

# Bilevel Learning for Inverse Problems

Matthias J. Ehrhardt

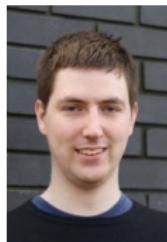
Department of Mathematical Sciences, University of Bath, UK

June 17, 2022

Joint work with:

L. Roberts (ANU, Australia)

F. Sherry, M. Graves, G. Maierhofer, G. Williams, C.-B. Schönlieb (all Cambridge, UK), M. Benning (Queen Mary, UK),  
J.C. De los Reyes (EPN, Ecuador)



Lindon Roberts



Ferdia Sherry



The Leverhulme Trust



Engineering and  
Physical Sciences  
Research Council



UNIVERSITY OF  
**BATH**

# Outline

## 1) Motivation



$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \mathcal{R}(x)$$

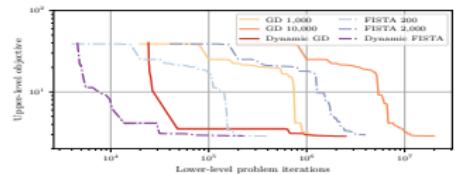
$$\min_{x,y} f(x,y)$$

$$x \in \arg \min_z g(z,y)$$

## 2) Efficient solution?

Yes, e.g. inexact DFO algorithms

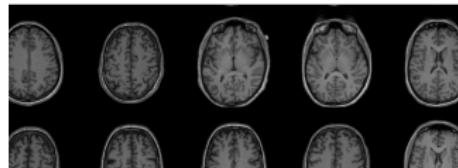
[Ehrhardt and Roberts JMV '21](#)



## 3) High-dimensional learning?

Yes, e.g. learn MRI sampling

[Sherry et al. IEEE TMI '20](#)



# Inverse problems and Variational Regularization

$$A\textcolor{red}{x} = \textcolor{blue}{y}$$

$\textcolor{red}{x}$  : desired solution

$\textcolor{blue}{y}$  : observed data

$A$  : mathematical model

**Goal:** recover  $\textcolor{red}{X}$  given  $\textcolor{blue}{y}$

# Inverse problems and Variational Regularization

$$A\mathbf{x} = \mathbf{y}$$

$\mathbf{x}$  : desired solution

$\mathbf{y}$  : observed data

$A$  : mathematical model

**Goal:** recover  $\mathbf{X}$  given  $\mathbf{y}$

## Variational regularization

Approximate a solution  $\mathbf{x}^*$  of  $A\mathbf{x} = \mathbf{y}$  via

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \left\{ \mathcal{D}(A\mathbf{x}, \mathbf{y}) + \lambda \mathcal{R}(\mathbf{x}) \right\}$$

$\mathcal{D}$  **data fidelity**: related to noise statistics

$\mathcal{R}$  **regularizer**: penalizes unwanted features, stability

$\lambda \geq 0$  **regularization parameter**: weights data and regularizer

# Example: Magnetic Resonance Imaging (MRI)

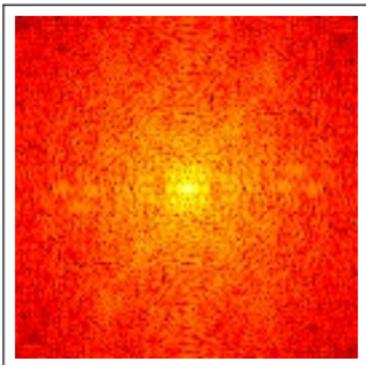
**MRI Reconstruction** Lustig et al. '07

Fourier transform  $\mathcal{F}$ , sampling  $S\mathbf{w} = (w_i)_{i \in \Omega}$

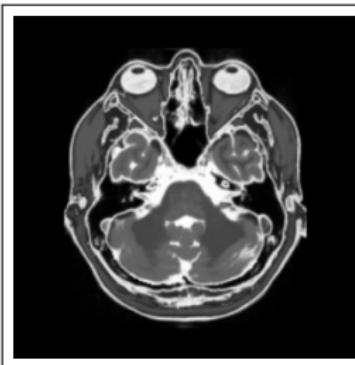
$$\min_{\mathbf{x}} \left\{ \sum_{i \in \Omega} |(\mathcal{F}\mathbf{x})_i - \mathbf{y}_i|^2 + \lambda \|\nabla \mathbf{x}\|_1 \right\}$$



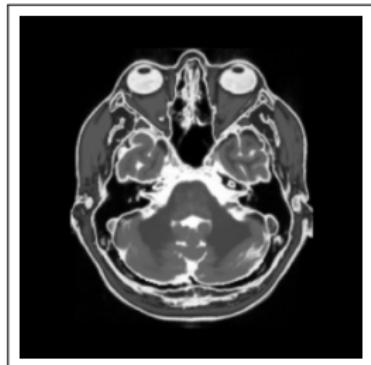
MRI scanner



sampling  $S^* \mathbf{y}$



$\lambda = 0$



$\lambda = 1$

# Example: Magnetic Resonance Imaging (MRI)

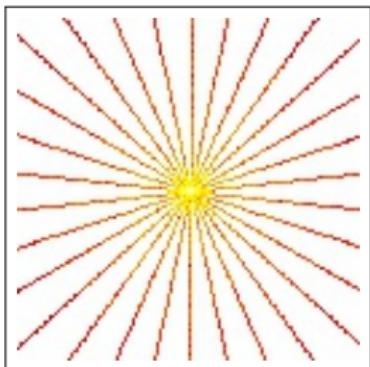
**MRI Reconstruction** Lustig et al. '07

Fourier transform  $\mathcal{F}$ , sampling  $S\mathbf{w} = (w_i)_{i \in \Omega}$

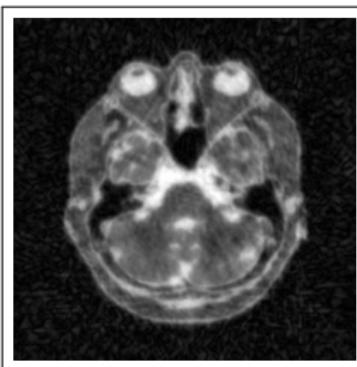
$$\min_{\mathbf{x}} \left\{ \sum_{i \in \Omega} |(\mathcal{F}\mathbf{x})_i - \mathbf{y}_i|^2 + \lambda \|\nabla \mathbf{x}\|_1 \right\}$$



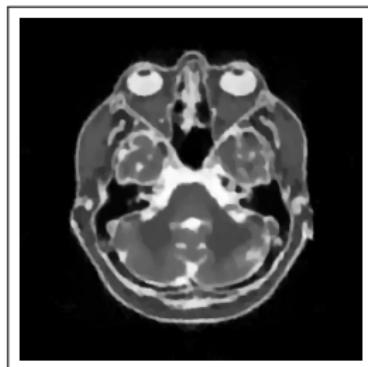
MRI scanner



sampling  $S^* \mathbf{y}$



$\lambda = 0$



$\lambda = 10^{-4}$

# Example: Magnetic Resonance Imaging (MRI)

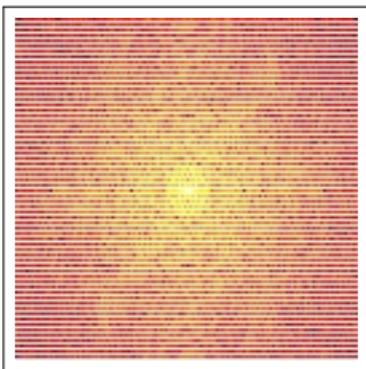
## MRI Reconstruction Lustig et al. '07

Fourier transform  $\mathcal{F}$ , sampling  $S\mathbf{w} = (w_i)_{i \in \Omega}$

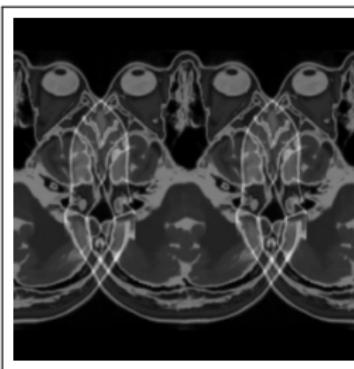
$$\min_{\mathbf{x}} \left\{ \sum_{i \in \Omega} |(\mathcal{F}\mathbf{x})_i - y_i|^2 + \lambda \|\nabla \mathbf{x}\|_1 \right\}$$



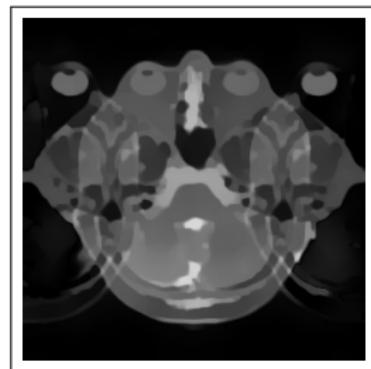
MRI scanner



sampling  $S^*y$



$\lambda = 0$

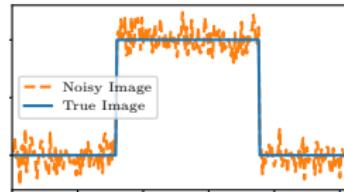


$\lambda = 10^{-3}$

How to choose the sampling  $\Omega$ ? Should it depend on  $\mathcal{R}$  and  $\lambda$ ?

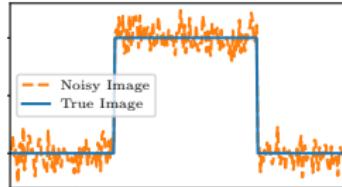
## More “complicated” regularizers

$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \alpha \left( \underbrace{\sum_j \|(\nabla x)_j\|_2}_{=TV(x)} \right)$$



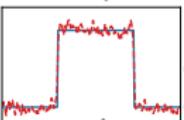
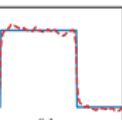
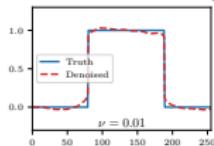
## More “complicated” regularizers

$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \alpha \left( \underbrace{\sum_j \sqrt{\|(\nabla x)_j\|_2^2 + \nu^2} + \frac{\xi}{2} \|x\|_2^2}_{\approx \text{TV}(x)} \right)$$

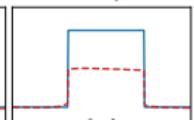
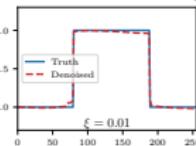


- ▶ Smooth and strongly convex
- ▶ Solution depends on choices of  $\alpha$ ,  $\nu$  and  $\xi$

Vary  $\nu$  ( $\alpha = 1$ ,  $\xi = 10^{-3}$ )



Vary  $\xi$  ( $\alpha = 1$ ,  $\nu = 10^{-3}$ )



How to choose all these parameters?

# Bilevel learning for inverse problems

**Upper level** (learning):

Given  $(x, y), y = Ax + \varepsilon$ , solve

$$\min_{\lambda \geq 0, \hat{x}} \|\hat{x} - x\|_2^2$$

**Lower level** (solve inverse problem):

$$\hat{x} \in \arg \min_z \{\mathcal{D}(Az, y) + \lambda \mathcal{R}(z)\}$$

von Stackelberg 1934, Kunisch and Pock '13, De los Reyes and Schönlieb '13

# Bilevel learning for inverse problems

**Upper level** (learning):

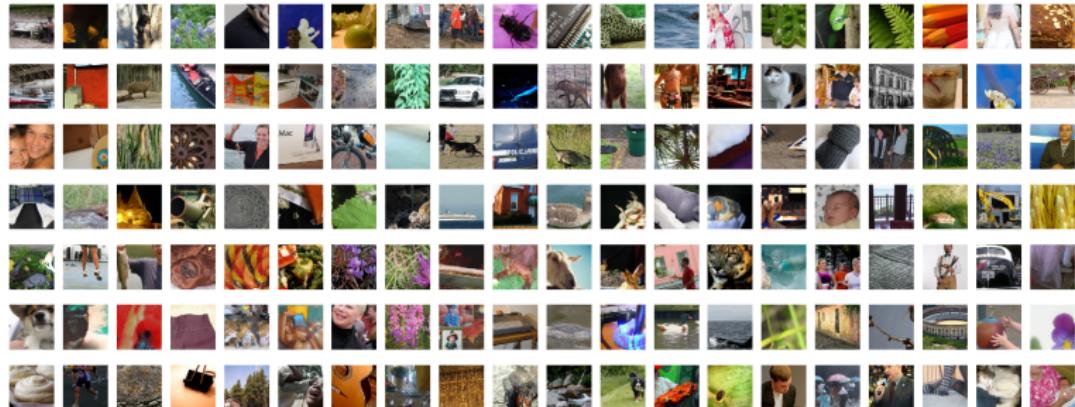
Given  $(x_i, y_i)_{i=1}^n, y_i = Ax_i + \varepsilon_i$ , solve

$$\min_{\lambda \geq 0, \hat{x}_i} \frac{1}{n} \sum_{i=1}^n \|\hat{x}_i - x_i\|_2^2$$

**Lower level** (solve inverse problem):

$$\hat{x}_i \in \arg \min_z \{\mathcal{D}(Az, y_i) + \lambda \mathcal{R}(z)\}$$

von Stackelberg 1934, Kunisch and Pock '13, De los Reyes and Schönlieb '13



# Inexact Algorithms for Bilevel Learning

## Bilevel learning: Reduced formulation

**Upper level:**

$$\min_{\lambda, \hat{x}} U(\hat{x})$$

**Lower level:**

$$\hat{x} = \arg \min_z L(z, \lambda)$$

# Bilevel learning: Reduced formulation

**Upper level:**

$$\min_{\lambda, \hat{x}} U(\hat{x})$$

**Lower level:**

$$\hat{x}(\lambda) := \hat{x} = \arg \min_z L(z, \lambda)$$

**Reduced formulation:**  $\min_{\lambda} U(\hat{x}(\lambda)) =: \tilde{U}(\lambda)$

# Bilevel learning: Reduced formulation

**Upper level:**

$$\min_{\lambda, \hat{x}} U(\hat{x})$$

**Lower level:**

$$\hat{x}(\lambda) := \hat{x} = \arg \min_z L(z, \lambda)$$

**Reduced formulation:**  $\min_{\lambda} U(\hat{x}(\lambda)) =: \tilde{U}(\lambda)$

$$0 = \partial_x^2 L(\hat{x}(\lambda), \lambda) \hat{x}'(\lambda) + \partial_\lambda \partial_x L(\hat{x}(\lambda), \lambda) \quad \Leftrightarrow \quad \hat{x}'(\lambda) = -B^{-1}A$$

$$\nabla \tilde{U}(\lambda) = (\hat{x}'(\lambda))^* \nabla U(\hat{x}(\lambda)) = -A^* w$$

where  $w$  solves  $Bw = \nabla U(\hat{x}(\lambda))$ .

# Algorithm for Bilevel learning

**Reduced formulation:**  $\min_{\lambda} U(\hat{x}(\lambda)) =: \tilde{U}(\lambda)$

- ▶ Compute gradients: Given  $\lambda$ 
  - (1) Compute  $\hat{x}(\lambda)$ , e.g. via PDHG Chambolle and Pock '11
  - (2) Solve  $Bw = \nabla U(\hat{x}(\lambda))$ ,  $B := \partial_x^2 L(\hat{x}(\lambda), \lambda)$  e.g. via CG
  - (3) Compute  $\nabla \tilde{U}(\lambda) = -A^* w$ ,  $A := \partial_\lambda \partial_x L(\hat{x}(\lambda), \lambda)$
- ▶ Solve reduced formulation via L-BFGS-B Nocedal and Wright '00

# Algorithm for Bilevel learning

**Reduced formulation:**  $\min_{\lambda} U(\hat{x}(\lambda)) =: \tilde{U}(\lambda)$

- ▶ Compute gradients: Given  $\lambda$ 
  - (1) Compute  $\hat{x}(\lambda)$ , e.g. via PDHG Chambolle and Pock '11
  - (2) Solve  $Bw = \nabla U(\hat{x}(\lambda))$ ,  $B := \partial_x^2 L(\hat{x}(\lambda), \lambda)$  e.g. via CG
  - (3) Compute  $\nabla \tilde{U}(\lambda) = -A^* w$ ,  $A := \partial_\lambda \partial_x L(\hat{x}(\lambda), \lambda)$
- ▶ Solve reduced formulation via L-BFGS-B Nocedal and Wright '00

This approach has a number of problems:

- ▶  $\hat{x}(\lambda)$  has to be computed
- ▶ Derivative assumes  $\hat{x}(\lambda)$  is exact minimizer
- ▶ Large system of linear equations has to be solved

# How to solve Bilevel Learning Problems?

- ▶ Ignore “problems”, just compute it. e.g. Sherry et al. '20
- ▶ Semi-smooth Newton: similar problems Kunisch and Pock '13
- ▶ Replace lower level problem by finite number of iterations of algorithms: not bilevel anymore Ochs et al. '15

**Use algorithm that acknowledges difficulties:**  
e.g. inexact DFO Ehrhardt and Roberts '21



Lindon Roberts

# Dynamic Accuracy Derivative Free Optimization

$$\min_{\theta} f(\theta)$$

**Key idea:** Use  $f_\epsilon$ :

$$|f(\theta) - f_\epsilon(\theta)| < \epsilon$$

Accuracy as low as possible, but as high as necessary.

E.g. if

$$f_{\epsilon^{k+1}}(\theta^{k+1}) < f_{\epsilon^k}(\theta^k) - \epsilon^k - \epsilon^{k+1},$$

then

$$f(\theta^{k+1}) < f(\theta^k)$$

# Dynamic Accuracy Derivative Free Optimization

$$\min_{\theta} f(\theta)$$

For  $k = 0, 1, 2, \dots$

- 1) Sample  $f_{\epsilon^k}$  in a neighbourhood of  $\theta_k$
- 2) Build model  $m_k(\theta) \approx f_{\epsilon^k}$
- 3) Minimise  $m_k$  around  $\theta_k$  to get  $\theta_{k+1}$
- 4) If model decrease is sufficient compared to function error: accept step

**Algorithm 1** Dynamic accuracy DFO algorithm for (22).

**Inputs:** Starting point  $\theta^0 \in \mathbb{R}^n$ , initial trust-region radius  $0 < \Delta^0 \leq \Delta_{\max}$ .

**Parameters:** strictly positive values  $\Delta_{\max}, \gamma_{\text{dec}}, \gamma_{\text{inc}}, \eta_1, \eta_2, \eta'_1, \epsilon$  satisfying  $\gamma_{\text{inc}} < 1 < \gamma_{\text{dec}}$ ,  $\eta_1 \leq \eta_2 < 1$ , and  $\eta'_1 < \min(\eta_1, 1 - \eta_2)/2$ .

- 1: Select an arbitrary interpolation set and construct  $m^0$  (26).
- 2: for  $k = 0, 1, 2, \dots$  do
- 3:   repeat
- 4:     Evaluate  $\tilde{f}(\theta^k)$  to sufficient accuracy that (32) holds with  $\eta'_1$  (using  $s^k$  from the previous iteration of this inner repeat/until loop). Do nothing in the first iteration of this repeat/until loop.
- 5:     if  $\|x^k\| \leq \epsilon$  then
- 6:       By replacing  $\Delta^k$  with  $\gamma'_{\text{dec}} \Delta^k$  for  $i = 0, 1, 2, \dots$ , find  $m^k$  and  $\Delta^k$  such that  $m^k$  is fully linear in  $B(\theta^k, \Delta^k)$  and  $\Delta^k \leq \|s^k\|$ . *[criticality phase]*
- 7:       end if
- 8:       Calculate  $s^k$  by (approximately) solving (27).
- 9:     until the accuracy in the evaluation of  $\tilde{f}(\theta^k)$  satisfies (32) with  $\eta'_1$  *[accuracy phase]*
- 10:   Evaluate  $\tilde{f}(\theta^k + s^k)$  so that (32) is satisfied with  $\eta'_1$  for  $\tilde{f}(\theta^k + s^k)$ , and calculate  $\hat{\rho}^k$  (29).
- 11:   Set  $\theta^{k+1}$  and  $\Delta^{k+1}$  as:
 
$$\theta^{k+1} = \begin{cases} \theta^k + s^k, & \hat{\rho}^k \geq \eta_2, \text{ or } \hat{\rho}^k \geq \eta_1 \text{ and } m^k \\ & \text{fully linear in } B(\theta^k, \Delta^k), \\ \theta^k, & \text{otherwise,} \end{cases} \quad (33)$$
- 12:   and
 
$$\Delta^{k+1} = \begin{cases} \min(\gamma_{\text{inc}} \Delta^k, \Delta_{\max}), & \hat{\rho}^k \geq \eta_2, \\ \Delta^k, & \hat{\rho}^k < \eta_2 \text{ and } m^k \text{ not} \\ & \text{fully linear in } B(\theta^k, \Delta^k), \\ \gamma_{\text{dec}} \Delta^k, & \text{otherwise.} \end{cases} \quad (34)$$
- 13: end for

## Theorem Ehrhardt and Roberts '21

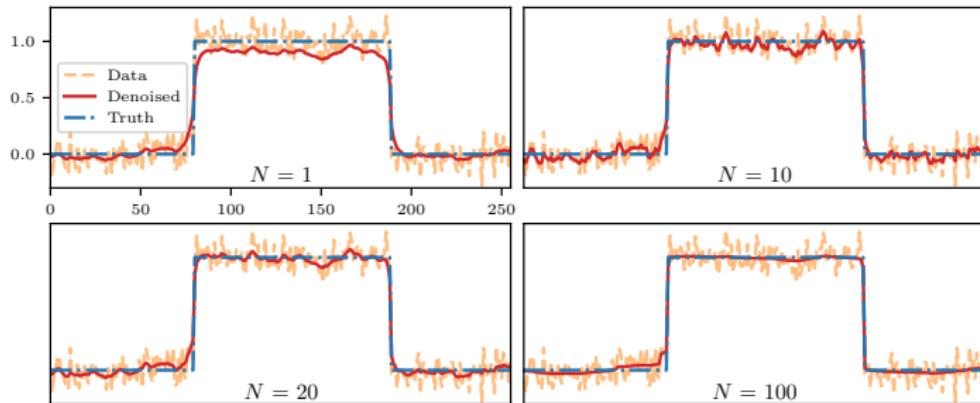
If  $f$  is sufficiently smooth and bounded below, then the algorithm is globally convergent in the sense that

$$\lim_{k \rightarrow \infty} \|\nabla f(\theta_k)\| = 0.$$

# 1D Denoising Problem (learn $\alpha$ , $\nu$ and $\xi$ ) Ehrhardt and Roberts '21

$$\min_{\theta} \left\{ \frac{1}{2} \sum_i \|\hat{x}_i(\theta) - x_i\|_2^2 + \beta \kappa^2(\theta) \right\}, \quad \theta = (\alpha, \nu, \xi)$$

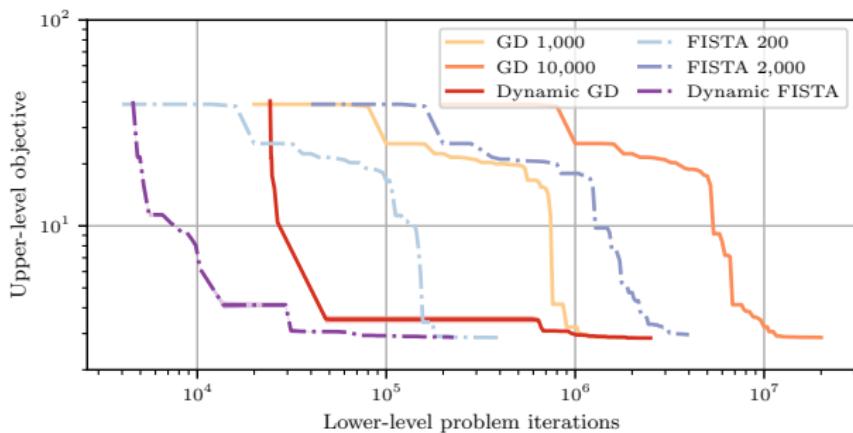
$$\hat{x}_i(\theta) = \arg \min_z \left\{ \frac{1}{2} \|z - y_i\|_2^2 + \alpha \left( \sum_j \sqrt{\|(\nabla z)_j\|_2^2 + \nu^2} + \frac{\xi}{2} \|z\|_2^2 \right) \right\}$$



**Reconstruction of  $\hat{x}_1$  after  $N$  evaluations of  $f(\theta)$**

# 1D Denoising Problem (learn $\alpha$ , $\nu$ and $\xi$ ) Ehrhardt and Roberts '21

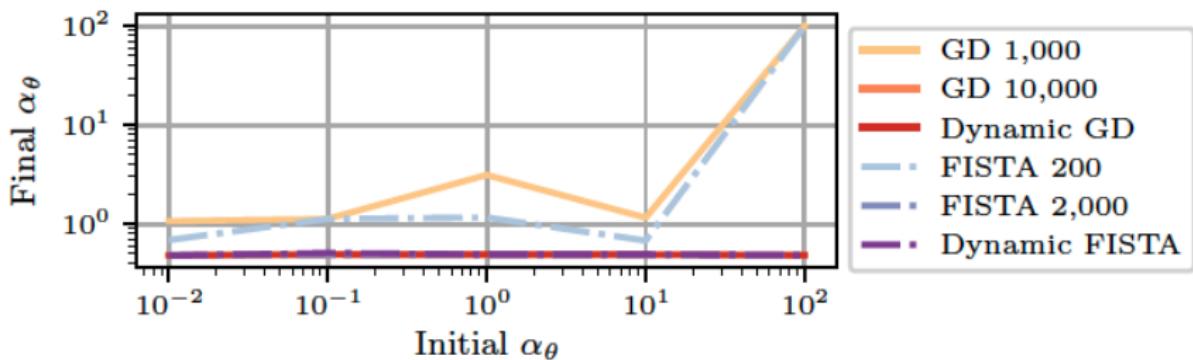
Dynamic accuracy is faster than “fixed accuracy”: **10x speedup**:



**Objective value  $f(\theta)$  vs. computational effort**

# 1D Denoising Problem Ehrhardt and Roberts '21

Always learns the same parameter for sufficient accuracy.



**Robustness to initialization**

**Learn sampling pattern in MRI**

# Learn sampling pattern in MRI



Ferdia Sherry

**Upper level** (learning):

Given **training data**  $(x_i, y_i)_{i=1}^n$ , solve

$$\min_{\lambda \geq 0, s \in \{0,1\}^m} \frac{1}{n} \sum_{i=1}^n \|\hat{x}_i(\lambda, s) - x_i\|_2^2 + \beta_1 \sum_{j=1}^m s_j$$

**Lower level** (MRI reconstruction):

$$\hat{x}_i(\lambda, s) = \arg \min_z \left\{ \sum_{j=1}^N s_j^2 |(Fz - y_i)_j|^2 + \lambda \mathcal{R}(z) \right\} \quad s_j \in \{0, 1\}$$

Sherry et al. '20

# Learn sampling pattern in MRI



Ferdia Sherry

**Upper level** (learning):

Given **training data**  $(x_i, y_i)_{i=1}^n$ , solve

$$\min_{\lambda \geq 0, s \in [0,1]^m} \frac{1}{n} \sum_{i=1}^n \|\hat{x}_i(\lambda, s) - x_i\|_2^2 + \beta_1 \sum_{j=1}^m s_j + \beta_2 \sum_{j=1}^m s_j(1 - s_j)$$

**Lower level** (MRI reconstruction):

$$\hat{x}_i(\lambda, s) = \arg \min_z \left\{ \sum_{j=1}^N s_j^2 |(Fz - y_i)_j|^2 + \lambda \mathcal{R}(z) \right\} \quad s_j \in [0, 1]$$

Sherry et al. '20

# Warm up

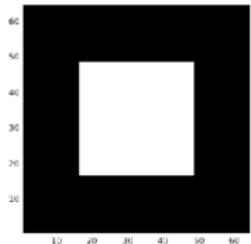
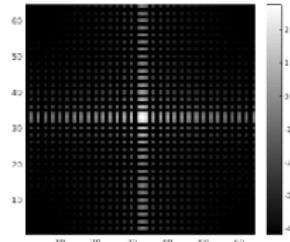
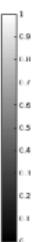
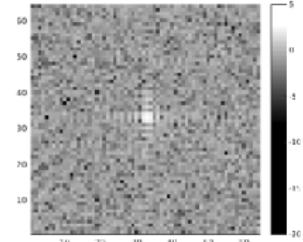


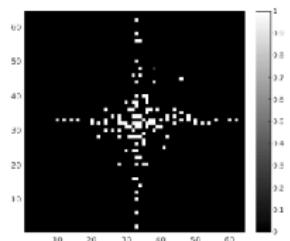
Figure: Discrete 2d bump



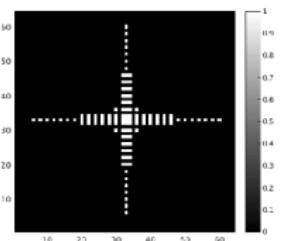
(a) Original data:  $\log |y|$



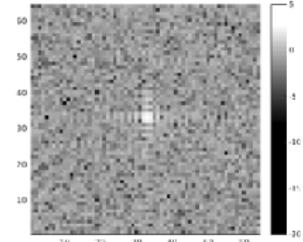
(b) Noisy data:  $\log |\tilde{y}|$



(c) Learned sampling pattern



(d) Largest 2.76% Fourier Coefficients



# Warm up

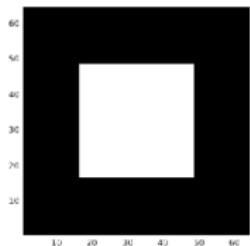
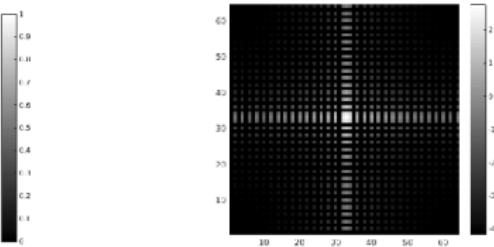
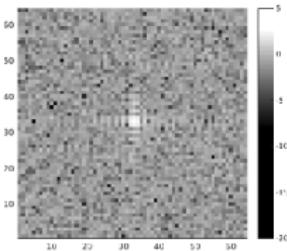


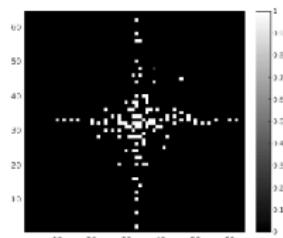
Figure: Discrete 2d bump



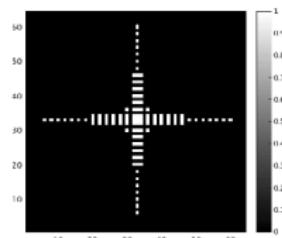
(a) Original data:  $\log |y|$



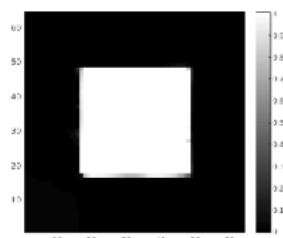
(b) Noisy data:  $\log |\tilde{y}|$



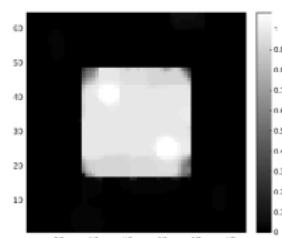
(c) Learned sampling pattern



(d) Largest 2.76% Fourier Coefficients

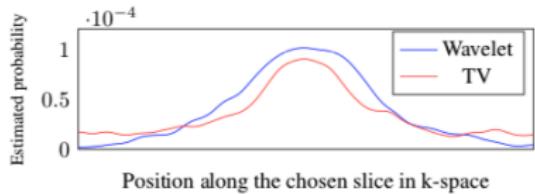
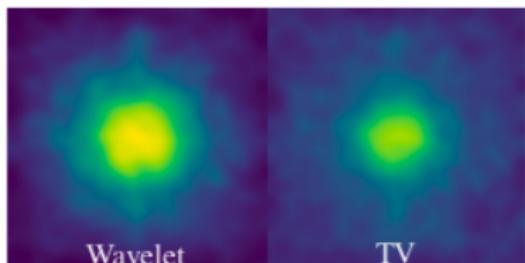
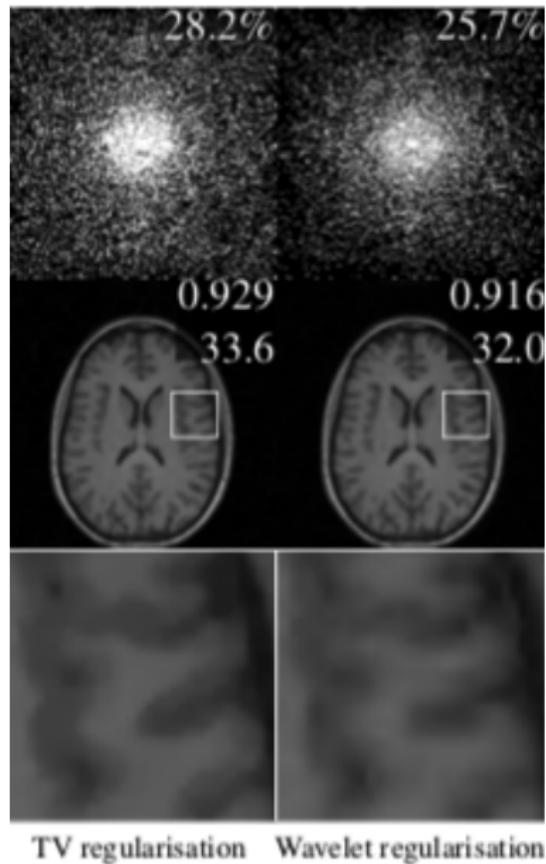


(e) Learned sampling pattern

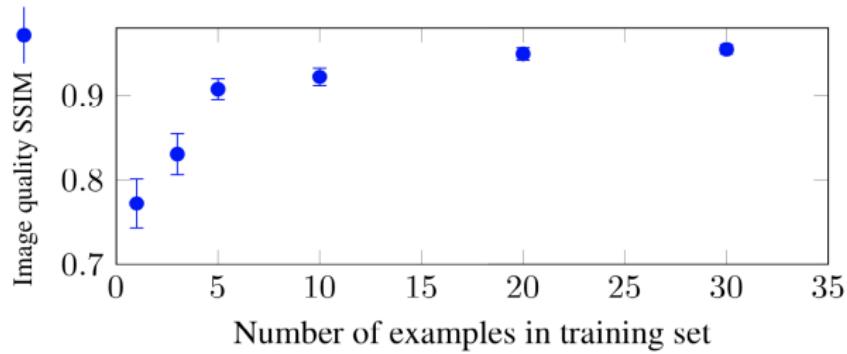


(f) Largest 2.76% Fourier Coefficients

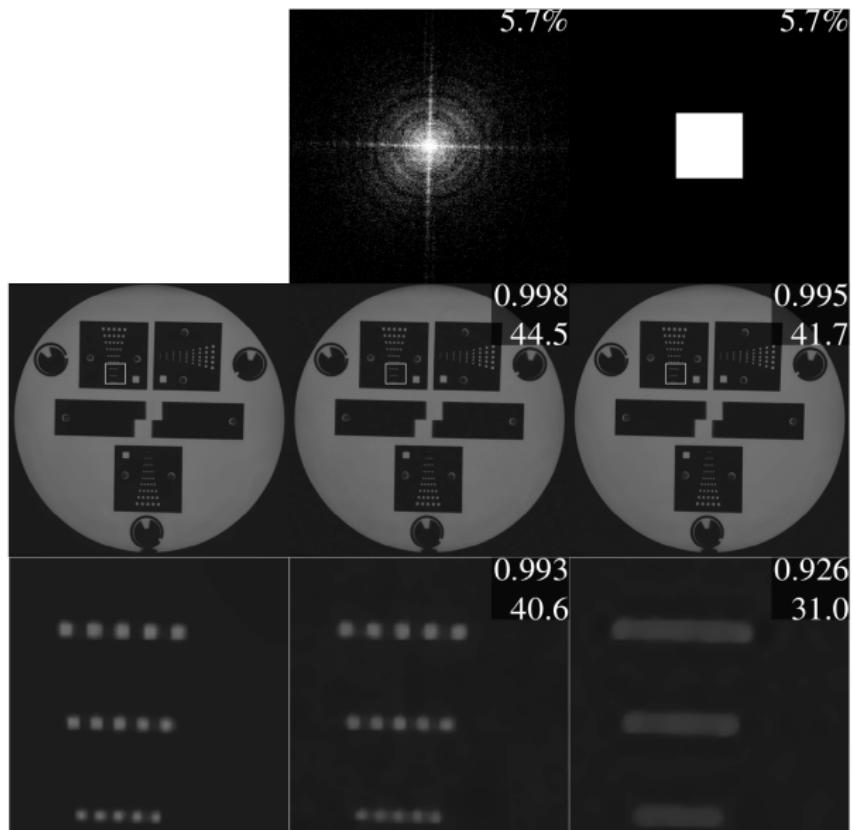
# Compare regularizers Sherry et al. '20



## More insights: sampling and number of data [Sherry et al. '20](#)



# High resolution imaging: $1024^2$ Sherry et al. '20



## Conclusions

- ▶ **Bilevel learning:** supervised learning for variational regularization
- ▶ **Accuracy** in the optimization algorithm is important
- ▶ **High-dimensional** parametrizations can be learned