Interview talk

Candidate	Guanxiong Luo
Title	Development of Advanced Generative Priors for MRI Reconstruction
Date	Nov 21, 2023
Place	online via Teams

Outline

- **MRI reconstruction as inverse problems**
- Bayesian MRI reconstruction using diffusion priors
- Phase augmentation for training priors
- Software
- Summary and outlook

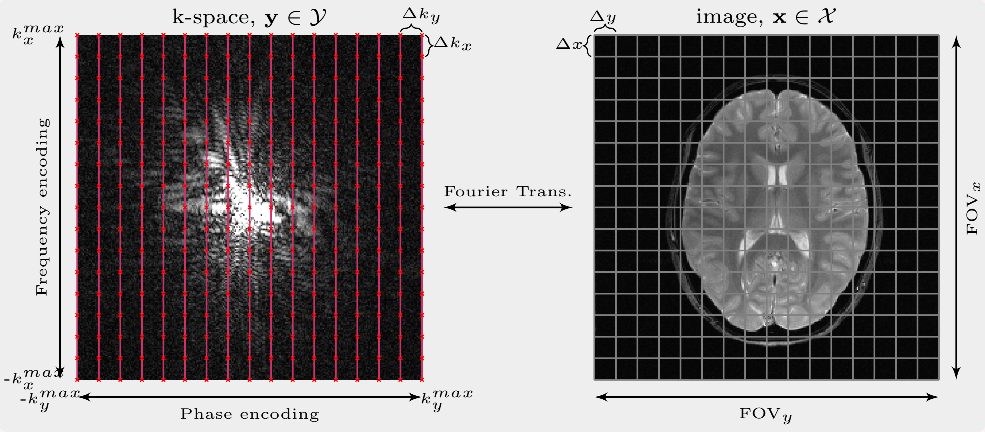
MR scans have revolutionized clinical practice in numerous ways

- non-invasive, radioactive-free
- organs, tissues and skeletal system
- multi-contrast, high-resolution
- functional imaging
- …
- intrinsically slow

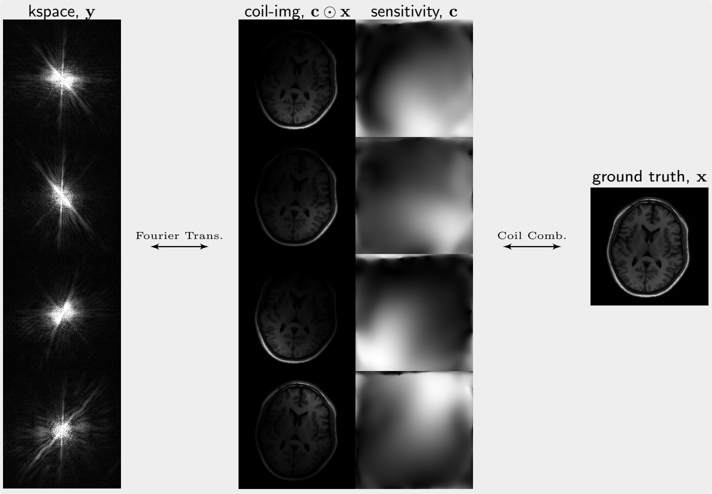
Created by DALL·E·3

The acquisition in k-space, full-sampling

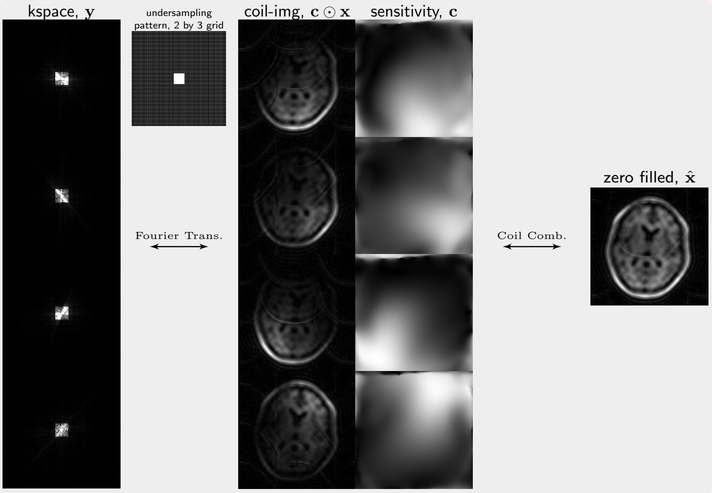
Nyquist Theorem at least requires: $\text{FOV}_x = 1/\Delta k_x,\ \Delta x = 1/(2 * k_x^{max}),\ \text{FOV}_x = \text{N} * \Delta x.$



Parallel imaging, multiple coils imaging, full-sampling



Parallel imaging, multiple coils imaging, under-sampling



Linear and non-linear reconstruction

Parallel MRI is formulated as an inverse problem

$$(\mathcal{F}_S(\mathbf{x}\cdot c_1),\cdots,\mathcal{F}_S(\mathbf{x}\cdot c_N)) = \mathbf{y}.$$

- $\mathcal{F}_S$, undersampling Fourier operator. $\mathbf{y} = (y_1,\cdots,y_N)$, measured k-space data
- $\mathbf{x}$, image. N, number of coils. $\mathbf{c} = (c_1,\cdots,c_N)$, coil sensitivities

Linear and non-linear reconstruction

Parallel MRI is formulated as an inverse problem

$$(\mathcal{F}_S(\mathbf{x} \cdot c_1), \cdots, \mathcal{F}_S(\mathbf{x} \cdot c_N)) = \mathbf{y}.$$

- $\mathcal{F}_S$, undersampling Fourier operator. $\mathbf{y} = (y_1, \cdots, y_N)$, measured k-space data
- $\mathbf{x}$, image. N, number of coils. $\mathbf{c} = (c_1, \cdots, c_N)$, coil sensitivities

With pre-computed sensitivities, the forward operator is $\mathcal{A} : \mathbf{x} \to \mathbf{y}$. The reconstruction is achieved by minimizing

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathcal{A}\mathbf{x} - \mathbf{y}\|^2 + \alpha R(\mathbf{x}) .$$

Linear and non-linear reconstruction

Parallel MRI is formulated as an inverse problem

$$(\mathcal{F}_S(\mathbf{x} \cdot c_1), \cdots, \mathcal{F}_S(\mathbf{x} \cdot c_N)) = \mathbf{y}.$$

- $\mathcal{F}_S$, undersampling Fourier operator. $\mathbf{y} = (y_1, \cdots, y_N)$, measured k-space data
- $\mathbf{x}$, image. N, number of coils. $\mathbf{c} = (c_1, \cdots, c_N)$, coil sensitivities

With pre-computed sensitivities, the forward operator is $\mathcal{A} : \mathbf{x} \to \mathbf{y}$. The reconstruction is achieved by minimizing

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathcal{A}\mathbf{x} - \mathbf{y}\|^2 + \alpha R(\mathbf{x}) \ .$$

- Compressed sensing: L1 prior in wavelet domain $R(\mathbf{x}) = \|\Phi \mathbf{x}\|_1$, and random sampling pattern.
- Coil sensitivities estimation[1]
- Joint estimation of **c** and **x** via nonlinear inversion[2]

[1] Uecker et al. MRM 2014. [2] Uecker et al. MRM 2008.

The application of deep learning in fast MRI

- Many unrolling neural networks[1,2,3] aim to predict the reconstruction from the undersampled k-space $\mathbf{y}$.
- A dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, \cdots, N\}$, consisting of paired undersampled k-space data $\mathbf{y}$ and ground truth images $\mathbf{x}$.
- The network $F_\theta^\dagger$ parametrized by θ is trained by minimizing the error

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_D \left[\| F_\theta^\dagger(\mathbf{y}) - \mathbf{x} \|_2^2 \right] . \tag{1}$$

The application of deep learning in fast MRI

- Many unrolling neural networks[1,2,3] aim to predict the reconstruction from the undersampled k-space $\mathbf{y}$.
- A dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, \cdots, N\}$, consisting of paired undersampled k-space data $\mathbf{y}$ and ground truth images $\mathbf{x}$.
- The network $F_\theta^\dagger$ parametrized by θ is trained by minimizing the error

$$\hat{\theta} = \arg\min_\theta \mathbb{E}_D \left[\| F_\theta^\dagger(\mathbf{y}) - \mathbf{x} \|_2^2 \right] . \tag{1}$$

- The network depends on the pre-definition of the forward operator $\mathcal{A}$

[1] Yang et al. NIPS 2016. [2] Hammernik et al. MRM 2018. [3] Aggarwal et al. TMI(2019).

Challenges 1: Generalizability

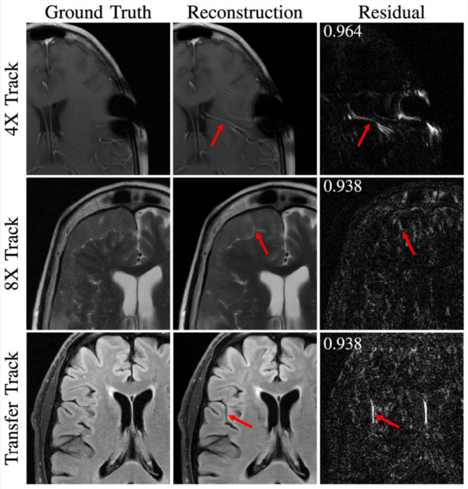
Study[1] reported that tiny perturbations, contrast changes, and sampling pattern variations can fail most unrolling neural networks $F_\theta^\dagger$.

[1] Knoll et al. MRM 2019.

Challenges 2: Hallucination

Studies[1,2] reported hallucinations generated by an unrolling neural network $F_\theta^\dagger$.

- (top) a false-generated vessel
- (middle) a linear bright signal mimicking a cleft of cerebrospinal fluid
- (bottom) a false-generated sulcus or prominent vessel.



[1] Muckley et al. IEEE TMI 2021. [2] Bhadra et al. IEEE TMI 2021.

Challenges 3: Data availability

- FastMRI[1] includes full-sampled k-space data and four types of contrast. Limited organ types
- but to collect more is expensive and takes time

[1] Zbontar et al. Radiology 2020.

Outline

- MRI reconstruction as inverse problems
- **Bayesian MRI reconstruction using diffusion priors**
- Phase augmentation for training priors
- Software
- Summary and outlook

Reconstruction from Bayesian Perspective

- **posterior**: the image **x** conditioned on the acquired k-space data **y** is

$$\underbrace{p(\mathbf{x}|\mathbf{y})}_{\text{posterior}} = \frac{p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \propto \underbrace{p(\mathbf{y} \mid \mathbf{x})}_{\text{likelihood}} \cdot \underbrace{p(\mathbf{x})}_{\text{prior}}, \tag{2}$$

- **likelihood**: $p(\mathbf{y} \mid \mathbf{x})$ → the probability of the acquired k-space data **y** for a given image **x**
- **prior**: $p(\mathbf{x})$ → the prior model of image **x**

The likelihood for k-space

- The likelihood $p(\mathbf{y}|\mathbf{x})$ for observing the $\mathbf{y}$ determined by $\mathbf{y} = \mathcal{A}\mathbf{x} + \boldsymbol{\eta}$
- The noise $\boldsymbol{\eta}$ is zero mean and normal distributed with covariance matrix $\sigma_\eta^2 \mathbf{I}$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{CN}(\mathbf{y}; \mathcal{A}\mathbf{x}, \sigma_\eta^2 \mathbf{I}). \tag{3}$$

then, we have

$$p(\mathbf{x}|\mathbf{y}) \propto \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp(-\frac{1}{2\sigma_\eta^2}\|\mathbf{y} - \mathcal{A}\mathbf{x}\|_2^2) \cdot p(\mathbf{x}) \tag{4}$$

$$\log p(\mathbf{x}|\mathbf{y}) \propto -\frac{1}{2\sigma_\eta^2}\|\mathbf{y} - \mathcal{A}\mathbf{x}\|_2^2 + \underbrace{\log p(\mathbf{x})}_{\text{prior}} \tag{5}$$

Non-learned and learned priors

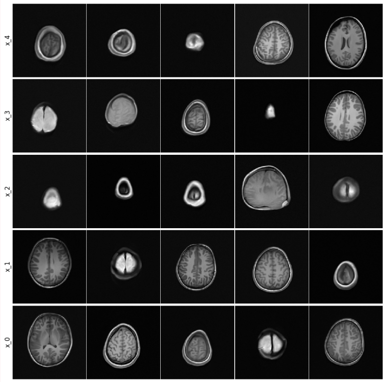
- Laplace distribution, $p(\mathbf{x} \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|\mathbf{x}-\mu\mathbf{I}|}{b}\right)$, can promote sparsity in the wavelet domain. (Compressed sensing)
- We can learn an empirical prior $p_\theta(\mathbf{x})$ from the i.i.d. samples $D_n = \{\mathbf{x}^1, ..., \mathbf{x}^{(n)}\}$, using generative modeling.

Non-learned and learned priors

- Laplace distribution, $p(\mathbf{x} \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|\mathbf{x}-\mu\mathbf{I}|}{b}\right)$, can promote sparsity in the wavelet domain. (Compressed sensing)
- We can learn an empirical prior $p_\theta(\mathbf{x})$ from the i.i.d. samples $D_n = \{\mathbf{x}^1, ..., \mathbf{x}^{(n)}\}$, using generative modeling.
- Generative priors for MRI reconstruction

Generative model/prior

A generative model is a statistical model $p_\theta(X = \mathbf{x})$ that is used to approximate distribution of a high-dimensional random variable $p_D(X = \mathbf{x})$ from an observed dataset D.



MAP and MMSE

With the posterior $p(\mathbf{x}|\mathbf{y})$, the reconstruction can be achieved by

- Maximize posterior (**MAP**)

$$\hat{\mathbf{x}}_{\mathrm{MAP}}(\mathbf{y}) = \arg\max_{\mathbf{x}} \ \log p(\mathbf{x}|\mathbf{y})$$

- Minimum mean square error (**MMSE**)

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} \int \|\tilde{\mathbf{x}} - \mathbf{x}\|^2 p(\mathbf{x}|\mathbf{y}) d\mathbf{x} = \mathbb{E}[x|y]$$

- Explore the posterior to estimate the uncertainty of reconstruction

Sample the posterior

- We generate samples from $p(\mathbf{x}|\mathbf{y})$[1,2,3], using **Langevin dynamics**

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^{k} + \frac{\gamma}{2}\nabla_{\mathbf{x}} \log p_{\theta}(\mathbf{x}^{k} \mid \mathbf{y}) + \sqrt{\gamma}\mathbf{z}, \quad z \sim \mathcal{CN}(0, \mathbf{I})\,. \tag{6}$$

Sample the posterior

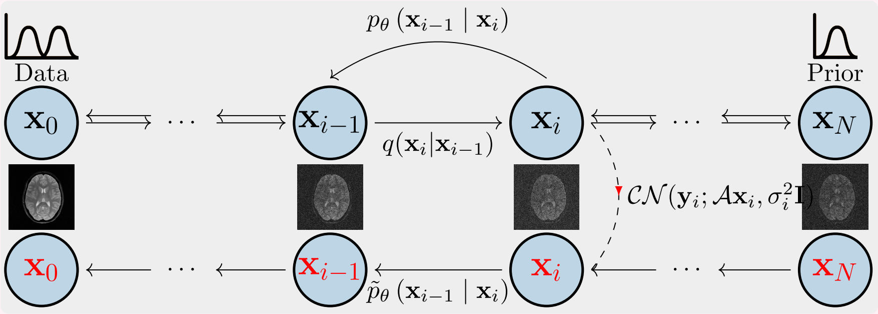
- We generate samples from $p(\mathbf{x}|\mathbf{y})$[1,2,3], using **Langevin dynamics**

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^{k} + \frac{\gamma}{2}\nabla_{\mathbf{x}} \log p_{\theta}(\mathbf{x}^{k} \mid \mathbf{y}) + \sqrt{\gamma}\mathbf{z}, \quad z \sim \mathcal{CN}(0, \mathbf{I})\,. \tag{6}$$

- The variance computed pixelwise over samples is an indicator of **uncertainty**.
- The mean computed pixelwise over samples is the estimation for $\mathbf{x}_{\text{MMSE}}$.
- The prior distribution of image is difficult to determine.

[1] Jalal et al. NIPS (2021). [2] Chung et al. MIA (2022). [3] Luo et al. ISMRM (2022).

Denoising score matching



- $\mathbf{x}_i = \mathbf{x}_0 + \sigma_i \mathbf{z}$. Learn **score function** $\nabla_{\mathbf{x}_i} \log p_\theta^{data}(\mathbf{x}_i)$ at different noise scales σ_i, using a deep neural network[1,2,3]
- Incorporate the k-space likelihood term

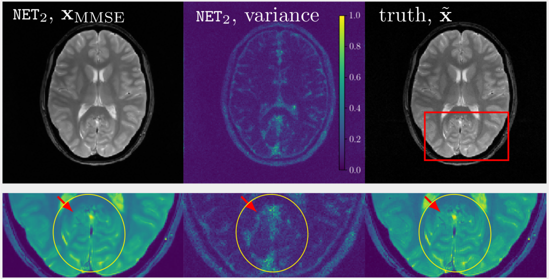
[1]Sohl-Dickstein et al. ICML (2015). [2]Song et al. NIPS (2019). [3]Song et al. ICLR (2021).

Experiments

- $\mathbf{x}_{\mathrm{MMSE}}$ and variance
- Uncertainty estimation
- Transferability of a learned prior

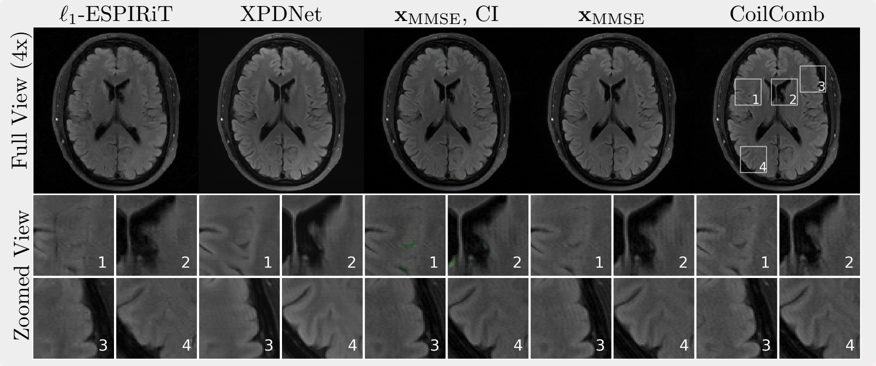
MMSE and variance map

- The undersampling pattern covers 11.8% k-space using variable density poisson disc. The central 20×20 region is fully acquired.
- 10 images were drawn from $p(\mathbf{x}|\mathbf{y})$.



Uncertainty and hallucination

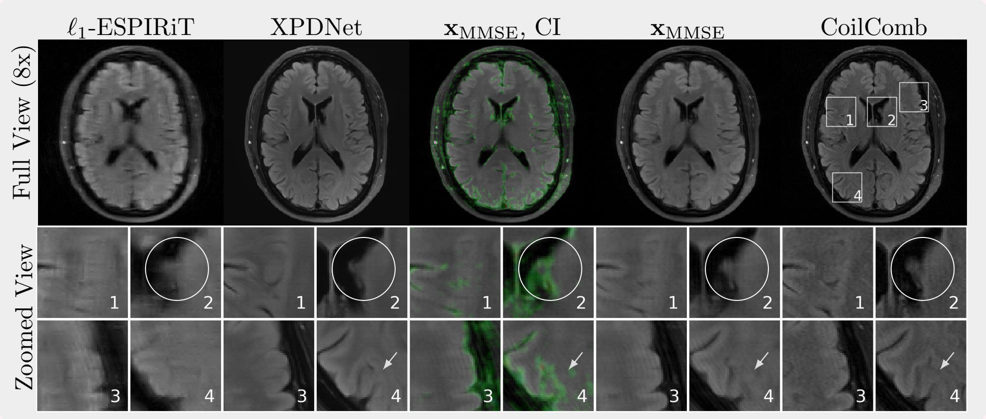
- 4x undersampling along phase direction in k-space.
- Reconstructions are ℓ_1-wavelet, XPDNet[1], $\mathbf{x}_{\text{MMSE}}$ highlighted with thresholding credibility interval (CI), $\mathbf{x}_{\text{MMSE}}$ and a fully-sampled coil-combined image (ground truth).



[1] Ramzi et al. arXiv (2020)

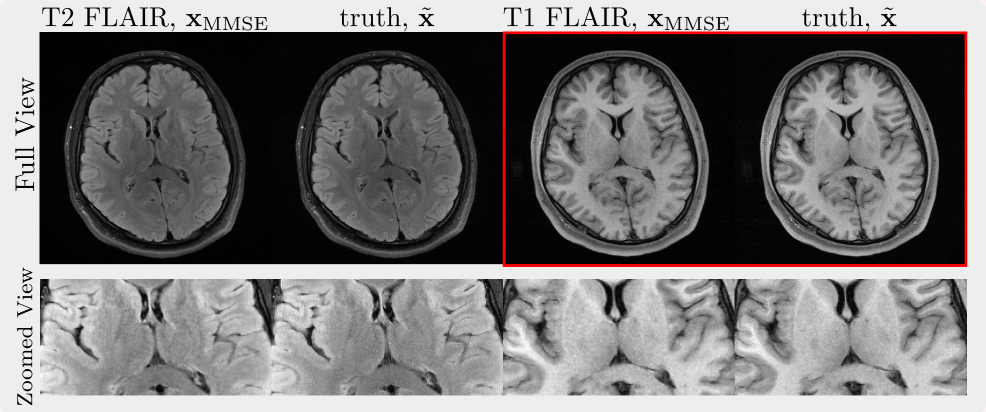
Uncertainty and hallucination

- Hallucinations appear when using 8-fold acceleration and are highlighted with the CI after thresholding. Selected regions of interests are presented in a zoomed view.



Transferability of learned information

- Reconstruction of T2 and T1 FLAIR images with a prior trained on T2 FLAIR.
- Use a Poisson-disk pattern of 8.2x undersampling in k-space, 320x320.



Recap 1

- We combine concepts from machine learning, Bayesian inference and image reconstruction.
- The image reconstruction is realized by drawing samples from the posterior term $p(\mathbf{x}|\mathbf{y})$ using a learned prior.
- This method provides a minimum mean square reconstruction and uncertainty estimation.
- This method shows good performance and transferability to different contrasts and sampling patterns.

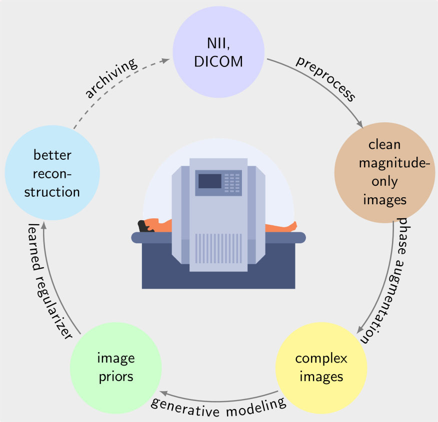
[1] G. Luo, M. Blumenthal, M. Heide, M. Uecker. "Bayesian MRI reconstruction with joint uncertainty estimation using diffusion models". Magn Reson Med. March 2023;1-17. DOI: 10.1002/mrm.29624

Outline

- MRI reconstruction as inverse problems
- Bayesian MRI reconstruction using diffusion priors
- **Phase augmentation for training priors**
- Software
- Summary and outlook

Construct dataset from magnitude-only images

- MR images are complex-valued
- Easy to get magnitude-only images in hospital
- Especially pathological data



Phase augmentation using Langevin sampling

- **prior**: a prior for complex-valued images $p(\mathbf{x})$.
- **likelihood**: the likelihood term of the magnitude

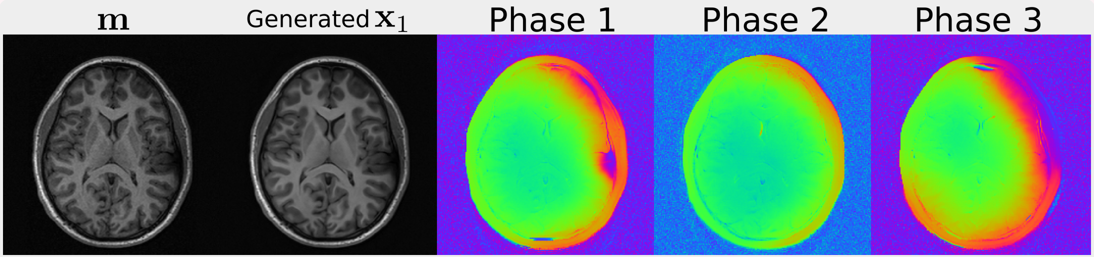
$$p(\mathbf{m}|\mathbf{x}) \propto \exp\left(-\epsilon\left\|\mathbf{m} - \sqrt{\mathbf{x}_r^2 + \mathbf{x}_i^2}\right\|_2^2\right).$$

- **posterior**: the complex image is proportional to

$$p(\mathbf{x}|\mathbf{m}) \propto p(\mathbf{x}) \cdot p(\mathbf{m}|\mathbf{x}) \ .$$

Examples of generated phase maps

- The magnitude and its corresponding phase maps of the generated complex-valued images



MRI Image Priors

When training a prior,

- what is the effect of phase?
- how does the dataset size affect the prior?

MRI Image Priors

When training a prior,

- what is the effect of phase?
- how does the dataset size affect the prior?

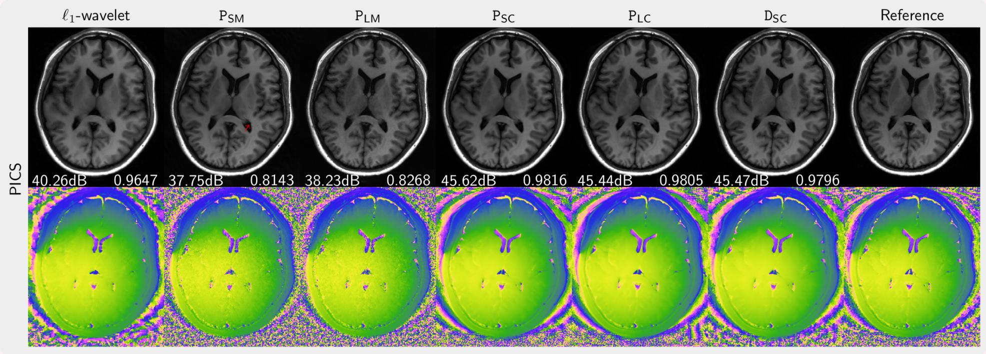
Table: Datasets and computational resources used to train the six different priors used in this work.

Prior		Model	Phase	Nr. of Images	MR Contrasts	GPUs	Parameters	Time × epochs
P_{SC}	(small, complex)	PixelCNN[1]	preserved	1000	T_1, T_2, T_2-FLAIR, T_2^*	4×A100, 80G	~22M	~40s × 500
P_{SM}	(small, magnitude)	PixelCNN	not available	1000	T_1, T_2, T_2-FLAIR, T_2^*	4×V100, 32G	~22M	~144s × 500
P_{LM}	(large, magnitude)	PixelCNN	not available	23078	MPRAGE	4×A100, 80G	~22M	~748s × 100
P_{LC}	(large, complex)	PixelCNN	generated	23078	MPRAGE	3×A100, 80G	~22M	~1058s × 100
D_{SC}	(SMLD, complex)	Diffusion	generated	79058	MPRAGE	4×A100, 80G	~8M	~2330s × 50
D_{PC}	(DDPM, complex)	Diffusion	generated	79058	MPRAGE	8×V100, 32G	~8M	~1430s × 200

[1] Salimans et al. ICLR 2016.

Influence of phase maps

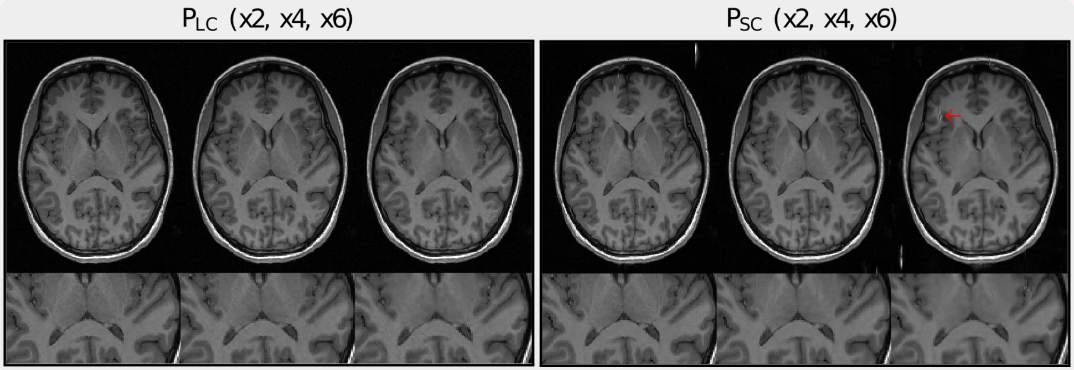
learned priors	P_{SC}	P_{SM}	P_{LM}	P_{LC}	D_{SC}
Phase	preserved	not available	not available	generated	generated
Nr. Images	1000	1000	23078	23078	79058



8.2x undersampling in k-space with Poisson disc

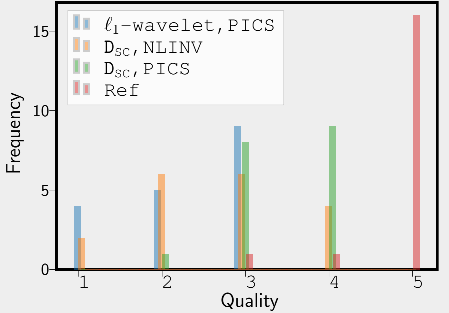
Small prior P_{SC} vs Large prior P_{LC}

- MPRAGE (Magnetization Prepared-RApid Gradient Echo) sequence
- 2-fold acceleration with grappa and 4/5 partial parallel Fourier imaging
- Dimensions (256, 256, 224) and voxel sizes (1mm, 1mm, 1mm).

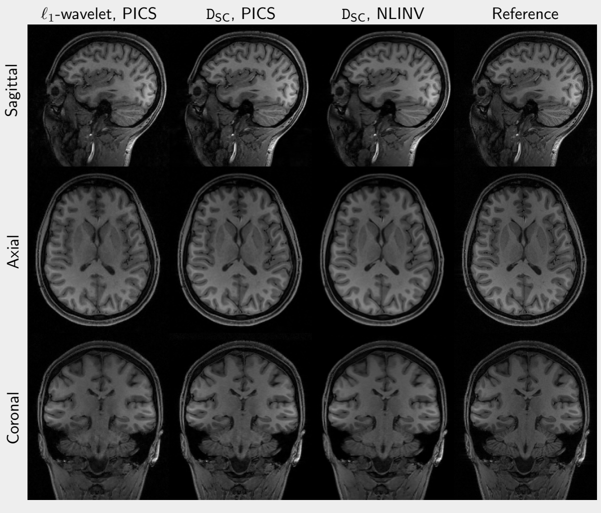


Clinical results

- 5-Excellent, 1-Bad
- 3D acquisition using Turbo-FLASH
- 8.2x undersampling



Collaborative work with Prof. Niels Focke at UMG



Recap 2

- We demonstrate that priors trained on complex images are superior to priors trained on magnitude-only images
- We leverage a diffusion model trained on a small dataset of complex images to augment a much larger dataset
- We integrate priors at regularization term into reconstruction.

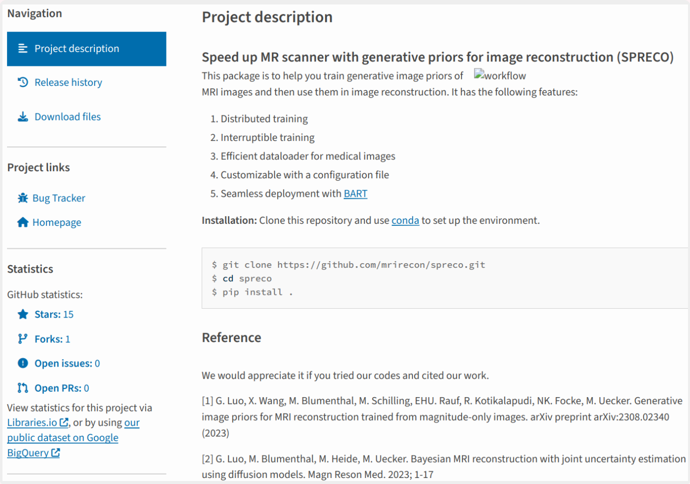
G. Luo, X. Wang, M. Blumenthal, M. Schilling, EHU Rauf, R. Kotikalapudi, N. K. Focke, M. Uecker. "Generative Image Priors for MRI Reconstruction Constructed from Magnitude-Only Images Using Phase Augmentation" arXiv: 2308.02340

Outline

- MRI reconstruction as inverse problems
- Bayesian MRI reconstruction using diffusion priors
- Phase augmentation for training priors
- **Software**
- Summary and outlook

A tool to develop priors: spreco

- spreco[1]
- image-priors[2]



Navigation

Project description

Release history

Download files

Project links

Bug Tracker

Homepage

Statistics

GitHub statistics:

Stars: 15

Forks: 1

Open issues: 0

Open PRs: 0

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Project description

Speed up MR scanner with generative priors for image reconstruction (SPRECO)

This package is to help you train generative image priors of ![workflow] MRI images and then use them in image reconstruction. It has the following features:

1. Distributed training
2. Interruptible training
3. Efficient dataloader for medical images
4. Customizable with a configuration file
5. Seamless deployment with BART

Installation: Clone this repository and use conda to set up the environment.

```
$ git clone https://github.com/mrirecon/spreco.git
$ cd spreco
$ pip install .
```

Reference

We would appreciate it if you tried our codes and cited our work.

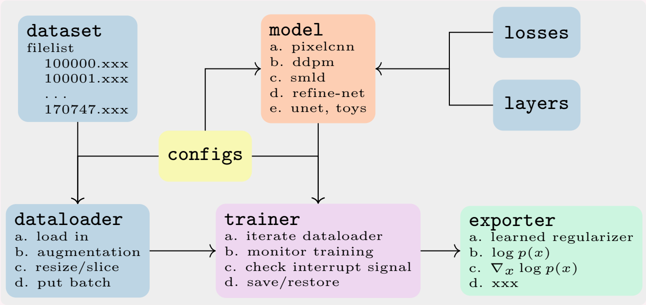
[1] G. Luo, X. Wang, M. Blumenthal, M. Schilling, EHU. Rauf, R. Kotikalapudi, NK. Focke, M. Uecker. Generative image priors for MRI reconstruction trained from magnitude-only images. arXiv preprint arXiv:2308.02340 (2023)

[2] G. Luo, M. Blumenthal, M. Heide, M. Uecker. Bayesian MRI reconstruction with joint uncertainty estimation using diffusion models. Magn Reson Med. 2023; 1-17

[1] https://github.com/mrirecon/spreco [2] https://github.com/mrirecon/image-priors

Overview

- The loss and layer functions are used to create models.
- The trainer is fed by the dataloader and trains the model on multiple GPUs.
- The exporter is used to customize trained models for deployment.



It is implemented with : 1) tensorflow; 2) numpy; 3) pyyaml; 4)matplotlib; 5)scikit-image; 6) tqdm.

Hands-on

1. Generative prior trained on a large public dataset
   Open in Colab
2. Sample the posterior
   Open in Colab
3. Train an image prior
   Open in Colab
4. Using Prior with BART
   Open in Colab

Summary

- What are the advantages of a prior?

Summary

- What are the advantages of a prior? Decoupled from the forward operator, which leads to better generalizability to many acquisition scenarios.
- How to use a prior for reconstruction?

Summary

- What are the advantages of a prior? Decoupled from the forward operator, which leads to better generalizability to many acquisition scenarios.
- How to use a prior for reconstruction? Sample the posterior or use it as regularization term, which permits uncertainty estimation and flexibility.
- How to get more data for training?

Summary

- What are the advantages of a prior? Decoupled from the forward operator, which leads to better generalizability to many acquisition scenarios.
- How to use a prior for reconstruction? Sample the posterior or use it as regularization term, which permits uncertainty estimation and flexibility.
- How to get more data for training? Phase augmentation of a large magnitude-only dataset using diffusion priors trained on small complex-valued images.
- How to develop and deploy a prior? spreco

Outlook

- How to determine the performance bound of a prior?
- How does it relate to the uncertainties of the reconstructed image?
- How many samples need to be acquired to keep the uncertainty within an acceptable limit?
- How effectively can the prior information transfer to a broader range of scenarios?

a figure for outlook a backup for pics and nlinv, phase maps

Q&A

https://ggluo.github.io/assets/pdf/phd_thesis_draft.pdf

Algorithm

Algorithm 1 Sampling the posterior via Markov chain Monte Carlo

Give the acquired k-space $\mathbf{y}_0$
Construct the forward operator $\mathcal{A}$ with sampling pattern $\mathcal{P}$ and coil sensitivities $\mathcal{S}$
Set the Langevin steps K, Langevin step coefficient β, for the tuning factor λ, the start noise level index N.
Initialize $\mathbf{x}_{\mathrm{N}}^0$ with samples from a uniform distribution $\mathcal{U}_{[-1,1]}$
for i in $\{\mathrm{N}-1, \cdots, 0\}$ **do**
    Draw samples from $\tilde{p}(\mathbf{x}_i \mid \mathbf{x}_{i+1})$ by running K Langevin steps.
end for

Network, Data, Training

- Refine-Net. 1. conditional instance normalization layer 2. Gaussian position embedding.
- Self-attention modules are added to model high resolution images (320x320).
- Small dataset (1300 images) is used for development. FastMRI dataset is used for performance evaluation.
- $\mathtt{NET}_1$ and $\mathtt{NET}_2$ are trained with small dataset. $\mathtt{NET}_3$ is trained with fastMRI dataset
- $\mathtt{NET}_1$ uses conditional instance normalization layer (10). $\mathtt{NET}_2$ and $\mathtt{NET}_3$ use Gaussian position embedding.
- MCMC sampling algorithm is implemented with TensorFlow and Numpy.

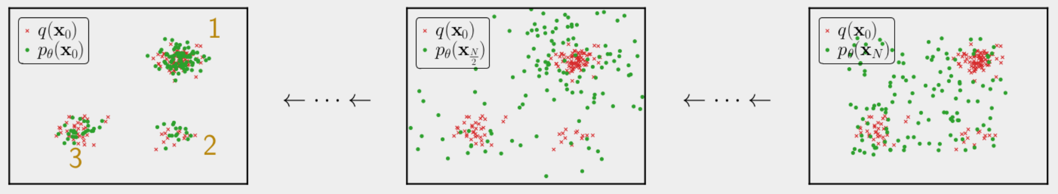
Generate samples

For a multi-modal distribution, use annealed Langevin dynamics. For σ_i in $\{\sigma_N, ..., \sigma_1\}$,

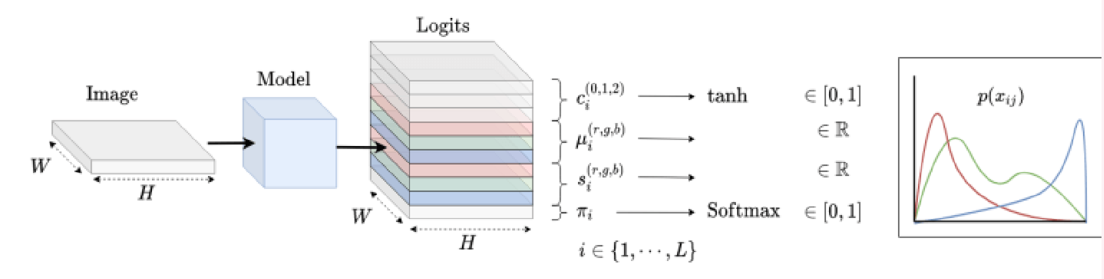
$$\mathbf{x}_i^k = \mathbf{x}_i^{k-1} - \frac{\lambda}{2} \nabla_{\mathbf{x}} \log p(\mathbf{x}_i^{k-1}) + \omega_i^k$$

A mixture of bivariate Gaussian as prior distribution

- $p(\mathbf{x}) = \sum_{i=1}^{K} \phi_i \mathcal{CN}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$
- ϕ_i is the mixture indicator, $\phi_i = \{0.7, 0.2, 0.1\}$
- $\boldsymbol{\mu_i} = \{[5, 5], [-5, -5], [5, -5]\}$, $\boldsymbol{\Sigma_i} = \mathbf{I}$.



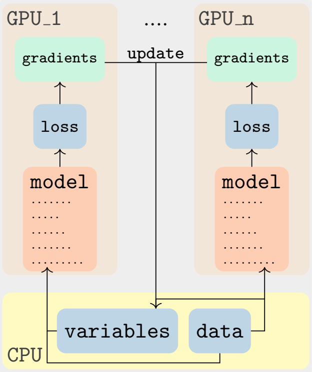
PixelCNN

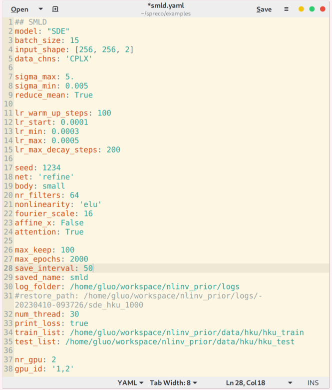


- $\pi_i \in (0,1)$ the probability of each logistic distribution
- $\mu_{i,j} \in (-\infty, \infty)$ the centre of the ith logistic distribution for the jth color
- $s_{i,j} \in (0, +\infty)$ the width of the ith logistic distribution for the jth color
- $c_{k,i}$ the $k = 0, 1, 2$ linear coefficients for $r \to g$ and $r, g \to b$, for each logistic function i

Spreco

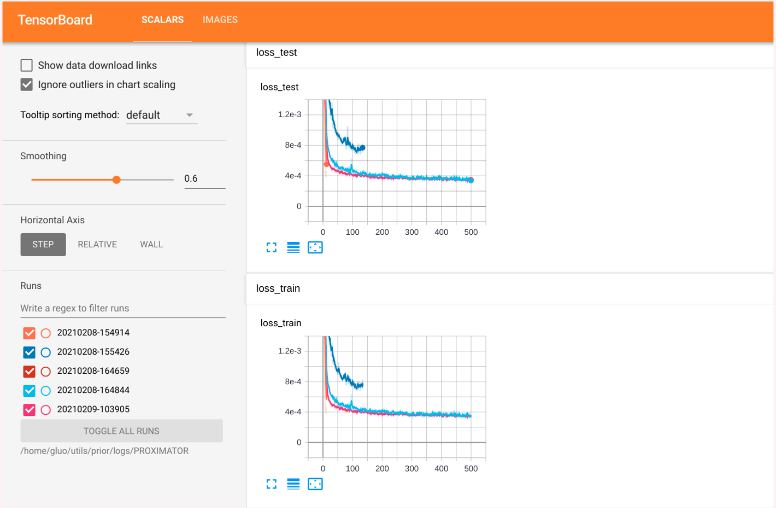
- The trainable variables are shared across GPUs
- Dataloader feeds data to every GPU worker.
- Config file for training models






```yaml
## SMLD
model: "SDE"
batch_size: 15
input_shape: [256, 256, 2]
data_chns: 'CPLX'

sigma_max: 5.
sigma_min: 0.005
reduce_mean: True

lr_warm_up_steps: 100
lr_start: 0.0001
lr_min: 0.0003
lr_max: 0.0005
lr_max_decay_steps: 200

seed: 1234
net: 'refine'
body: small
nr_filters: 64
nonlinearity: 'elu'
fourier_scale: 16
affine_x: False
attention: True

max_keep: 100
max_epochs: 2000
save_interval: 50
saved_name: smld
log_folder: /home/gluo/workspace/nlinv_prior/logs
#restore_path: /home/gluo/workspace/nlinv_prior/logs/-20230410-093726/sde_hku_1000
num_thread: 30
print_loss: true
train_list: /home/gluo/workspace/nlinv_prior/data/hku/hku_train
test_list: /home/gluo/workspace/nlinv_prior/data/hku/hku_test

nr_gpu: 2
gpu_id: '1,2'
```

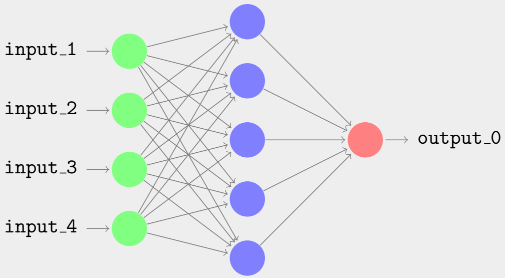
Spreco


TensorBoard
SCALARS
IMAGES
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
20210208-154914
20210208-155426
20210208-164659
20210208-164844
20210209-103905
TOGGLE ALL RUNS
/home/gluo/utils/prior/logs/PROXIMATOR
loss_test
loss_test
1.2e-3
8e-4
4e-4
0
0
100
200
300
400
500
loss_train
loss_train
1.2e-3
8e-4
4e-4
0
0
100
200
300
400
500

BART with tensorflow computation graph

- Export the trained model with customized labels for inputs $\mathbf{x}$ and outputs $\mathbf{y} = \mathtt{Net}_\theta(\mathbf{x})$;
- Initialize an exported graph, the restoration of a saved model using C API;
- Wrap the exported computation graph into BART's non-linear operator (nlop)

(a) Export the trained model as computation graph



(b) Use the graph as regularization in BART

```
$ bart pics -R TF:<graph_path>:λ <kspace> <coils> <reco>
```