International Congress on Industrial and Applied Mathematics
Waseda University, Tokyo, Japan August 20-25, 2023

Deep Unrolling Networks with Recurrent Momentum Acceleration for Nonlinear Inverse Problems

Qingping Zhou[1]
in collaboration with
Jiayu Qian[1], Junqi Tang[2], and Jinglai Li[2]

[1] School of Mathematics and Statistics, Central South University, China
[2] School of Mathematics, University of Birmingham, UK

August 25th, 2023

What is an Inverse Problem?

- Formulation of an inverse problem

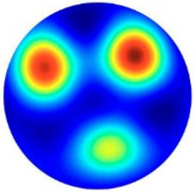
$$y = \mathcal{A}(x) + \epsilon,$$

where $\mathcal{A}$ represents a forward operator and ϵ is the observation noise. We want to infer a quantity x from indirect observations y.

Examples: Computed Tomography

$\mathcal{A}(x) \approx Ax$ is Linear.
Many research papers

Examples: Electrical Impedance Tomography



$\mathcal{A}(x) = A(x)$ is nonlinear.
A few research papers!
(This talk)

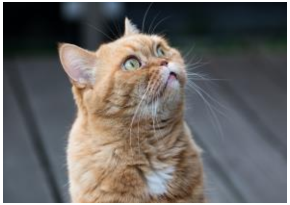
Computational Challenges

- ill-posed
- large-scale nature

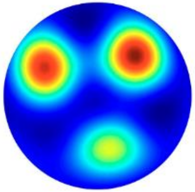
Computational Challenges

- ill-posed
- large-scale nature
- lack of relevant training data

This may be irrelevant in settings with lots of training data



but is critical in more limited-data settings



AI-driven methods for Inverse Problems

Fully data-driven methods: train an input-to-solution DNN

- fast inference: fewer layers than classic optimization iterations
- slow training: too many parameters
- Inaccurate solutions: poor generalization

Model-based methods: modify classical optimization algorithms

- deep unrolling networks (**DuNets**), a.k.a algorithm unrolling (this talk)
- Plug-and-play
- Deep equilibrium or fixed-point network
- …

Deep unrolling networks (DuNets)

- DuNets consists of two steps

(1) Pick a classic iterative optimization algorithm and unroll it to an DNN

(2) Select a set of DNN parameters to learn

- Example: assume $y = Ax + noise$; recover x by minimizing

$$\arg\min_{x' \in X} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \mathcal{R}(x)$$

Deep unrolling networks (DuNets)

- DuNets consists of two steps

(1) Pick a classic iterative optimization algorithm and unroll it to an DNN

(2) Select a set of DNN parameters to learn

- Example: assume $\mathrm{y} = Ax + noise$; recover x by minimizing

$$\arg\min_{x' \in X} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \mathcal{R}(x)$$

- Proximal gradient descent (PGD):

$$x_t = \mathcal{P}_{\lambda\mathcal{R}}(x_{t-1} - \alpha \mathrm{A}^T (Ax_{t-1} - y)),$$

where $\mathcal{P}_{\lambda\mathcal{R}}(\cdot)$ is the proximal operator:

$$\mathcal{P}_{\lambda\mathcal{R}}(x) = \arg\min_{x' \in X} \frac{1}{2} \|x' - x\|_X^2 + \lambda \mathcal{R}(x').$$

Learned Proximal gradient descent (LPGD)

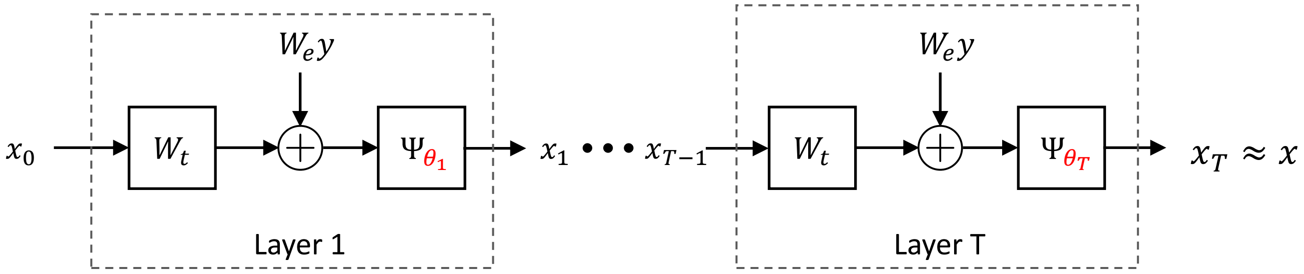
- Introduce $W_e = \alpha \mathrm{A}^T$ and $W_t = \mathrm{I} - \alpha \mathrm{A}^T A$, rewrite PGD as

$$x_t = \mathcal{P}_{\lambda\mathcal{R}}(W_t x_{t-1} + W_e y).$$

- LearnedPGD: replace $\mathcal{P}_{\lambda\mathcal{R}}$ with learned network Ψ_{θ_t}

$$x_t = \Psi_{\theta_t}(W_t x_{t-1} + W_e y), \text{ for } t = 1, \ldots, T,$$

which resembles a DNN



The idea was successfully applied to other algorithms and many applications:

- Iterative Shrinkage and Thresholding Algorithm (ISTA)

  Signal processing: [Gregor and *LeCun, 2010*]

  Super-resolution: [Wang et al., 2015]

  computed tomography: [Jin et al., 2017]

- Alternating direction method of multipliers-ADMM

  Rain removal: [Ding et al., 2018]

  Medical resonance imaging: [Yang et al., 2019]

- Primal dual hybrid gradient-PDHG

  Computed tomography: [*Adler-Öktem,* 2018]

- Proximal interior point

  Image deblurring: [Bertocchi, 2020]

- Proximal gradient descent-PGD

  Medical resonance imaging: [Hosseini et al., 2019]

Computational challenges

- In a nonlinear setting, most DuNets methods only use the current gradient, overlooking a significant amount of historical gradient data

linear: $\frac{\partial\ Ax}{\partial x_t} = \mathrm{A}$ noninear: $\frac{\partial A(x)}{\partial x_t}$

- How to use the historical gradient data more effectively?

- A possible answer: The momentum acceleration strategy

Momentum

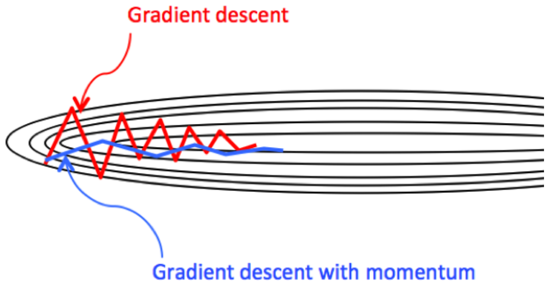
- Gradient Descent

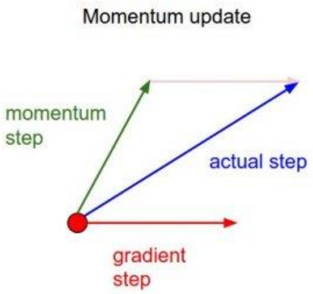
$$x_{t+1} = x_t - \alpha g_t$$

- Gradient Descent with Momentum

$$v_{t+1} = \beta v_t + g_t$$

$$x_{t+1} = x_t - \alpha v_{t+1}$$





- Gradient descent is primarily sensitive to the choice of learning rate
- Momentum can dampen oscillations in directions of high curvature, potentially leading to faster convergence in practice for some problems
- Momentum may navigate more efficiently through poorly conditioned or non-convex landscapes

Unroll momentum

- Unroll the velocity update step $v_{t+1} = \beta v_t + g_t$, we have

$$v_1 = \beta v_0 + g_0 = g_0$$

$$v_2 = \beta v_1 + g_1 = \beta^2 g_0 + \beta g_1$$

$$\cdots$$

$$v_{t+1} = \sum_{i=0}^{t} \beta^{t-i} g_i$$

Then, we unroll the $x_{t+1} = x_t - \alpha v_t$, we have

$$x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \frac{\left(1 - \beta^{k+1-i}\right)}{1 - \beta} g_i$$

Unroll momentum

- Unroll the velocity update step $v_{t+1} = \beta v_t + g_t$, we have

$$v_1 = \beta v_0 + g_0 = g_0$$
$$v_2 = \beta v_1 + g_1 = \beta^2 g_0 + \beta g_1$$
$$\cdots$$
$$v_{t+1} = \sum_{i=0}^{t} \beta^{t-i} g_i$$

Then, we unroll the $x_{t+1} = x_t - \alpha v_t$, we have

$$x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \frac{\left(1 - \beta^{k+1-i}\right)}{1 - \beta} g_i$$

- Many optimization algorithms can be written in this unrolled form.

(1) $x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \gamma_i^k g_i$: Gradient descent($\gamma_i^k = 1$), Conjugate gradient, AdaMax

(2) $x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \Gamma_i^k g_i$: AdaGrad, Adam

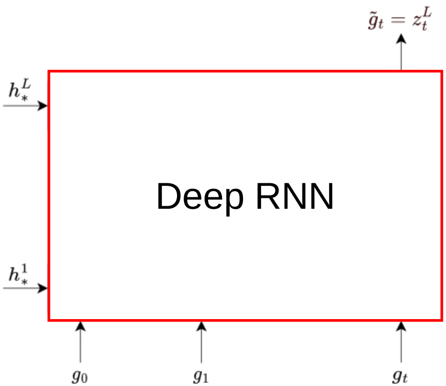
- Could one perhaps choose the α and β intelligently and adaptively?

Recurrent Momentum Acceleration via deep RNN

- A more flexible scheme that uses the recurrent neural networks (RNN) to learn the velocity term

(1) $x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \gamma_i^k g_i$ (2) $x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \Gamma_i^k g_i$

$$x_{t+1} = x_0 + RNN(g_0, g_1, \cdots, g_t)$$



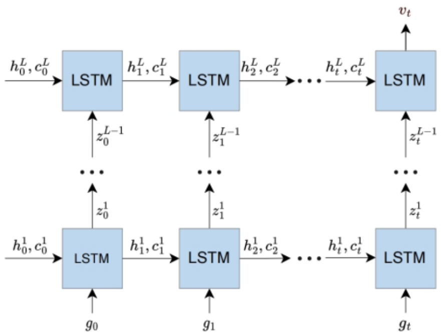
Recurrent Momentum Acceleration via deep RNN

- A more flexible scheme that uses the recurrent neural networks (RNN) to learn the velocity term

$$(1)\ x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \gamma_i^k g_i \qquad (2)\ x_{t+1} = x_0 + \alpha \sum_{i=0}^{k} \Gamma_i^k g_i$$

$$\Downarrow$$

$$x_{t+1} = x_0 + \Xi_{\vartheta}(g_0, g_1, \cdots, g_t)$$



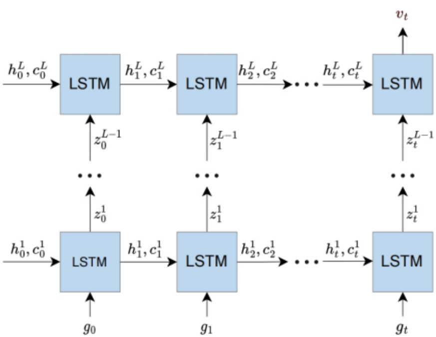
$$(z_t^l, h_t^l, c_t^l) = \mathrm{LSTM}^l(z_t^{l-1}, h_{t-1}^l, c_{t-1}^l)$$

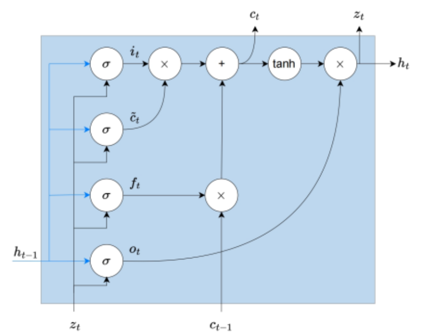
Recurrent Momentum Acceleration via deep RNN

- A more flexible scheme that uses the recurrent neural networks (RNN) to learn the velocity term

$$(1)\ x_{t+1} = x_0 + \alpha \textstyle\sum_{i=0}^{k} \gamma_i^k g_i \qquad (2)\ x_{t+1} = x_0 + \alpha \textstyle\sum_{i=0}^{k} \Gamma_i^k g_i$$

$$x_{t+1} = x_0 + \Xi_\vartheta(g_0, g_1, \cdots, g_t)$$





$$\tilde{c}_t = \tanh\left(W_{hc} h_{t-1}^{l} + W_{gc} z_t^{l-1} + b_c\right)$$

$$f_t = \sigma\left(W_{hx} h_{t-1}^{l} + W_{gx} z_t^{l-1} + b_x\right)$$
$$i_t = \sigma\left(W_{hi} h_{t-1}^{l} + W_{gi} z_t^{l-1} + b_i\right)$$
$$o_t = \sigma\left(W_{ho} h_{t-1}^{l} + W_{go} z_t^{l-1} + b_o\right)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t$$
$$h_t = o_t \otimes \tanh(c_t) = z_t$$

LPGD-MA/RMA methods

- Problem $\arg\min_{x} \mathcal{D}(\mathcal{A}(x_t), y) + \lambda\mathcal{R}(x)$

- Proximal gradient descent (PGD)

$$x_t = \mathcal{P}_{\lambda\mathcal{R}}(x_{t-1} - \alpha_t g_{t-1})$$

- LearnedPGD

$$x_t = \Psi_{\theta_t}(x_{t-1}, g_{t-1}), \text{ for } t = 1, \dots, T.$$

- Apply RMA to LPGD

Algorithm 3 LPGD-MA

Input: $x_0 \in X$, $v_0 = 0$
Output: x_T

for $t = 1, \dots, T$ **do**
  $g_{t-1} = \nabla_{x_{t-1}} \mathcal{D}(\mathcal{A}(x_{t-1}), y)$
  $v_t = \gamma v_{t-1} - \eta g_{t-1}$
  $x_t = \Psi_{\theta_t}(x_{t-1}, v_t)$
end for

Algorithm 4 LPGD-RMA algorithm

Input: $x_0 \in X, h_0 = 0, c_0 = 0$
Output: x_T

for $t = 1, \dots, T$ **do**
  $g_{t-1} = \nabla_{x_{t-1}} \mathcal{D}(\mathcal{A}(x_{t-1}), y)$
  $(v_t, h_t, c_t) = \Xi_\vartheta(g_{t-1}, h_{t-1}, c_{t-1})$
  $x_t = \Psi_{\theta_t}(x_{t-1}, v_t)$
end for

LPGD-MA/RMA methods

- Problem $\arg\min_x \mathcal{D}(\mathcal{A}(x_t), y) + \lambda\mathcal{R}(x)$

- Proximal gradient descent (PGD)

$$x_t = \mathcal{P}_{\lambda\mathcal{R}}(x_{t-1} - \alpha_t g_{t-1})$$

- LearnedPGD

$$x_t = \Psi_{\theta_t}(x_{t-1}, g_{t-1}), \text{ for } t = 1, \dots, T.$$

- Apply RMA to LPGD and LPSDSW($\theta_t = \theta$, for t=1,….,T)

LPGDSW-MA, LPGDSW-RMA

LPD-MA/RMA methods

- Hybrid gradient primal-dual method

$$u_{t+1} = \mathrm{prox}_{\sigma\mathcal{A}^*}\big(u_t + \sigma\mathcal{A}(\bar{x}_t)\big)$$
$$x_{t+1} = \mathrm{prox}_{\tau\mathcal{R}}\big(x_t - \tau[\partial\mathcal{A}(x_t)]^*(u_{t+1})\big)$$
$$\bar{x}_{t+1} = x_{t+1} + \gamma(x_{t+1} - x_t)$$

- Learned primal-dual (LPD), for $t = 1, \ldots, T$:

$$u_t = \Gamma_{\theta_t^d}(u_{t-1}, \mathcal{A}(x_{t-1}), y)$$
$$x_t = \Lambda_{\theta_t^p}\big(x_{t-1}, [\partial\mathcal{A}(x_{t-1})]^*(u_t)\big)$$

- Apply RMA to LPD

Algorithm 5 LPD-MA algorithm

Input: $x_0 \in X^{N_{\text{primal}}}, u_0 \in Y^{N_{\text{dual}}}$
Output: $x_T^{(1)}$

for $t = 1, \ldots, T$ **do**
  $u_t = \Gamma_{\theta_t^d}\left(u_{t-1}, \mathcal{A}(x_{t-1}^{(2)}), y\right)$
  $g_{t-1} = \left[\partial\mathcal{A}(x_{t-1}^{(1)})\right]^* u_t^{(1)}$
  $v_t = \gamma v_{t-1} - \eta g_{t-1}$
  $x_t = \Lambda_{\theta_t^p}(x_{t-1}, v_t)$
end for

Algorithm 6 LPD-RMA algorithm

Input: $x_0 \in X^{N_{\text{primal}}}, u_0 \in Y^{N_{\text{dual}}}, h_0 = 0, c_0 = 0$
Output: $x_T^{(1)}$

for $t = 1, \ldots, T$ **do**
  $u_t = \Gamma_{\theta_t^d}\left(u_{t-1}, \mathcal{A}(x_{t-1}^{(2)}), y\right)$
  $g_{t-1} = \left[\partial\mathcal{A}(x_{t-1}^{(1)})\right]^* (u_t^{(1)})$
  $(v_t, h_t, c_t) = \Xi_\vartheta(g_{t-1}, h_{t-1}, c_{t-1})$
  $x_t = \Lambda_{\theta_t^p}(x_{t-1}, v_t)$
end for

Numerical experiments

Example1: a nonlinear deconvolution

- $\mathrm{x} \in \mathbb{R}^{53}$ and $\mathrm{y} \in \mathbb{R}^{12}$
- Training set: Validation set: Test set = 10000:1000:1000

Examples 2: an electrical impedance tomography (EIT) image reconstruction

- $\mathrm{x} \in \mathbb{R}^{1342}$ and $\mathrm{y} \in \mathbb{R}^{208}$
- Training set: Validation set: Test set = 400:20:20

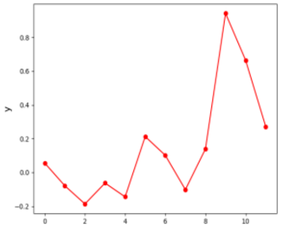
Network structures

- LPGD-type: $T = 20$ LPD-type: $T = 10,$ $\mathrm{N}^{primal} = \mathrm{N}^{primal}$=5
- Adam optimizer with cosine annealing
- Training 20 epochs
- Loss function

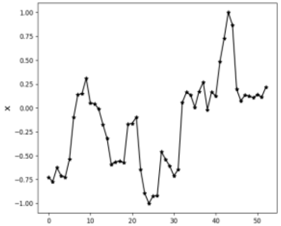
$$\ell(\Phi) = \frac{1}{N}\sum_{n=1}^{N} \|\hat{x}_i(y_i;\Phi) - x_i\|_2^2$$

Results

Observation



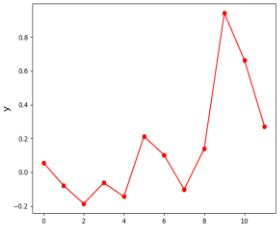
Ground Truth



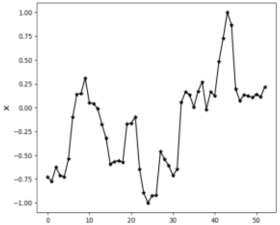
Inverse problem:
Given the nonlinear convolution result y via $y = a \cdot x' W_2 x + w_1' x + b$, we want to infer x.

Results

Observation



Ground Truth

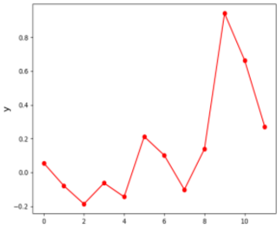


	$a = 0$
LPGD	—
LPGD-MA	3.21E-02
LPGD-RMA	3.23E-02
LPGDSW	3.01E-02
LPGDSW-MA	3.02E-02
LPGDSW-RMA	3.01E-02
LPD	2.69E-02
LPD-MA	2.68E-02
LPD-RMA	2.67E-02

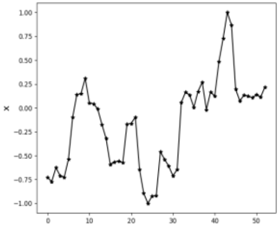
- when a = 0 , $y = 0 \cdot x'W_2x + w_1'x + b,$ DuNets methods are almost the same

Results

Observation



Ground Truth

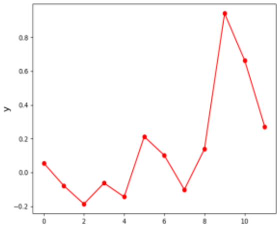


	$a = 1$	$a = 2$	$a = 4$
LPGD	—	—	—
LPGD-MA	5.37E-02	6.49E-02	7.76E-02
LPGD-RMA	3.56E-02	4.43E-02	4.97E-02
LPGDSW	4.61E-02	5.88E-02	6.85E-02
LPGDSW-MA	4.54E-02	5.32E-02	5.78E-02
LPGDSW-RMA	3.89E-02	4.68E-02	5.24E-02
LPD	3.65E-02	4.61E-02	5.17E-02
LPD-MA	3.71E-02	4.65E-02	5.22E-02
LPD-RMA	3.35E-02	4.04E-02	4.33E-02
	8%	12%	16%

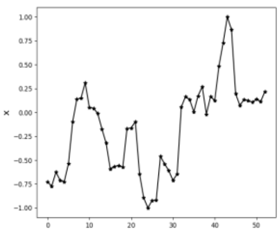
- when a = 0, DuNets methods are almost the same
- when a > 0, DuNets-RMA methods outperform other methods

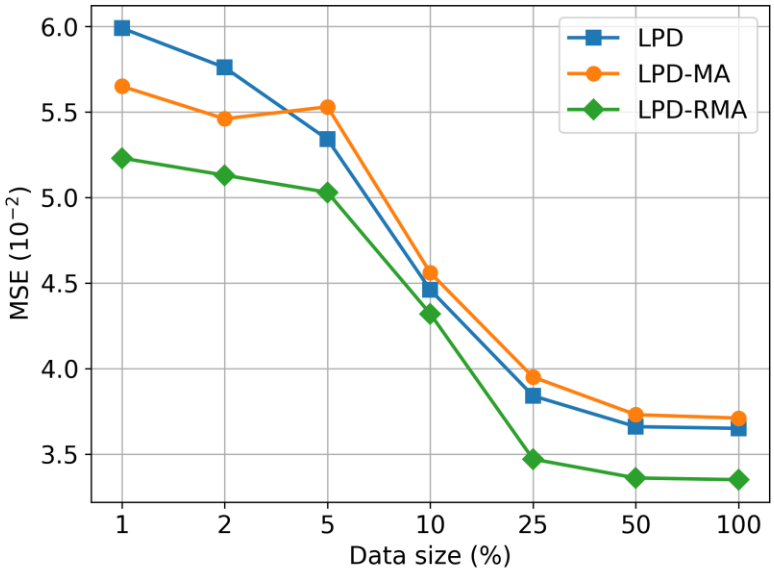
Results

Observation



Ground Truth





- when a = 0, DuNets method are almost the same
- when a > 0, DuNets-RMA methods outperform other methods
- LPD-RMA is considerably more data efficient

Application: EIT inverse problem

- Mathematical model of EIT

  Conductivity equation:

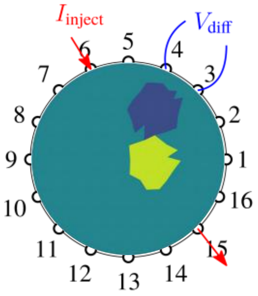
  $$\nabla \cdot (\sigma \nabla u) = 0 \qquad \text{in } \Omega$$

  Boundary conditions (known):

  Voltages: $u = V \mid_{\partial\Omega}$ (Dirichlet BC)

  Currents: $\sigma \dfrac{\partial u}{\partial e} = I \mid_{\partial\Omega}$ (Neumann BC)

  Dirichlet-to-Neumann (DN) map: $\Lambda_\sigma : u \mid_{\partial\Omega} \to \sigma \dfrac{\partial u}{\partial e} \mid_{\partial\Omega}$

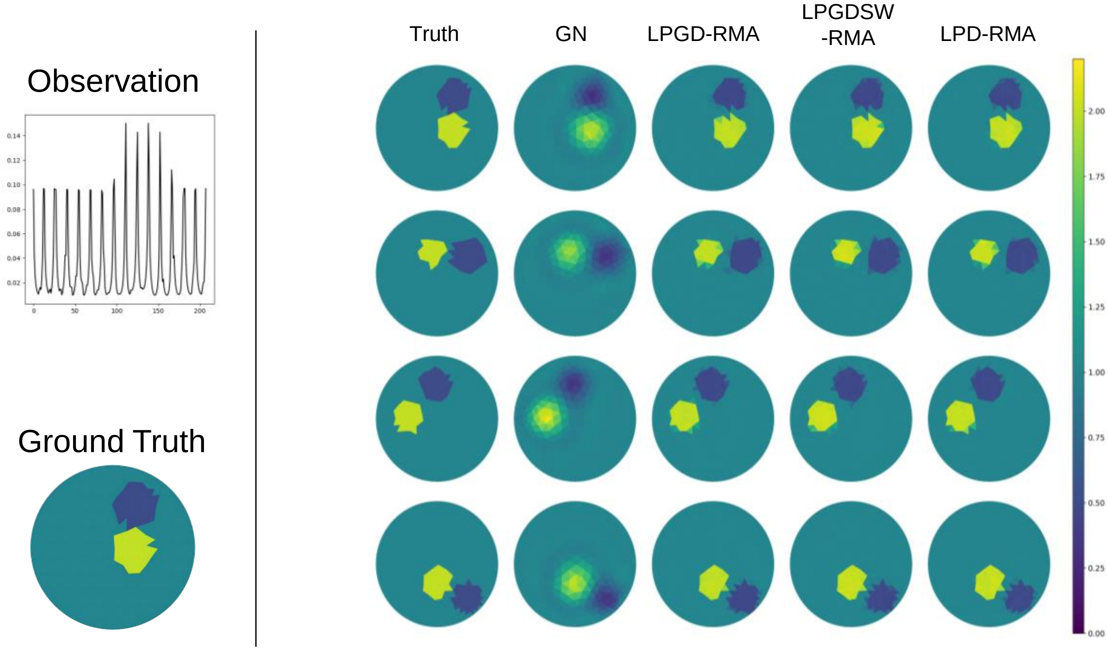


- The EIT inverse problem: given known Λ_σ , recover σ in Ω

- We discretize the object domain and define a mapping F representing the discrete version of the forward operator:

$$v = F(\sigma) + \eta$$

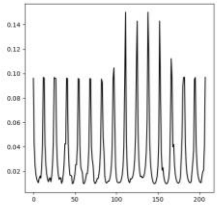
Results : two circles



- DuNets-RMA models yield accurate reconstruction for all the inclusions with different geometry and topology

Results: two circles

Observation



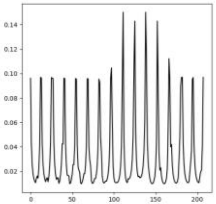
Ground Truth

	50	200	400
LPGD	—	—	—
LPGD-MA	6.13E-03 (± 16.1E-04)	5.17E-03 (± 14.1E-04)	4.18E-03 (± 10.5E-04)
LPGD-RMA	3.02E-03 (± 1.34E-04)	2.48E-03 (± 1.08E-04)	2.25E-03 (± 1.41E-04)
LPGDSW	4.33E-03 (± 13.7E-04)	2.87E-03 (± 1.15E-04)	3.07E-03 (± 1.86E-04)
LPGDSW-MA	3.81E-03 (± 3.15E-04)	2.92E-03 (± 1.40E-04)	2.95E-03 (± 1.55E-04)
LPGDSW-RMA	3.92E-03 (± 4.79E-04)	2.65E-03 (± 1.99E-04)	2.63E-03 (± 1.14E-04)
LPD	3.25E-03 (± 1.87E-04)	2.55E-03 (± 1.34E-04)	2.35E-03 (± 2.13E-04)
LPD-MA	3.29E-03 (± 1.09E-04)	2.71E-03 (± 1.46E-04)	2.44E-03 (± 1.64E-04)
LPD-RMA	3.11E-03 (± 1.52E-04)	2.17E-03 (± 1.36E-04)	2.04E-03 (± 1.89E-04)

- DuNets-RMA models yield accurate reconstruction for all the inclusions
  DuNets-RMA models achieve the best performance in all but one case (LPGDSW method with 50 training samples)

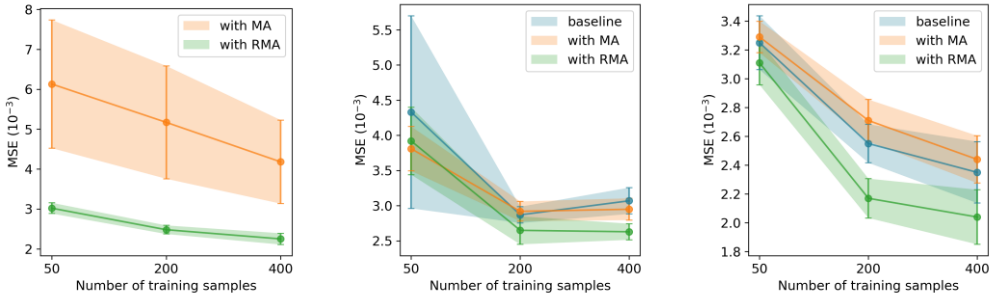
Results: two circles

Observation



Ground Truth

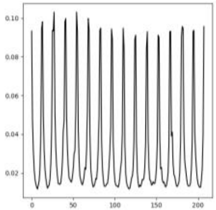
LPGD LPGDSW LPD



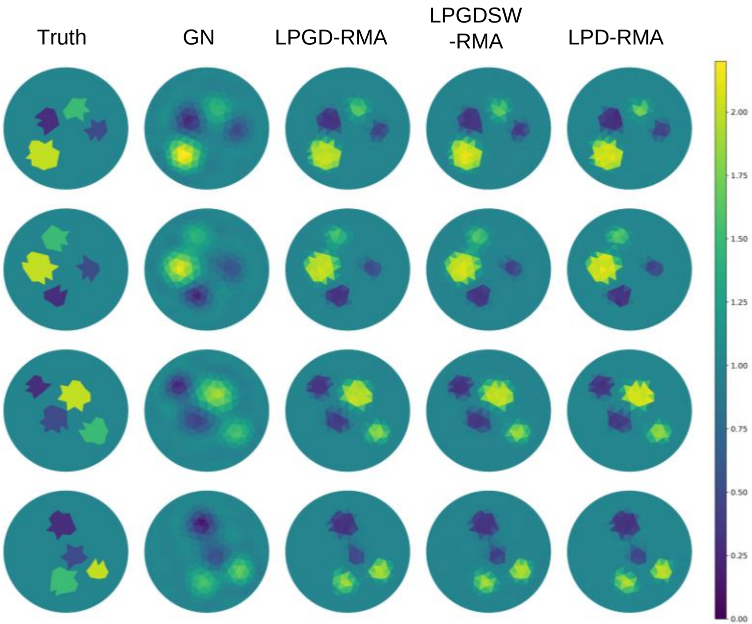
- DuNets-RMA models yield accurate reconstruction for all the inclusions having different geometry and topology
- DuNets-RMA models achieve the best performance in all but one case (LPGDSW method with 50 training ……samples)
- DuNets-RMA scheme has better stability and data efficiency

Results: four circles

Observation



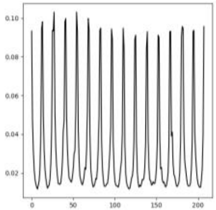
Ground Truth



- The number of inclusions affects the quality of the reconstructions, slightly corrupting the identification of the different anomalies.

Results: four circles

Observation



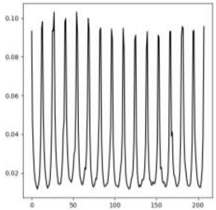
Ground Truth

data size	50	200	400	two circles 400
LPGD	—	—	—	—
LPGD-MA	10.3E-03 (± 1.54E-04)	8.60E-03 (± 1.45E-04)	7.87E-03 (± 1.15E-04)	4.18E-03 (± 10.5E-04)
LPGD-RMA	7.87E-03 (± 1.46E-04)	7.43E-03 (± 1.27E-04)	6.66E-03 (± 1.28E-04)	2.25E-03 (± 1.41E-04)
LPGDSW	7.81E-03 (± 1.07E-04)	6.88E-03 (± 2.13E-04)	6.69E-03 (± 2.39E-04)	3.07E-03 (± 1.86E-04)
LPGDSW-MA	6.96E-03 (± 5.04E-04)	5.46E-03 (± 2.01E-04)	5.53E-03 (± 2.26E-04)	2.95E-03 (± 1.55E-04)
LPGDSW-RMA	6.82E-03 (± 3.42E-04)	5.16E-03 (± 2.26E-04)	5.02E-03 (± 2.15E-04)	2.63E-03 (± 1.14E-04)
LPD	6.66E-03 (± 2.72E-04)	5.46-03 (± 1.90E-04)	5.10E-03 (± 1.71E-04)	2.35E-03 (± 2.13E-04)
LPD-MA	6.32E-03 (± 1.95E-04)	5.49E-03 (± 1.63E-04)	5.18E-03 (± 1.86E-04)	2.44E-03 (± 1.64E-04)
LPD-RMA	6.01E-03 (± 1.60E-04)	5.13E-03 (± 1.91E-04)	4.93E-03 (± 1.49E-04)	2.04E-03 (± 1.89E-04)

- The number of inclusions affects the quality of the reconstructions, slightly corrupting the identification of the different anomalies.
- DuNets-RMA models achieve the best performance in all cases

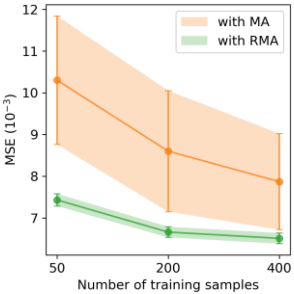
Results: two circles

Observation

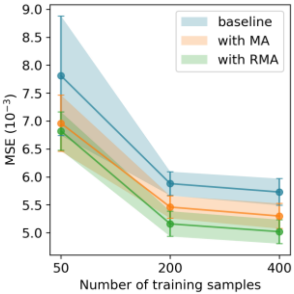


Ground Truth

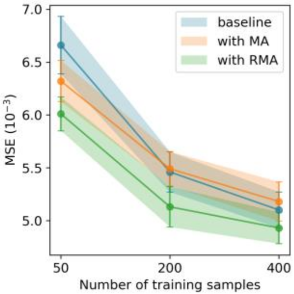
LPGD



LPGDSW



LPD



- The number of inclusions affects the quality of the reconstructions, slightly corrupting the identification of the different anomalies.
- DuNets-RMA models achieve the best performance in all cases
- DuNets-RMA scheme has better stability and data efficiency

Summary

- We develop a new deep unrolling networks incorporating recurrent momentum acceleration for solving nonlinear inverse problem more accurate

- Future research direction
  - theoretical analysis, such as convergence
  - design an unrolling structure based on PDE's theoretical properties

- Check our paper for more

Zhou, Q., Qian, J., Tang, J., & Li, J. (2023). Deep Unrolling Networks with Recurrent Momentum Acceleration for Nonlinear Inverse Problems. https://arxiv.org/abs/2307.16120

Reference

- K. Gregor and Y. LeCun, “Learning Fast Approximations of Sparse Coding,” in Proc. Int. Conf. Machine Learning, 2010.
- Lohit, Suhas, et al. "Unrolled projected gradient descent for multi-spectral image fusion." *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.
- Mardani, Morteza, et al. "Neural proximal gradient descent for compressive imaging." *Advances in Neural Information Processing Systems* 31 (2018).
- Qian, Ning. "On the momentum term in gradient descent learning algorithms." *Neural networks* 12.1 (1999): 145-151.