JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Ground-Truth Free Meta-Learning for Deep Compressive Sampling

Xinran Qin

Yuhui Quan

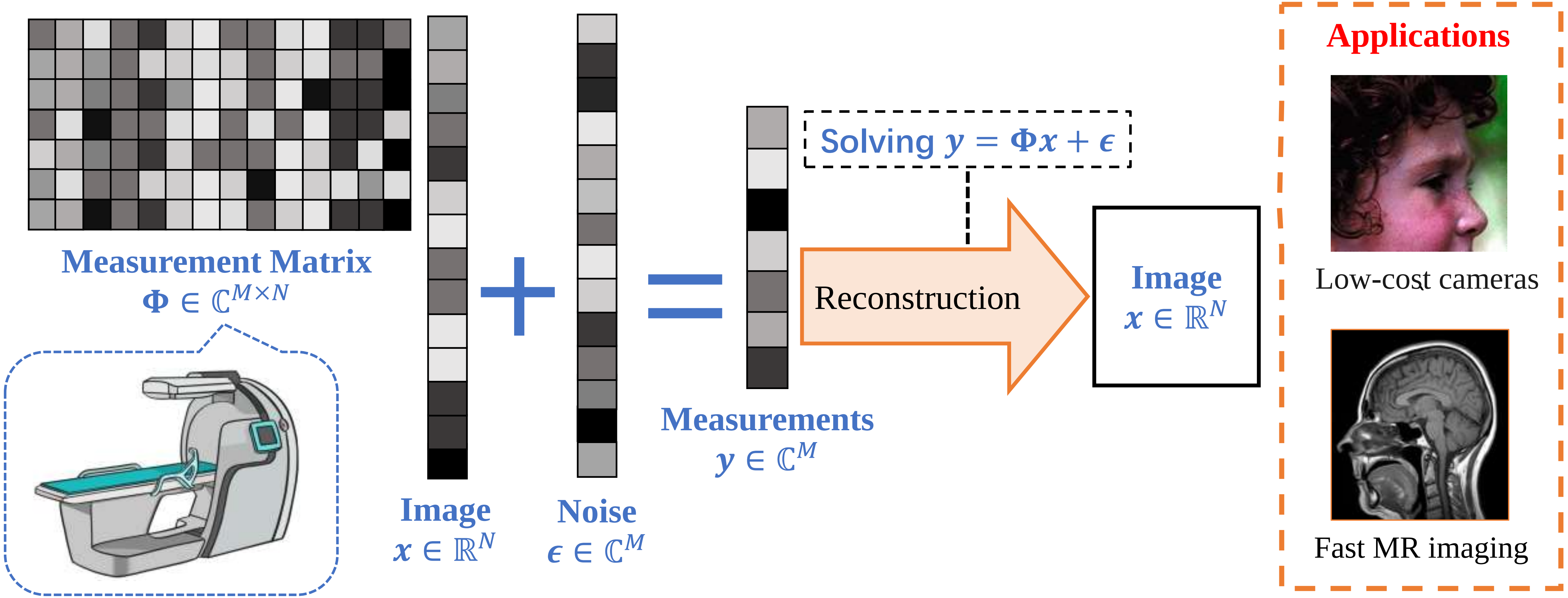
School of Computer Science and Engineering
South China University of Technology

Tongyao Pang

Hui Ji

Department of Mathematics
National University of Singapore

Compressive Sampling (CS) in Imaging



Motivation

Unsupervised External Learning (End2End DNNs)

(+) Exploiting large datasets

(+) Fast inference

(-) Non-adaptivity to test samples

Test-Time Internal Learning (Untrained DNNs)

(+) Sample-specific learning

(+) No dataset bias and OOD issues

(-) High-cost sample-wise fitting

Unsupervised (GT-Free) Meta-Learning for CS-Based Image Reconstruction

Framework

External

Training data $\mathbb{D} \coloneqq \{\boldsymbol{y}_j, \boldsymbol{\Phi}\}_j$

Model: $\mathcal{F}_{\boldsymbol{\omega}}$ (initial para. $\boldsymbol{\omega}$)

$\mathcal{F}_{\boldsymbol{\omega}} \xrightarrow{\textit{Meta Learning}} \mathcal{F}_{\widetilde{\boldsymbol{\omega}}}$

Internal

Test Sample $\boldsymbol{y}^*$

Meta-trained model: $\mathcal{F}_{\widetilde{\boldsymbol{\omega}}}$

$\xrightarrow{\textit{Model Adaption}} \mathcal{F}_{\boldsymbol{\omega}^*}$

Keys

Range-nullspace decommission $\mathbb{C}^N = \mathrm{Range}(\boldsymbol{\Phi}^{\mathrm{H}}) \oplus \mathrm{Null}(\boldsymbol{\Phi})$

- How to suppress measurement noise on $\mathrm{Range}(\boldsymbol{\Phi}^{\mathrm{H}})$ during training/adaption?
- How to learn reconstruction *w.r.t.* $\mathrm{Null}(\boldsymbol{\Phi})$?

Loss Functions | **Meta-Learning Scheme** | **NN Design**

Approach

Improved SURE (iSURE) loss for meta-learning and adaption

iSURE-based GT-free model-agnostic meta-learning (MAML)

Nullspace-consistent model adaptation

Unrolling CNN with bias-tuning

The iSure Loss

iSURE is a noisy form of SURE to provide robust estimation in $\mathrm{Range}(\boldsymbol{\Phi}^{\mathbf{H}})$.

$$\ell^{\mathrm{SURE}}(\boldsymbol{\omega}; \boldsymbol{y}, \boldsymbol{\Phi}) := \|\boldsymbol{\Phi}\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y}, \boldsymbol{\Phi}) - \boldsymbol{y}\|_2^2 + 2\sigma \mathrm{tr}(\boldsymbol{\Phi}^+(\partial \mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y}), \boldsymbol{\Phi})/\partial \boldsymbol{y}))$$

$$\ell^{\mathrm{iSURE}}(\boldsymbol{\omega}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\epsilon}') := \|\boldsymbol{\Phi}\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}', \boldsymbol{\Phi}) - \boldsymbol{y}\|_2^2 + 2\sigma \mathrm{tr}(\boldsymbol{\Phi}^+(\partial \mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}'), \boldsymbol{\Phi})/\partial \boldsymbol{y}))$$

- Noise injection mitigates overfitting in GT-free meta-learning and model adaption, as well as allows ensemble in learning and inference.
- iSURE allows efficient gradient update, without using MCMC.

Theorem 1

Let $\boldsymbol{J}_{\boldsymbol{\omega}}$ be the Jacobian matrix w.r.t. $\boldsymbol{\omega}$, *i.e.* $\boldsymbol{J}_{\boldsymbol{\omega}}\boldsymbol{\mathcal{F}}_{\boldsymbol{\omega}} = \partial \boldsymbol{\mathcal{F}}_{\boldsymbol{\omega}}/\boldsymbol{\partial\omega}$ and $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x} + \boldsymbol{\epsilon}$. Assume $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}' \backsim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2\boldsymbol{I})$ are independent. Then, we have:

$$\nabla_{\boldsymbol{\omega}} \mathbb{E}_{\boldsymbol{y},\boldsymbol{\epsilon}'} \ell^{\mathrm{iSURE}}(\boldsymbol{\omega}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\epsilon}') = 2\mathbb{E}_{\boldsymbol{y},\boldsymbol{\epsilon}'}[\boldsymbol{J}_{\boldsymbol{\omega}}(\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}', \boldsymbol{\Phi})\boldsymbol{\Phi}^+(\boldsymbol{\Phi}\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}', \boldsymbol{\Phi}) - \boldsymbol{y} + \boldsymbol{\epsilon}')].$$

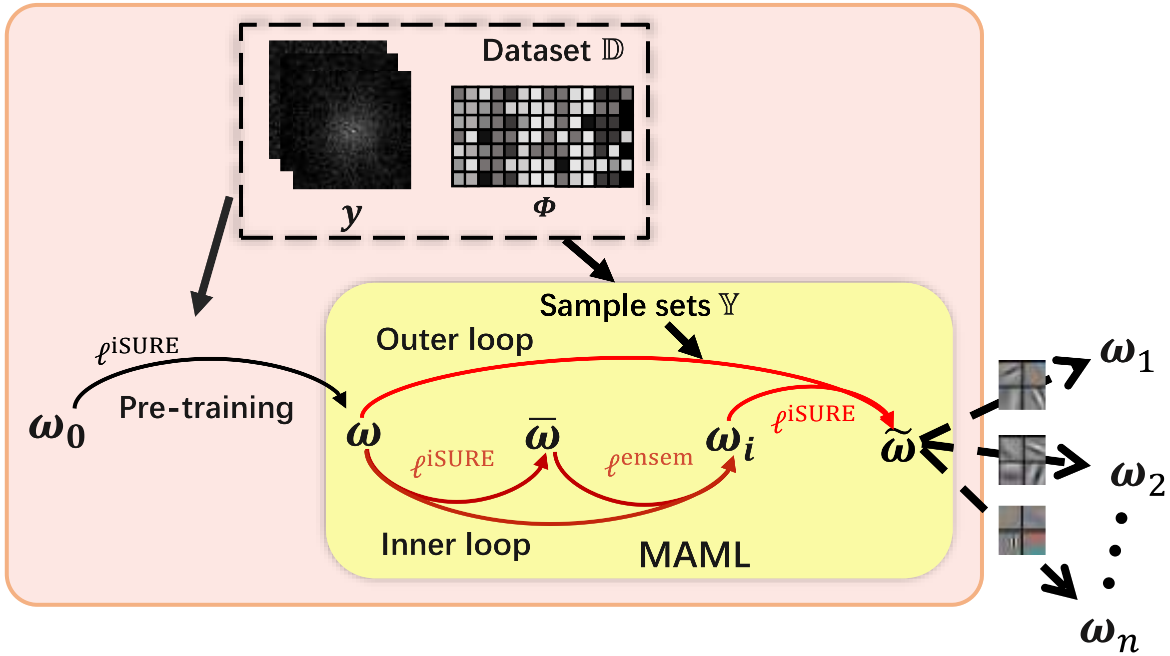
Unsupervised MAML with Improvement

Algorithm 1: GT-Free Meta-Learning for CSR

Input: Measurement dataset $\mathbb{D}$; Initial ω

Required: Learn_Rates α, β, γ; Inner_Iter_Num Q

```
Pre-training: update ω with an iSURE-based loss
while not done do                                        Outer
    Sample sets of measurements {𝕐_j}_j from 𝔻
    for each 𝕐_j do                                      Inner
        Initialize ω̄ with ω
        for q = 1, ··· , Q
            ω̄ ← ω̄ − γ∇_ω̄ L^iSURE_{𝕐_j}(ω̄)
        end
        ω_j = ω − α∇_ω L^ensem_{𝕐_j}(ω; ω̄)
    end
    Update ω ← ω − β∇_ω Σ_j L^iSURE_{𝕐_j}(ω_j)
end
```



$$\mathcal{L}_{\mathbb{Y}}^{\text{iSURE}}(\boldsymbol{\omega}) \coloneqq \mathbb{E}_{\boldsymbol{y}\in\mathbb{Y},\boldsymbol{\epsilon}'\sim\mathcal{N}(0,\sigma^2 I)}\,\ell^{\text{iSURE}}(\boldsymbol{\omega};\boldsymbol{y},\boldsymbol{\Phi},\boldsymbol{\epsilon}')$$

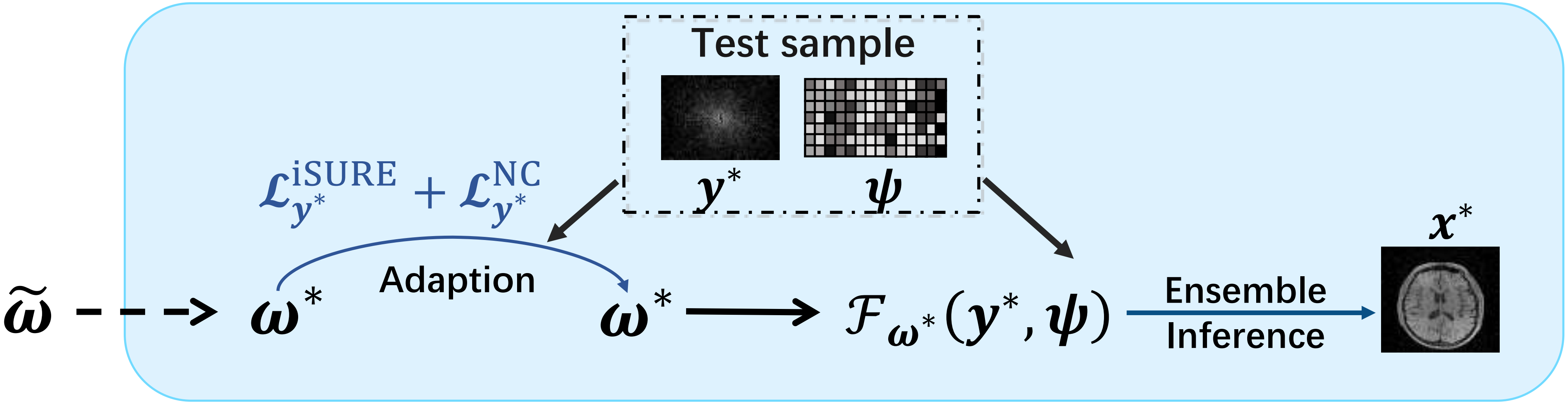
$$\mathcal{L}_{\mathbb{Y}}^{\text{ensem}}(\boldsymbol{\omega}) \coloneqq \mathbb{E}_{\boldsymbol{y}\in\mathbb{Y}}\left\|\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y},\boldsymbol{\Phi}) - 2\mathbb{E}_{\boldsymbol{\epsilon}''\sim\mathcal{N}(0,\sigma^2 I)}\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y}+\boldsymbol{\epsilon}'',\boldsymbol{\Phi})\right\|_2^2$$

- Incorporate iSURE into MAML.
- iSURE only measures errors on range space for learning.
- Inner loop with $\mathcal{L}_{\mathbb{Y}}^{\text{ensem}}$ for better addressing the ambiguity on $\text{Null}(\boldsymbol{\Phi})$.

Nullspace-Consistent (NC) Adaptation

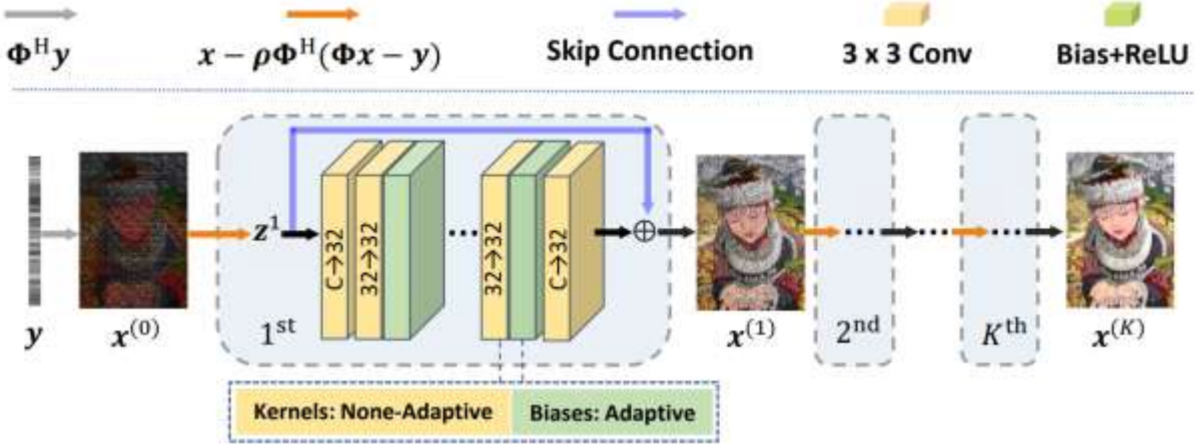
- Given a test sample $(\boldsymbol{y}^*, \boldsymbol{\psi})$, iSURE only considers reconstruction on $\mathrm{Range}(\boldsymbol{\psi}^{\mathrm{H}})$ and may bring negative effects to prediction on $\mathrm{Null}(\boldsymbol{\psi})$.
- NC adaptation mitigates negative effects in $\mathrm{Null}(\boldsymbol{\psi})$ by pulling the prediction on $\mathrm{Null}(\boldsymbol{\psi})$ back, done by the NC loss:

$$\mathcal{L}_{\boldsymbol{y}^*}^{\mathrm{NC}}(\boldsymbol{\omega}; \boldsymbol{\psi}, \boldsymbol{\omega}^*) := \left\|(\mathbf{I} - \boldsymbol{\psi}\boldsymbol{\psi}^{\dagger})(\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y}^*, \boldsymbol{\psi}) - \mathcal{F}_{\widetilde{\boldsymbol{\omega}}}(\boldsymbol{y}^*, \boldsymbol{\psi}))\right\|_2^2.$$



Unrolling CNN with Bias-Tuning

- Unroll $\boldsymbol{x}^{(k)} = \mathrm{Prox}_{\boldsymbol{\psi}}\left(\boldsymbol{x}^{(k-1)} - \rho\boldsymbol{\Phi}^{\mathrm{H}}(\boldsymbol{\Phi}\boldsymbol{x}^{(k-1)} - \boldsymbol{y})\right)$ and replace the proximal operator $\mathrm{Prox}_{\boldsymbol{\psi}}$ replaced by a sub-CNN.

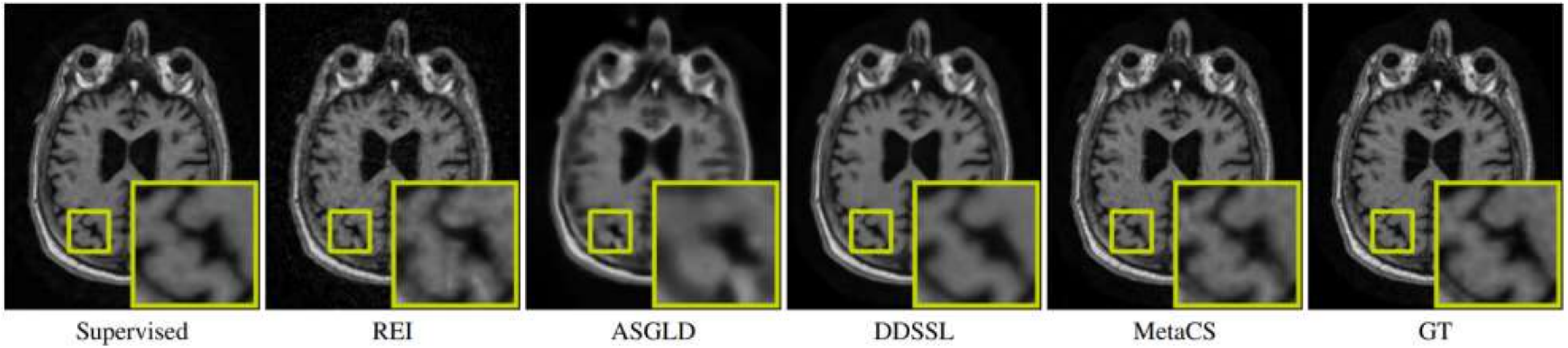


- An unrolling CNN acts as an iterative shrinkage process. The biases play a similar role to the thresholds of shrinkage, which are critical.
- **Bias-Tuning**: Only adjusting the bias parameters during adaption.

MR Image Reconstruction

Mean PSNR(dB)/SSIM of MR image reconstruction on MRI150 dataset.

CS Ratio	Noise Level	Regularized		Unsupervised	Internal		Unsupervised+Internel		Supervised	
		ZF [JMRI-01]	SparseMRI [MRM-07]	REI [CVPR-22]	BNN [ECCV-20]	ASGLD [CVPR-22]	DDSSL [ECCV-22]	MetaCS (Ours)	ADMMNet [TPAMI-19]	Supervised
20%	0	30.41/.72	35.46/.89	36.07/.90	35.54/.88	36.08/.90	36.73/.92	37.12/.93	37.17/.93	37.58/.94
	10	29.18/.64	29.74/.64	33.94/.89	35.54/.88	36.08/.90	33.79/.90	34.34/.91	34.04/.89	34.32/.90
30%	0	33.01/.80	37.72/.91	38.01/.92	37.14/.89	38.11/.93	38.47/.94	39.59/.95	39.84/.93	40.70/.94
	10	30.39/.67	30.55/.68	34.04/.89	31.86/.85	32.25/.87	34.51/.91	35.25/.92	34.82/.90	35.18/.91
40%	0	35.14/.85	38.51/.93	39.06/.95	38.63/.91	39.29/.95	41.00/.96	41.24/.97	41.56/.96	42.52/.98
	10	30.81/.68	31.24/.69	34.45/.90	33.32/.86	33.71/.87	34.83/.90	35.53/.91	35.31/.91	35.64/.91
50%	0	37.07/.89	39.93/.94	40.95/.95	40.24/.93	41.60/.95	42.53/.97	43.92/.98	43.00/.97	44.09/.98
	10	30.87/.66	31.54/.67	35.11/.91	34.39/.89	34.51/.89	35.35/.91	36.23/.93	35.71/.91	36.44/.93



Natural Image Reconstruction

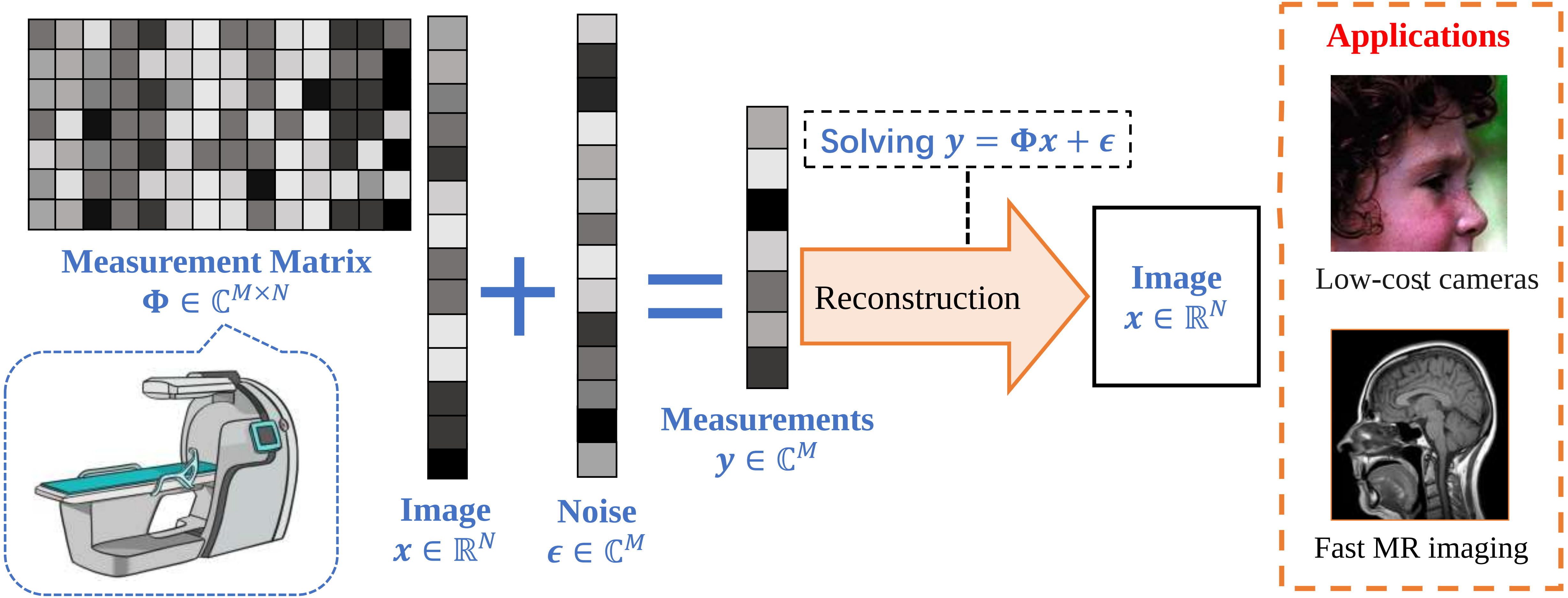
Mean PSNR(dB)/SSIM of natural image reconstruction on Set11 dataset.

CS Ratio	Noise Level	Regularized	Unsupervised		Internel		Unsupervised+Internel		Supervised		
		TVAL3 [COA-13]	L.SURE [Arxiv-20]	REI [CVPR-22]	BNN [ECCV-20]	ASGLD [CVPR-22]	DDSSL [ECCV-22]	MetaCS (Ours)	Supervised	COAST [TIP-21]	SSLIP [NIPS-21]
10%	0	22.45/0.38	25.00/.65	22.79/.64	27.49/.83	28.15/.83	27.48/.84	28.02/.84	26.94/.82	28.34/.84	25.02/.75
	10	21.02/0.54	23.31/.64	22.26/.66	25.23/.76	26.02/.76	26.10/.78	26.17/.78	25.03/.70	25.81/.78	24.48/.73
25%	0	27.63/0.62	31.31/.90	31.11/.90	32.30/.92	33.06/.92	33.28/.94	33.38/.94	32.44/.92	33.85/.94	30.42/.89
	10	24.75/0.67	28.14/.82	28.08/.81	28.67/.84	29.35/.85	29.61/.87	29.71/.87	29.49/.86	29.37/.86	28.71/.85
40%	0	31.21/0.75	33.30/95	35.63/.95	35.71/.95	35.87/.95	37.18/.96	37.25/.96	36.02/.96	36.94/.96	33.73/.93
	10	26.66/0.72	28.73/.81	28.99/.81	30.39/.88	31.11/.90	31.58/.88	31.64/.88	31.14/.89	31.16/.89	30.58/.89

COAST REI ASGLD DDSSL MetaCS GT

Now, let's dive into the details...

Compressive Sampling (CS) in Imaging



Ground Truth (GT)-Free Learning Meets CS

- New trend: Training deep neural networks (DNNs) for CS-based reconstruction w/o using GT images.
- Two types of GT-free deep learning:

End2End unsupervised learning	Untrained DNNs on test samples
EI/REI [1,2]; DDSSL [3]	BNN [4], SURE(s) [5],
External learning	Internal Learning

[1] Equivariant imaging: Learning beyond the range space. ICCV 2021.
[2] Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements. CVPR 2022.
[3] Dual-Domain Self-supervised Learning and Model Adaption for Deep Compressive Imaging. ECCV 2022.
[4] Self-supervised Bayesian deep learning for image recovery with applications to compressed sensing. ECCV 2020.
[5] Unsupervised learning with stein's unbiased risk estimator. arXivpreprint arXiv, 2018.

Motivation

Unsupervised External Learning (End2End DNNs)

(+) Exploiting large datasets

(+) Fast inference

(-) Non-adaptivity to test samples

Test-Time Internal Learning (Untrained DNNs)

(+) Sample-specific learning

(+) No dataset bias and OOD issues

(-) High-cost sample-wise fitting

Unsupervised (GT-Free) Meta-Learning for CS-Based Image Reconstruction

Framework

External

Training data $\mathbb{D} := \{\boldsymbol{y}_j, \boldsymbol{\Phi}\}_j$

Model: $\mathcal{F}_{\boldsymbol{\omega}}$ (initial para. $\boldsymbol{\omega}$)

$\mathcal{F}_{\boldsymbol{\omega}} \xrightarrow{\text{Meta Learning}} \mathcal{F}_{\widetilde{\boldsymbol{\omega}}}$

Internal

Test Sample $\boldsymbol{y}^*$

Meta-trained model: $\mathcal{F}_{\widetilde{\boldsymbol{\omega}}}$

$\mathcal{F}_{\widetilde{\boldsymbol{\omega}}} \xrightarrow{\text{Model Adaption}} \mathcal{F}_{\boldsymbol{\omega}^*}$

Keys

Range-nullspace decommission $\mathbb{C}^N = \mathrm{Range}\left(\boldsymbol{\Phi}^{\mathrm{H}}\right) \oplus \mathrm{Null}(\boldsymbol{\Phi})$

- How to suppress measurement noise on $\mathrm{Range}\left(\boldsymbol{\Phi}^{\mathrm{H}}\right)$ during training/adaption?
- How to learn reconstruction *w.r.t.* $\mathrm{Null}(\boldsymbol{\Phi})$?

Loss Functions | **Meta-Learning Scheme** | **NN Design**

Approach

Improved SURE (iSURE) loss for meta-learning and adaption

iSURE-based GT-free model-agnostic meta-learning (MAML)

Nullspace-consistent model adaptation

Unrolling CNN with bias-tuning

The iSure Loss

iSURE is a noisy form of SURE to provide robust estimation in $\mathrm{Range}(\boldsymbol{\Phi}^{\mathbf{H}})$.

$$\ell^{\mathrm{iSURE}}(\boldsymbol{\omega}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\epsilon}') \coloneqq \|\boldsymbol{\Phi}\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}', \boldsymbol{\Phi}) - \boldsymbol{y}\|_2^2 + 2\sigma \mathrm{tr}(\boldsymbol{\Phi}^{+}(\partial \mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}'), \boldsymbol{\Phi})/\partial \boldsymbol{y}))$$

- Noise injection mitigates overfitting in GT-free meta-learning and model adaption, as well as allows ensemble in learning and inference.
- iSURE allows efficient gradient update, without using MCMC.

Theorem 1

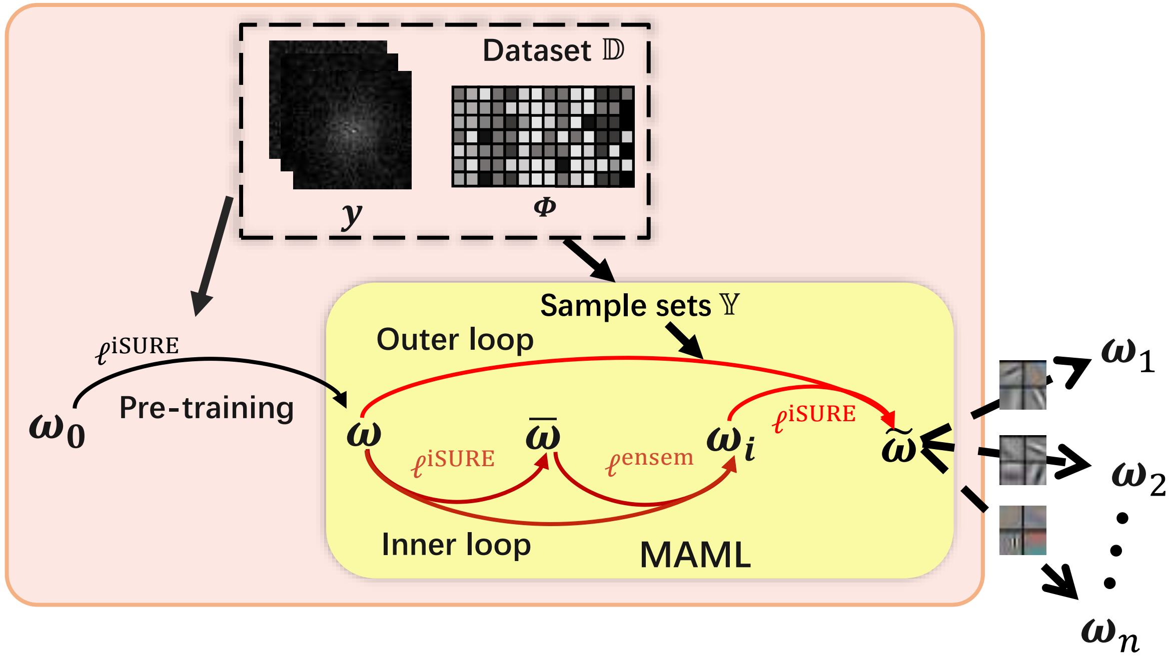
Let $\boldsymbol{J}_{\omega}$ be the Jacobian matrix w.r.t. $\boldsymbol{\omega}$, *i.e.* $\boldsymbol{J}_{\omega}\boldsymbol{\mathcal{F}}_{\omega} = \partial \boldsymbol{\mathcal{F}}_{\omega}/\partial \boldsymbol{\omega}$ and $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x} + \boldsymbol{\epsilon}$. Assume $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}' \backsim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2 \boldsymbol{I})$ are independent. Then, we have:

$$\nabla_{\boldsymbol{\omega}} \mathbb{E}_{\boldsymbol{y},\boldsymbol{\epsilon}'} \ell^{\mathrm{iSURE}}(\boldsymbol{\omega}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\epsilon}') = 2\mathbb{E}_{\boldsymbol{y},\boldsymbol{\epsilon}'}[\boldsymbol{J}_{\omega}(\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}', \boldsymbol{\Phi})\boldsymbol{\Phi}^{+}(\boldsymbol{\Phi}\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}', \boldsymbol{\Phi}) - \boldsymbol{y} + \boldsymbol{\epsilon}')].$$

Unsupervised MAML with Improvement

Algorithm 1: GT-Free Meta-Learning for CSR

Input: Measurement dataset $\mathbb{D}$; Initial ω
Required: Learn_Rates α, β, γ; Inner_Iter_Num Q
1. Pre-training: update ω with an iSURE-based loss
2. **while** not done **do** *Outer*
3. Sample sets of measurements $\{\mathbb{Y}_j\}_j$ from $\mathbb{D}$
4. **for each** $\mathbb{Y}_j$ **do** *Inner*
5. Initialize $\bar{\omega}$ with ω
6. **for** $q = 1, \cdots, Q$
7. $\bar{\omega} \leftarrow \bar{\omega} - \gamma \nabla_{\bar{\omega}} \mathcal{L}^{\text{iSURE}}_{\mathbb{Y}_j}(\bar{\omega})$
8. **end**
9. $\omega_j = \omega - \alpha \nabla_{\omega} \mathcal{L}^{\text{ensem}}_{\mathbb{Y}_j}(\omega; \bar{\omega})$
10. **end**
11. Update $\omega \leftarrow \omega - \beta \nabla_{\omega} \sum_j \mathcal{L}^{\text{iSURE}}_{\mathbb{Y}_j}(\omega_j)$
13. **end**



$$\mathcal{L}^{\text{iSURE}}_{\mathbb{Y}}(\boldsymbol{\omega}) \coloneqq \mathbb{E}_{\boldsymbol{y}\in\mathbb{Y}, \boldsymbol{\epsilon}' \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})} \ell^{\text{iSURE}}(\boldsymbol{\omega}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\epsilon}')$$

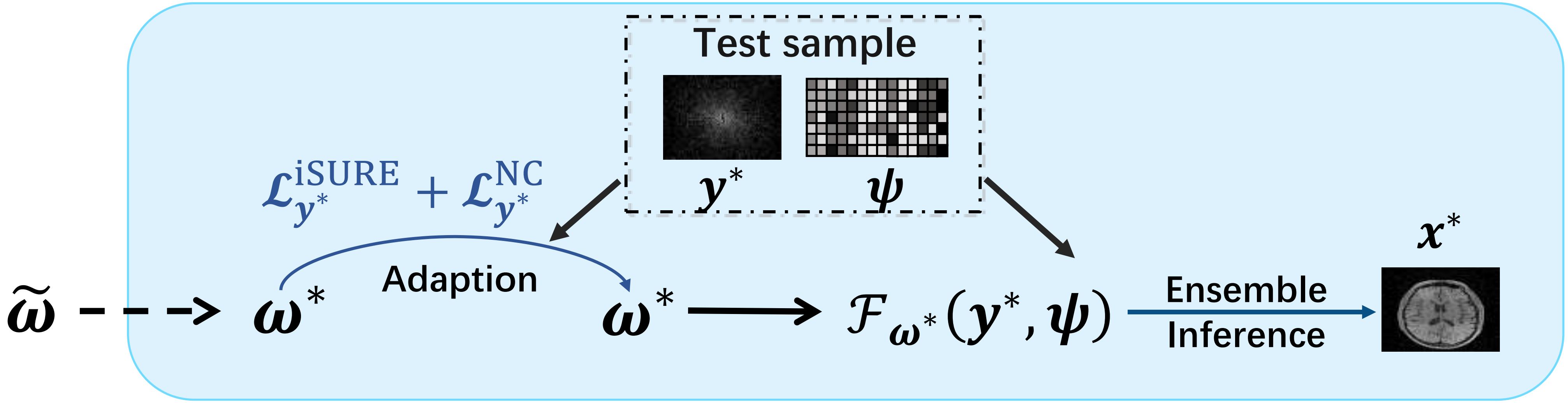
$$\mathcal{L}^{\text{ensem}}_{\mathbb{Y}}(\boldsymbol{\omega}) \coloneqq \mathbb{E}_{\boldsymbol{y}\in\mathbb{Y}} \left\| \mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y}, \boldsymbol{\Phi}) - 2\mathbb{E}_{\boldsymbol{\epsilon}'' \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})} \mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y} + \boldsymbol{\epsilon}'', \boldsymbol{\Phi}) \right\|_2^2$$

- Incorporate iSURE into MAML.
- iSURE only measures errors on range space for learning.
- Inner loop with $\mathcal{L}^{\text{ensem}}_{\mathbb{Y}}$ for better addressing the ambiguity on $\text{Null}(\boldsymbol{\Phi})$.

Nullspace-Consistent (NC) Adaptation

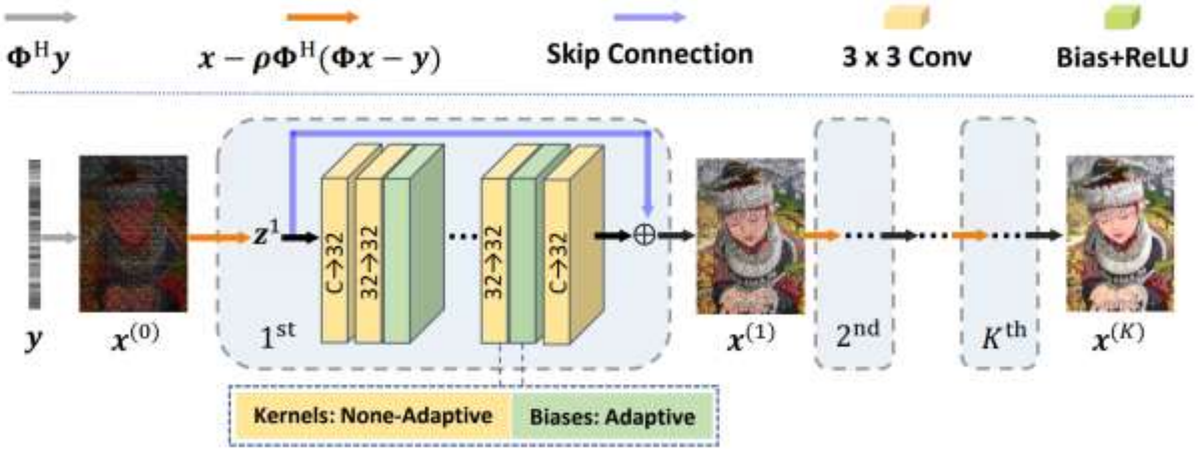
- Given a test sample $(\boldsymbol{y}^*, \boldsymbol{\psi})$, iSURE only considers reconstruction on $\mathrm{Range}(\boldsymbol{\psi}^{\mathrm{H}})$ and may bring negative effects to prediction on $\mathrm{Null}(\boldsymbol{\psi})$.
- NC adaptation mitigates negative effects in $\mathrm{Null}(\boldsymbol{\psi})$ by pulling the prediction on $\mathrm{Null}(\boldsymbol{\psi})$ back, done by the NC loss:

$$\mathcal{L}^{\mathrm{NC}}_{\boldsymbol{y}^*}(\boldsymbol{\omega};\boldsymbol{\psi},\boldsymbol{\omega}^*) := \left\|(\mathbf{I}-\boldsymbol{\psi}\boldsymbol{\psi}^{\dagger})\left(\mathcal{F}_{\boldsymbol{\omega}}(\boldsymbol{y}^*,\boldsymbol{\psi}) - \mathcal{F}_{\tilde{\boldsymbol{\omega}}}(\boldsymbol{y}^*,\boldsymbol{\psi})\right)\right\|_2^2.$$



Unrolling CNN with Bias-Tuning

- Unroll $\boldsymbol{x}^{(k)} = \mathrm{Prox}_{\boldsymbol{\psi}}\left(\boldsymbol{x}^{(k-1)} - \rho\boldsymbol{\Phi}^{\mathrm{H}}(\boldsymbol{\Phi}\boldsymbol{x}^{(k-1)} - \boldsymbol{y})\right)$ and replace the proximal operator $\mathrm{Prox}_{\boldsymbol{\psi}}$ replaced by a sub-CNN.

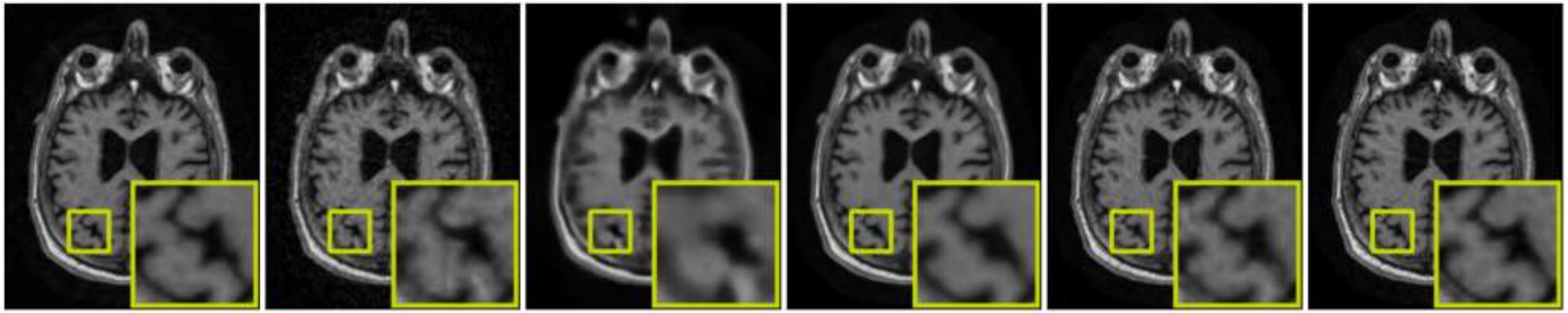


- An unrolling CNN acts as an iterative shrinkage process. The biases play a similar role to the thresholds of shrinkage, which are critical.
- **Bias-Tuning**: Only adjusting the bias parameters during adaption.

MR Image Reconstruction

Mean PSNR(dB)/SSIM of MR image reconstruction on MRI150 dataset.

CS Ratio	Noise Level	Regularized		Unsupervised	Internal		Unsupervised+Internel		Supervised	
		ZF [JMRI-01]	SparseMRI [MRM-07]	REI [CVPR-22]	BNN [ECCV-20]	ASGLD [CVPR-22]	DDSSL [ECCV-22]	MetaCS (Ours)	ADMMNet [TPAMI-19]	Supervised
20%	0	30.41/.72	35.46/.89	36.07/.90	35.54/.88	36.08/.90	36.73/.92	37.12/.93	37.17/.93	37.58/.94
	10	29.18/.64	29.74/.64	33.94/.89	35.54/.88	36.08/.90	33.79/.90	34.34/.91	34.04/.89	34.32/.90
30%	0	33.01/.80	37.72/.91	38.01/.92	37.14/.89	38.11/.93	38.47/.94	39.59/.95	39.84/.93	40.70/.94
	10	30.39/.67	30.55/.68	34.04/.89	31.86/.85	32.25/.87	34.51/.91	35.25/.92	34.82/.90	35.18/.91
40%	0	35.14/.85	38.51/.93	39.06/.95	38.63/.91	39.29/.95	41.00/.96	41.24/.97	41.56/.96	42.52/.98
	10	30.81/.68	31.24/.69	34.45/.90	33.32/.86	33.71/.87	34.83/.90	35.53/.91	35.31/.91	35.64/.91
50%	0	37.07/.89	39.93/.94	40.95/.95	40.24/.93	41.60/.95	42.53/.97	43.92/.98	43.00/.97	44.09/.98
	10	30.87/.66	31.54/.67	35.11/.91	34.39/.89	34.51/.89	35.35/.91	36.23/.93	35.71/.91	36.44/.93



Supervised REI ASGLD DDSSL MetaCS GT

Natural Image Reconstruction

Mean PSNR(dB)/SSIM of natural image reconstruction on Set11 dataset.

CS Ratio	Noise Level	Regularized	Unsupervised		Internel		Unsupervised+Internel		Supervised		
		TVAL3 [COA-13]	L.SURE [Arxiv-20]	REI [CVPR-22]	BNN [ECCV-20]	ASGLD [CVPR-22]	DDSSL [ECCV-22]	MetaCS (Ours)	Supervised	COAST [TIP-21]	SSLIP [NIPS-21]
10%	0	22.45/0.38	25.00/.65	22.79/.64	27.49/.83	28.15/.83	27.48/.84	28.02/.84	26.94/.82	28.34/.84	25.02/.75
	10	21.02/0.54	23.31/.64	22.26/.66	25.23/.76	26.02/.76	26.10/.78	26.17/.78	25.03/.70	25.81/.78	24.48/.73
25%	0	27.63/0.62	31.31/.90	31.11/.90	32.30/.92	33.06/.92	33.28/.94	33.38/.94	32.44/.92	33.85/.94	30.42/.89
	10	24.75/0.67	28.14/.82	28.08/.81	28.67/.84	29.35/.85	29.61/.87	29.71/.87	29.49/.86	29.37/.86	28.71/.85
40%	0	31.21/0.75	33.30/95	35.63/.95	35.71/.95	35.87/.95	37.18/.96	37.25/.96	36.02/.96	36.94/.96	33.73/.93
	10	26.66/0.72	28.73/.81	28.99/.81	30.39/.88	31.11/.90	31.58/.88	31.64/.88	31.14/.89	31.16/.89	30.58/.89

COAST REI ASGLD DDSSL MetaCS GT

Comparison on Computational Complexity

Running time (minutes) of different methods on Set11, tested on a TITAN RTX GPU.

Method	LSURE	BNN	ASGLD	DDSSL	MetaCS
r=40%	0.45	282	178	5.72	0.89

Compared to the latest adaption method DDSSL, our MetaCS takes only around 1/7 time!

Comparison in number of model parameters (M).

COAST	SSLIP	LDAMP-SURE	ASGLD	DDSSL	MetaCS (whole)	MetaCS (bias-only)
1.12	0.67	0.38	2.19	0.67	0.3756	0.0013

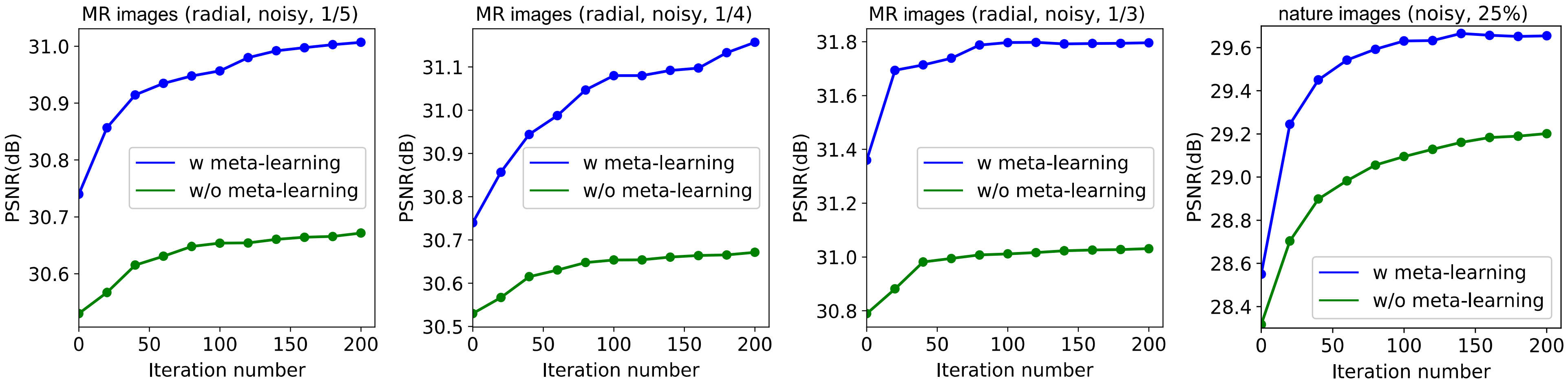
Our bias-tuning scheme significantly reduces the number of parameters being adapted.

Ablation Studies

Method	Noiseless MR Imaging			Noisy MR Imaging		
	r = 1/5	1/4	1/3	r = 1/5	1/4	1/3
iSure $\rightarrow$ gSURE	29.13	32.57	34.29	28.56	29.17	29.14
w/o Meta-Learn	32.93	33.97	35.68	29.52	30.03	30.34
Standard MAML	32.86	34.01	35.73	29.63	30.08	30.42
w/o Adaption	33.63	34.40	35.89	29.60	30.78	31.36
w/o NC	33.71	34.65	36.40	30.41	31.01	31.66
All weights	34.02	34.82	36.58	30.60	31.19	31.74
Gain-tuning	33.68	34.58	36.19	30.35	30.83	31.37
MetaCS	34.00	34.85	36.54	30.56	31.17	31.79

Each component of our approach has noticeable contribution to the performance.

Effectiveness of Meta-Learning & Adaption



Meta-learning leads to faster and more effective model adaptation.

Adaption for Unseen Measurement Matrices

Test Mask	Train=Test		Gauss, r=1/4	
	REI	MetaCS	REI	MetaCS
Gauss, r=1/3	36.72	37.87	36.38	37.8
Gauss, r=1/5	34.04	35.52	33.63	35.47
Radial, r=1/4	33.15	34.46	32.62	34.84

Thanks

See more at https://csyhquan.github.io