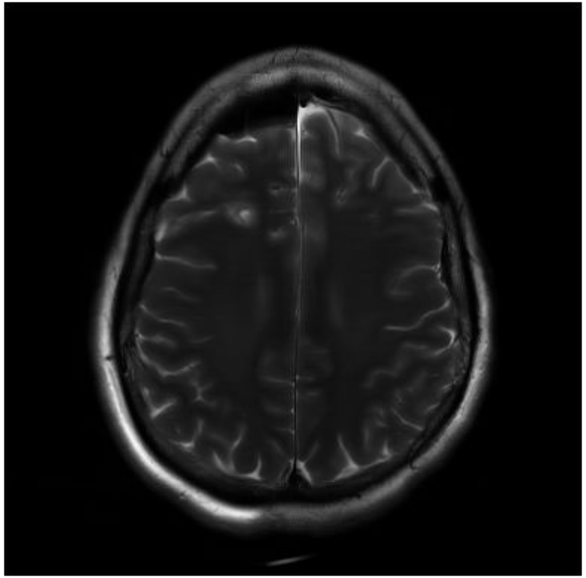
[7] State-of-the-art Machine Learning MRI Reconstruction in 2020: Results of the Second fastMRI Challenge, Mcukley et al. 2020

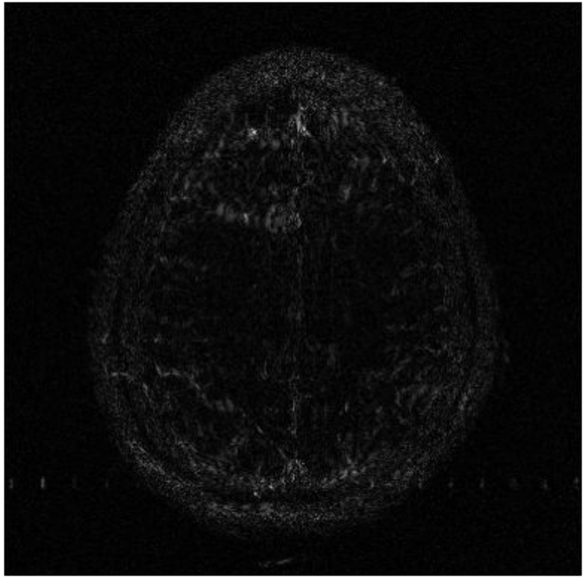
Qualitative validation results - T2 - R4



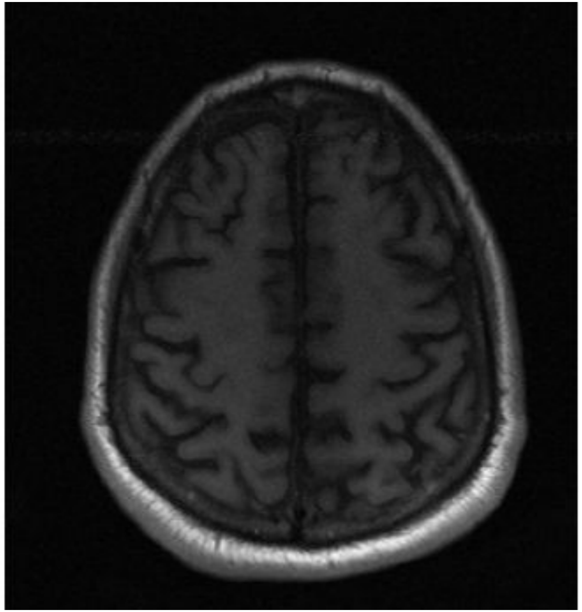
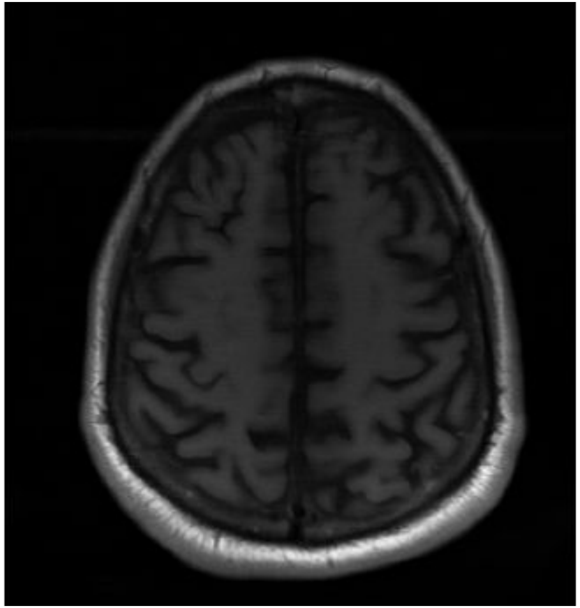
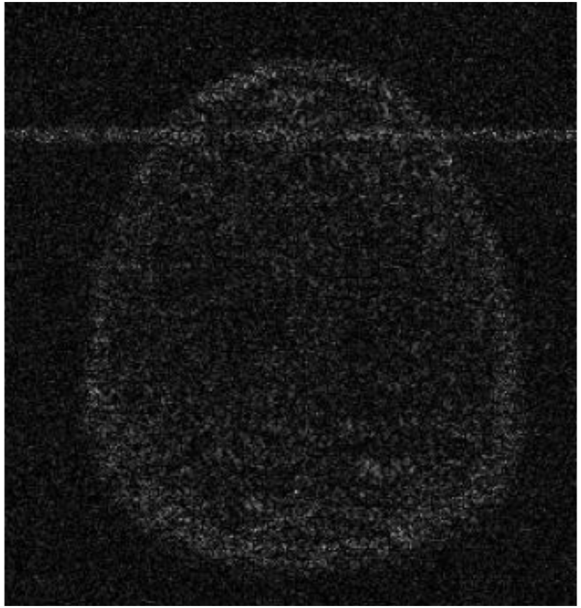
Qualitative validation results - T2 - R8



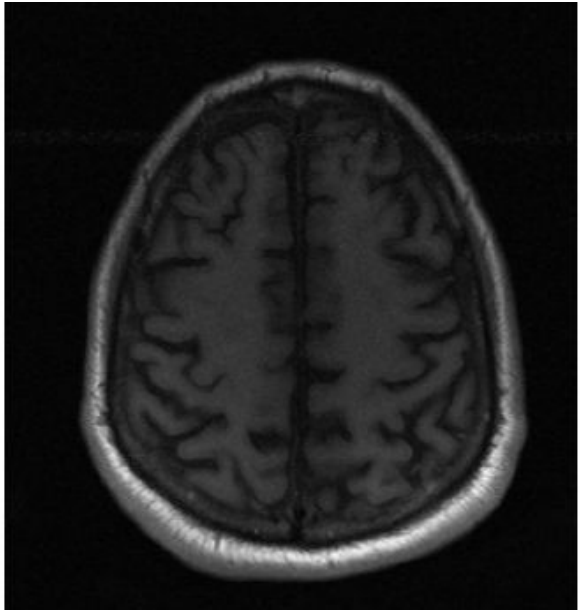


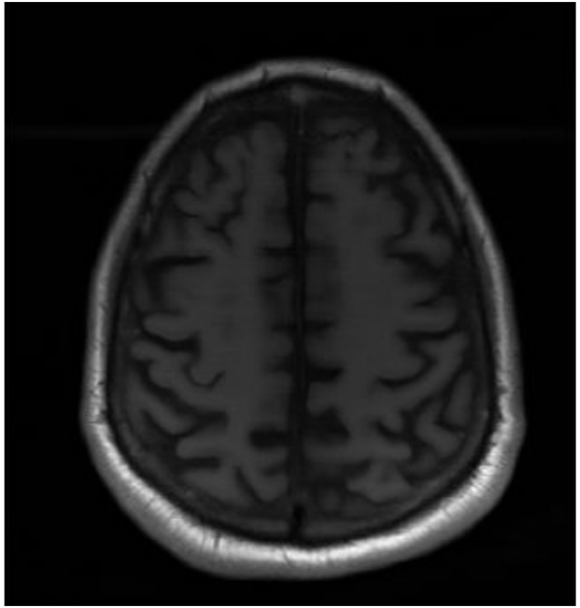


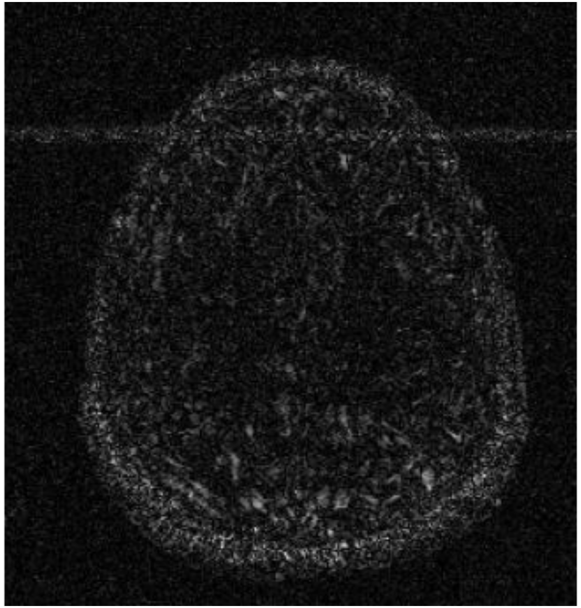
Qualitative validation results - T1 - R4

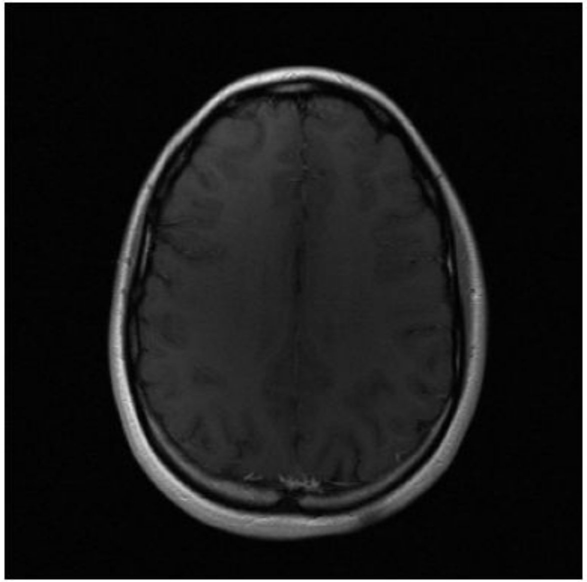
Qualitative validation results - T1 - R8

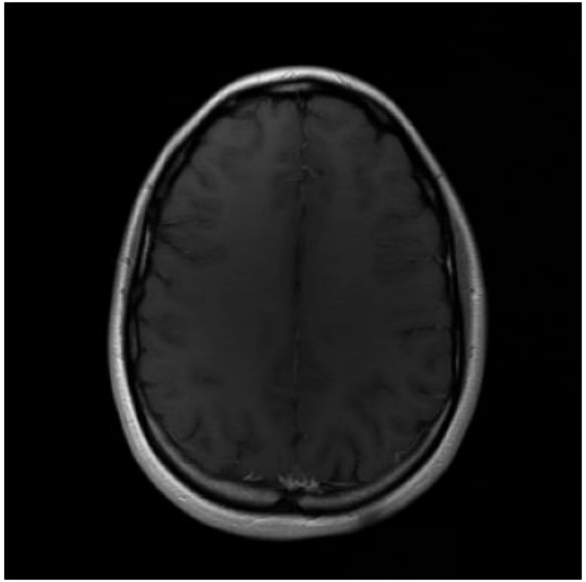


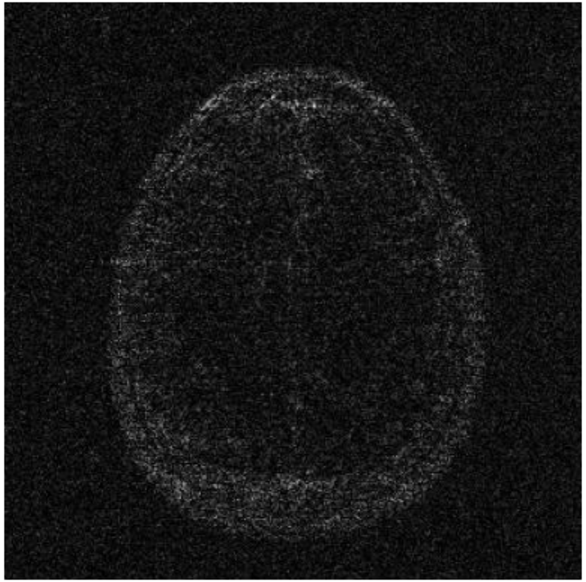




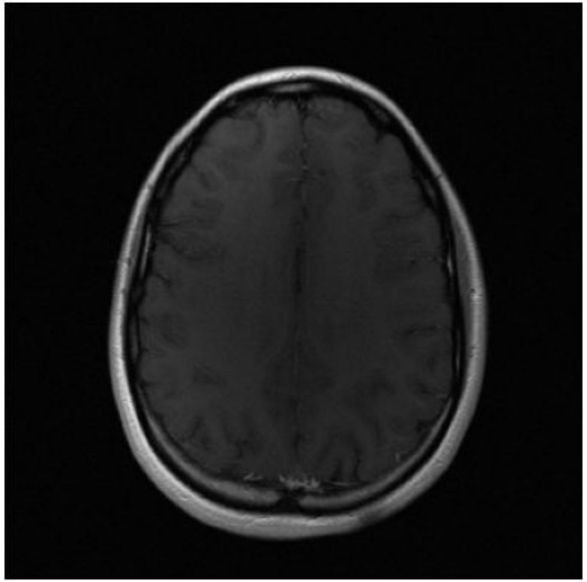
Qualitative validation results - T1POST - R4

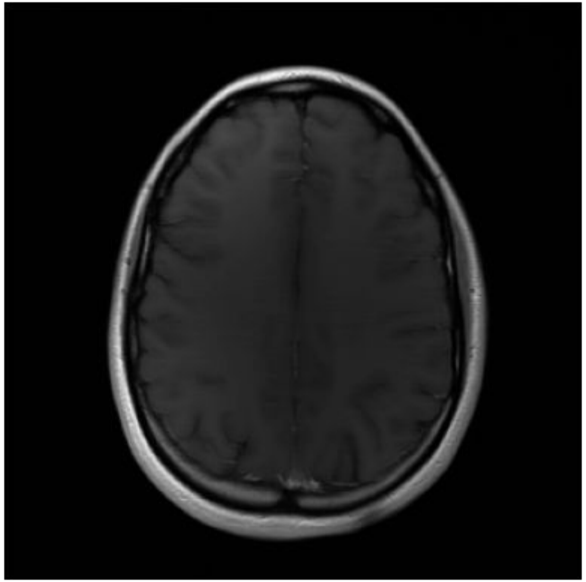


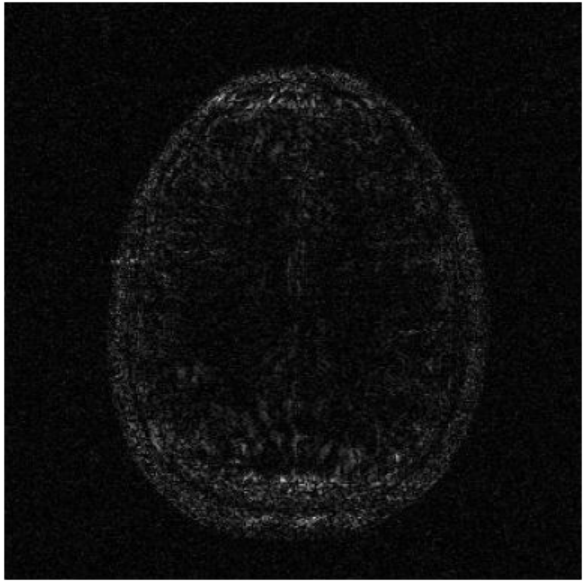




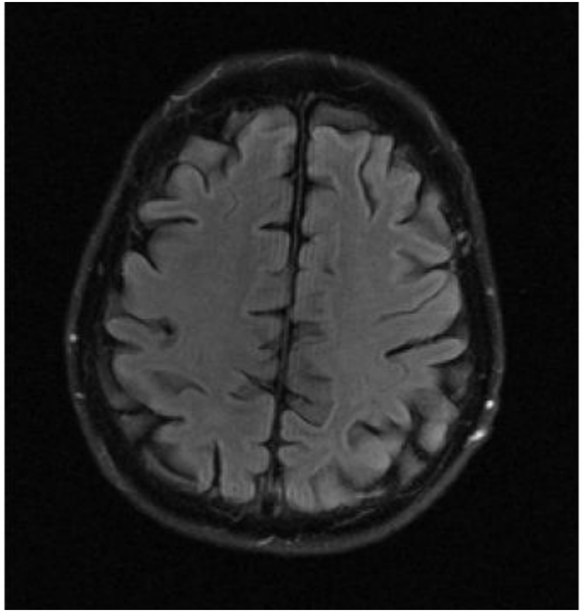
Qualitative validation results - T1POST - R8

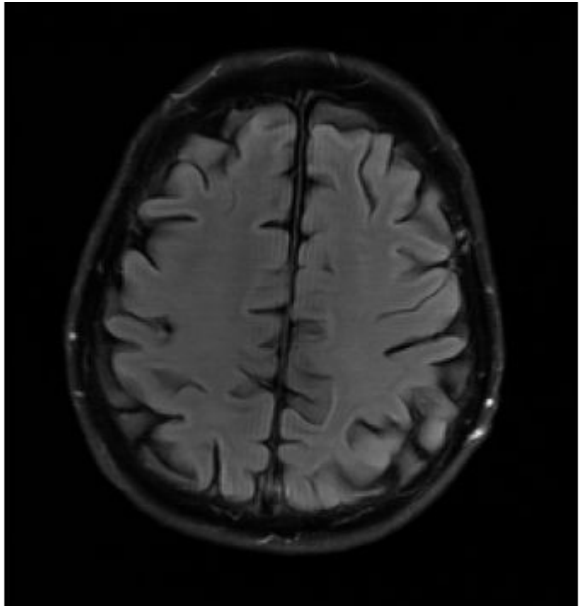


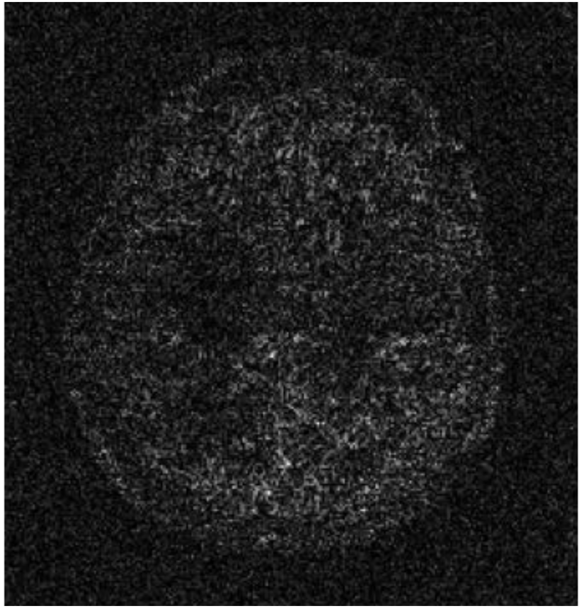




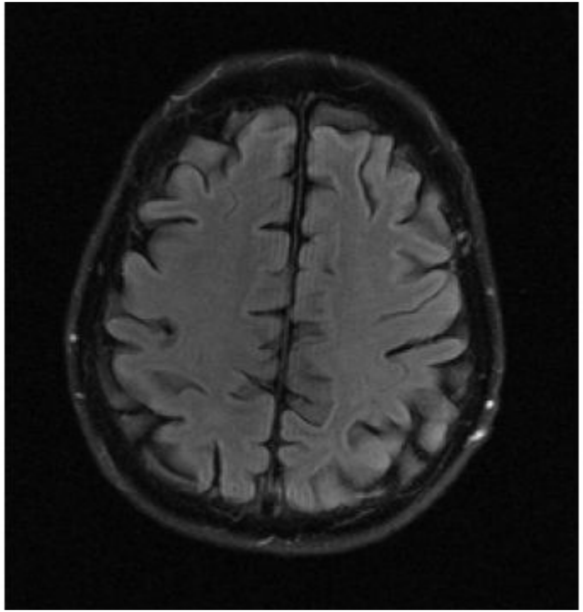
Qualitative validation results - FLAIR - R4

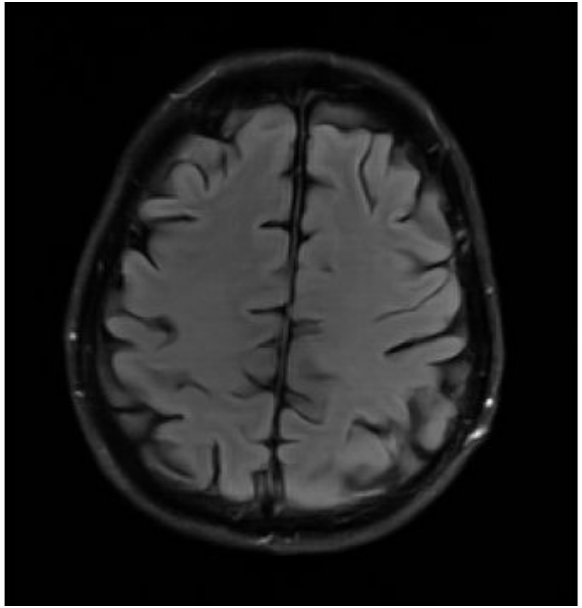


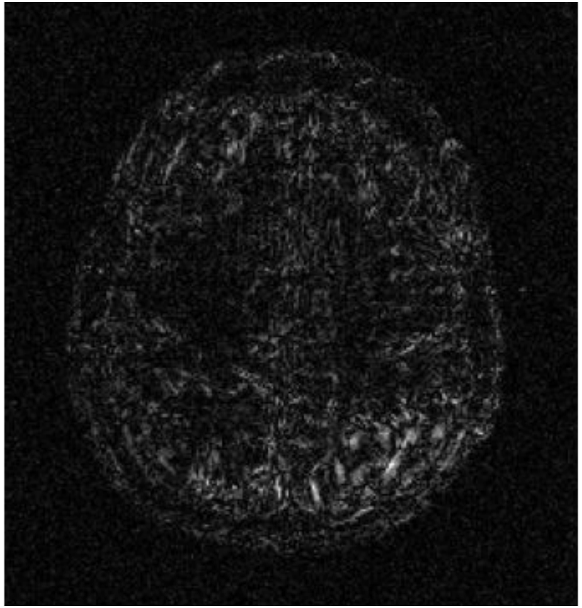




Qualitative validation results - FLAIR - R8







Why I am only second?

What winners did better than me:

- Unclear feature multi-domain learning
- 3D Post-processing (main network is 2D)
- Distributed training (4 GPUs)

Conclusion and future steps

Conclusion

- XPDNet is more than just **prior learning**: it can learn the **unrolled optim.** structure, the **measurements operator** and the **noise model**.
- Using this, we achieve 2nd spot in the fastMRI 2020 challenge.

Next steps

- Implement the missing bits (DC learning, post-processing, better/longer training, multi-domain learning)
- Full 3D using model parallelism
- Implicit Deep Learning

Acknowledgements and code

Code is available in TensorFlow on GitHub: github.com/zaccharieramzi/fastmri-reproducible-benchmark

Thanks to the French Institute of development and resources in scientific computing (IDRIS) for allowing the use of a supercluster (Jean Zay) of V100 GPUs.

Don’t hesitate to ask me questions about Jean Zay or refer to the user’s doc!

Thank you!

Questions ?

Robustness - generalization to OOD-prospective

Brain acquired on a Siemens scanner:

- different resolution
- different orientation
- different field strength
- different AF
- presents a cerebellum
- prospective acquisition