Inexact Algorithms for Bilevel Learning

Matthias J. Ehrhardt

Department of Mathematical Sciences, University of Bath, UK

14 January, 2025

Joint work with:

M. S. Salehi, H. S. Wong (both Bath),
S. Mukherjee (Kharagpur), L. Roberts (Sydney),
L. Bogensperger (Zurich), T. Pock (Graz)

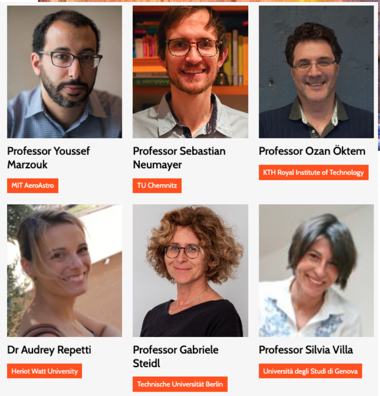
Mohammed Sadegh Salehi

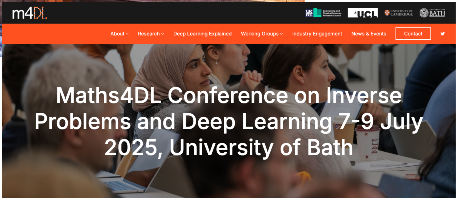
Hok Shing Wong

Lea Bogensperger

Engineering and Physical Sciences Research Council

Inverse Problems and Deep Learning: 7-9 July 2025

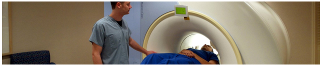




Deadline: 28 Feb 2025

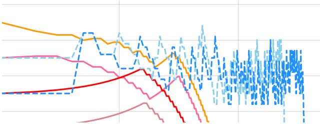
Outline

1) Bilevel learning of a regularizer



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

2) Inexact learning strategy



3) Numerical results

4) Inexact Piggyback

Inverse problems and Variational Regularization

$$Ax = y$$

x : desired solution

y : observed data

A : mathematical model

Goal: recover x given y

Variational regularization

Approximate a solution x^* of $Ax = y$ via

$$\hat{x} \in \arg\min_x \left\{ \mathcal{D}(Ax, y) + \lambda \mathcal{R}(x) \right\}$$

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

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

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

Scherzer et al. '08, Ito and Jin '15, Benning and Burger '18

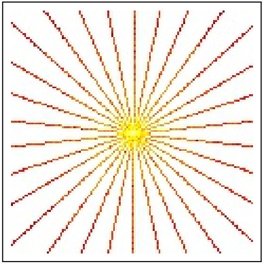
Example: Magnetic Resonance Imaging (MRI)

MRI Reconstruction Lustig et al. '07

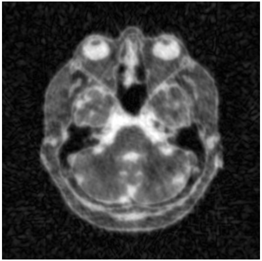
Fourier transform F, sampling $Sw = (w_i)_{i \in \Omega}$

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

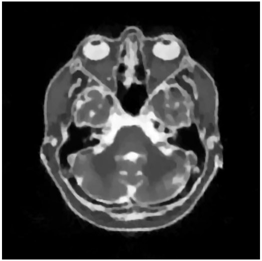
MRI scanner



data



poor choice



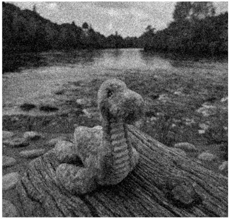
good choice

More Complicated Regularizers

Fields-of-Experts (FoE) Roth and Black ’05

$$\mathcal{R}(x) = \sum_{k=1}^{K} \lambda_k \|\kappa_k * x\|_{\gamma_k}$$

E.g. 48 kernels $7 \times 7 = 2448$ parameters



noisy

poor choice

well-trained

More Complicated Regularizers

Fields-of-Experts (FoE) Roth and Black '05

$$\mathcal{R}(x) = \sum_{k=1}^{K} \lambda_k \|\kappa_k * x\|_{\gamma_k}$$

E.g. 48 kernels $7 \times 7 = 2448$ parameters

Input Convex Neural Networks (ICNN) Amos et al. '17, Mukherjee et al. '24'

$$\mathcal{R}(x) = z_K,$$
$$z_{k+1} = \sigma(W_k z_k + V_k x + b_k), k = 0, \dots, K-1, z_0 = x$$

constraints on σ and W_k, e.g. 2 layers, 2000 parameters

- Convex Ridge Regularizers (CRR) Goujon et al. '22, ≈ 4000 parameters
- ...

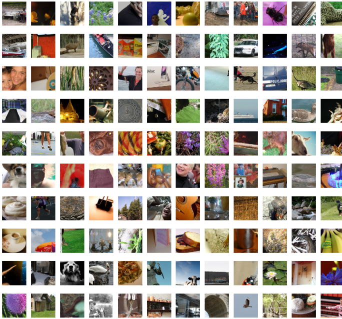
Bilevel learning for inverse problems

Upper level (learning):
Given $(x_i, y_i)_{i=1}^n, y_i \approx Ax_i$, solve

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \|\hat{x}_i(\theta) - x_i\|_2^2$$

Lower level (solve inverse problem):

$$\hat{x}_i(\theta) = \arg\min_x \{\mathcal{D}(Ax, y_i) + \mathcal{R}_\theta(x)\}$$



von Stackelberg 1934, 2003, Haber and Tenorio '03, Kunisch and Pock '13,

De los Reyes and Schönlieb '13, Crocket and Fessler '22, De los Reyes and Villacis '23

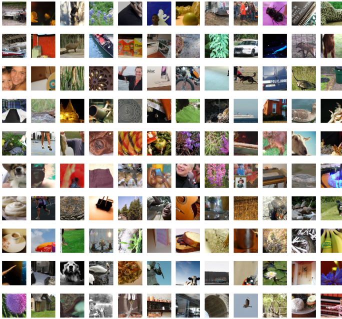
Bilevel learning for inverse problems

Upper level (learning):
Given $(x_i, y_i)_{i=1}^n$, $y_i \approx Ax_i$, solve

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \|\hat{x}_i(\theta) - x_i\|_2^2$$

Lower level (solve inverse problem):

$$\hat{x}_i(\theta) = \arg\min_x \{\mathcal{D}(Ax, y_i) + \mathcal{R}_\theta(x)\}$$



von Stackelberg 1934, 2003, Haber and Tenorio '03, Kunisch and Pock '13, De los Reyes and Schönlieb '13, Crocket and Fessler '22, De los Reyes and Villacis '23

- contrastive learning Hinton '02
- fitting prior distribution Roth and Black '05
- adversarial training Arjovsky et al. '17
- adverserial regularization Lunz et al. '18
- ...

Inexact Learning Strategy

Exact Approaches for Bilevel learning

Upper level: $$\min_{\theta} f(\theta) := g(\hat{x}(\theta))$$

Lower level: $$\hat{x}(\theta) := \arg\min_{x} h(x, \theta)$$

Access to **function values** $f(\theta)$

1) Compute $\hat{x}(\theta)$
2) Evaluate $f(\theta) := g(\hat{x}(\theta))$

Exact Approaches for Bilevel learning

Upper level: $$\min_{\theta} f(\theta) := g(\hat{x}(\theta))$$

Lower level: $$\hat{x}(\theta) := \arg\min_{x} h(x, \theta)$$

Access to **gradients** $\nabla f(\theta)$

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

$$\nabla f(\theta) = (\hat{x}'(\theta))^* \nabla g(\hat{x}(\theta)) = -A^* w, \quad \text{with } Bw = b$$

$$A = \partial_\theta \partial_x h(\hat{x}(\theta), \theta), \quad B = \partial_x^2 h(\hat{x}(\theta), \theta), \quad b = \nabla g(\hat{x}(\theta))$$

1) Compute $\hat{x}(\theta)$
2) Solve $Bw = b$
3) Compute $\nabla f(\theta) = -A^* w$

Exact Approaches for Bilevel learning

Upper level: $\min_{\theta} f(\theta) := g(\hat{x}(\theta))$

Lower level: $\hat{x}(\theta) := \arg\min_{x} h(x, \theta)$

Access to **gradients** $\nabla f(\theta)$

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

$$\nabla f(\theta) = (\hat{x}'(\theta))^* \nabla g(\hat{x}(\theta)) = -A^* w, \quad \text{with } Bw = b$$

$$A = \partial_\theta \partial_x h(\hat{x}(\theta), \theta), \quad B = \partial_x^2 h(\hat{x}(\theta), \theta), \quad b = \nabla g(\hat{x}(\theta))$$

1) Compute $\hat{x}(\theta)$
2) Solve $Bw = b$
3) Compute $\nabla f(\theta) = -A^* w$

This strategy has a number of problems:

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

Inexact Approaches for Bilevel learning

Upper level: $\min_{\theta} f(\theta) := g(\hat{x}(\theta))$

Lower level: $\hat{x}(\theta) := \arg\min_{x} h(x, \theta)$

Approximate function values $f_\varepsilon(\theta) \approx f(\theta)$:

1) Compute $\hat{x}_\varepsilon(\theta)$ to ε accuracy: $|\hat{x}_\varepsilon(\theta) - \hat{x}(\theta)| < \varepsilon$
2) Evaluate $f_\varepsilon(\theta) := g(\hat{x}_\varepsilon(\theta))$

Approximate gradients $z(\theta) \approx \nabla f(\theta)$:

$$A_\varepsilon = \partial_\theta \partial_x h(\hat{x}_\varepsilon(\theta), \theta), \quad B_\varepsilon = \partial_x^2 h(\hat{x}_\varepsilon(\theta), \theta), \quad b_\varepsilon = \nabla g(\hat{x}_\varepsilon(\theta))$$

1) Compute $\hat{x}_\varepsilon(\theta)$ to ε accuracy: $|\hat{x}_\varepsilon(\theta) - \hat{x}(\theta)| < \varepsilon$
2) Solve $B_\varepsilon w = b_\varepsilon$ to δ accuracy: $\|B_\varepsilon w - b_\varepsilon\| < \delta$
3) Compute $z(\theta) = -A_\varepsilon^* w$

Construction of Inexact Algorithms

Wish list:

- use gradients
- adaptive step-sizes (e.g. via backtracking): as large as possible as small as necessary, **maximize progress**
- adaptive accuracy: as low as possible as high as necessary, **minimize compute**

Construction of Inexact Algorithms

Wish list:

- use gradients
- adaptive step-sizes (e.g. via backtracking): as large as possible as small as necessary, **maximize progress**
- adaptive accuracy: as low as possible as high as necessary, **minimize compute**

Existing algorithms:

1) Zero-order: DFO-LS Ehrhardt and Roberts '21
   - adaptive accuracy using recent research in derivative-free optimization
   - does not scale well due to lack of gradients
2) First-order: HOAG Pedregosa '16
   - A-prior chosen accuracy ε_k
   - Convergence with stepsize $\alpha = 1/L$

Construction of Inexact Algorithms

Wish list:

- use gradients
- adaptive step-sizes (e.g. via backtracking): as large as possible as small as necessary, **maximize progress**
- adaptive accuracy: as low as possible as high as necessary, **minimize compute**

Existing algorithms:

1) Zero-order: DFO-LS Ehrhardt and Roberts '21
   - adaptive accuracy using recent research in derivative-free optimization
   - does not scale well due to lack of gradients
2) First-order: HOAG Pedregosa '16
   - A-prior chosen accuracy ε_k
   - Convergence with stepsize $\alpha = 1/L$

Ingredients:

- inexact gradient as descent direction
- inexact backtracking

Inexact Gradient as a Descent Direction

Assumptions:

- h is strongly convex and L_h-smooth
- g is L_g-smooth
- $\nabla_x^2 h(x, \theta)$ and $\nabla_{x\theta}^2 h(x, \theta)$ are Lipschitz

Lemma: Let $\|e_k\| \leq (1-\eta)\|z_k\|$, $\eta \in (0,1), e_k := z_k - \nabla f(\theta_k)$. Then $-z_k$ is a descent direction for f at θ_k.

Prop: Let $\hat{x}_k := \hat{x}_{\varepsilon_k}(\theta_k)$. There exists computable c_i:

$$\|e_k\| \leq c_1(\hat{x}_k)\varepsilon_k + c_2(\hat{x}_k)\delta_k + c_3\varepsilon_k^2 =: \omega_k$$

Inexact Gradient as a Descent Direction

Assumptions:

- h is strongly convex and L_h-smooth
- g is L_g-smooth
- $\nabla_x^2 h(x,\theta)$ and $\nabla_{x\theta}^2 h(x,\theta)$ are Lipschitz

Lemma: Let $\|e_k\| \leq (1-\eta)\|z_k\|$, $\eta \in (0,1), e_k := z_k - \nabla f(\theta_k)$. Then $-z_k$ is a descent direction for f at θ_k.

Prop: Let $\hat{x}_k := \hat{x}_{\varepsilon_k}(\theta_k)$. There exists computable c_i:

$$\|e_k\| \leq c_1(\hat{x}_k)\varepsilon_k + c_2(\hat{x}_k)\delta_k + c_3\varepsilon_k^2 =: \omega_k$$

1) Given ε_k, δ_k, compute $\hat{x}_k, z_k$ and ω_k
2) If $\omega_k > (1-\eta)\|z_k\|$, go to step 1) with smaller ε_k, δ_k

Theorem: If $\|\nabla f(\theta_k)\| > 0$, then z_k is a descent direction for all sufficiently small ε_k, δ_k.

Sufficient Decrease with Inexact Gradients

$$\theta_{k+1} = \theta_k - \alpha_k z_k$$

- $U_{k+1} := g(\hat{x}_{k+1}) + \|\nabla g(\hat{x}_{k+1})\| \varepsilon_{k+1} + \frac{L_{\nabla g}}{2} \varepsilon_{k+1}^2 \geq f(\theta_{k+1})$
- $L_k := g(\hat{x}_k) - \|\nabla g(\hat{x}_k)\| \varepsilon_k - \frac{L_{\nabla g}}{2} \varepsilon_k^2 \leq f(\theta_k)$

Theorem: If $U_{k+1} + \eta \alpha_k \|z_k\|^2 \leq L_k$, then

$$f(\theta_{k+1}) + \eta \alpha_k \|z_k\|^2 \leq f(\theta_k).$$

Sufficient Decrease with Inexact Gradients

$$\theta_{k+1} = \theta_k - \alpha_k z_k$$

- $U_{k+1} := g(\hat{x}_{k+1}) + \|\nabla g(\hat{x}_{k+1})\| \varepsilon_{k+1} + \frac{L_{\nabla g}}{2} \varepsilon_{k+1}^2 \geq f(\theta_{k+1})$
- $L_k := g(\hat{x}_k) - \|\nabla g(\hat{x}_k)\| \varepsilon_k - \frac{L_{\nabla g}}{2} \varepsilon_k^2 \leq f(\theta_k)$

Theorem: If $U_{k+1} + \eta \alpha_k \|z_k\|^2 \leq L_k$, then

$$f(\theta_{k+1}) + \eta \alpha_k \|z_k\|^2 \leq f(\theta_k).$$

Theorem: Let f be L_f-smooth and $\nabla f(\theta_k) \neq 0$.
If $\varepsilon_k, \varepsilon_{k+1} > 0$ are small enough, then there exists $\alpha_k > 0$, such that $U_{k+1} + \eta \alpha_k \|z_k\|^2 \leq L_k$.

Method of Adaptive Inexact Descent (MAID)

One iteration:

1) Compute inexact gradient z_k (possibly reducing ε_k, δ_k)
2) Attempt backtracking to compute α_k; if failed, go to step 1) with smaller ε_k, δ_k
3) Update estimate: $\theta_{k+1} = \theta_k - \alpha_k z_k$
4) Increase accuracies $\varepsilon_{k+1}, \delta_{k+1}$ and inital step size α_{k+1}

Method of Adaptive Inexact Descent (MAID)

One iteration:

1) Compute inexact gradient z_k (possibly reducing ε_k, δ_k)
2) Attempt backtracking to compute α_k; if failed, go to step 1) with smaller ε_k, δ_k
3) Update estimate: $\theta_{k+1} = \theta_k - \alpha_k z_k$
4) Increase accuracies $\varepsilon_{k+1}, \delta_{k+1}$ and inital step size α_{k+1}

Theorem: If $\nabla f(\theta_k) \neq 0$, then MAID updates θ_k in finite time.

Method of Adaptive Inexact Descent (MAID)

One iteration:

1) Compute inexact gradient z_k (possibly reducing ε_k, δ_k)
2) Attempt backtracking to compute α_k; if failed, go to step 1) with smaller ε_k, δ_k
3) Update estimate: $\theta_{k+1} = \theta_k - \alpha_k z_k$
4) Increase accuracies $\varepsilon_{k+1}, \delta_{k+1}$ and inital step size α_{k+1}

Theorem: If $\nabla f(\theta_k) \neq 0$, then MAID updates θ_k in finite time.

Theorem: Let f be bounded below. Then MAID's iterates θ_k satisfy $\|\nabla f(\theta_k)\| \to 0$.

Numerical Results

TV denoising: MAID vs DFO-LS (2 parameters)

$$h(x,\theta) = \frac{1}{2}\|x - y_t\|^2 + \underbrace{e^{\theta[1]} \sum_i \sqrt{|\nabla_1 x_i|^2 + |\nabla_2 x_i|^2 + (e^{\theta[2]})^2}}_{\text{smoothed TV}}$$

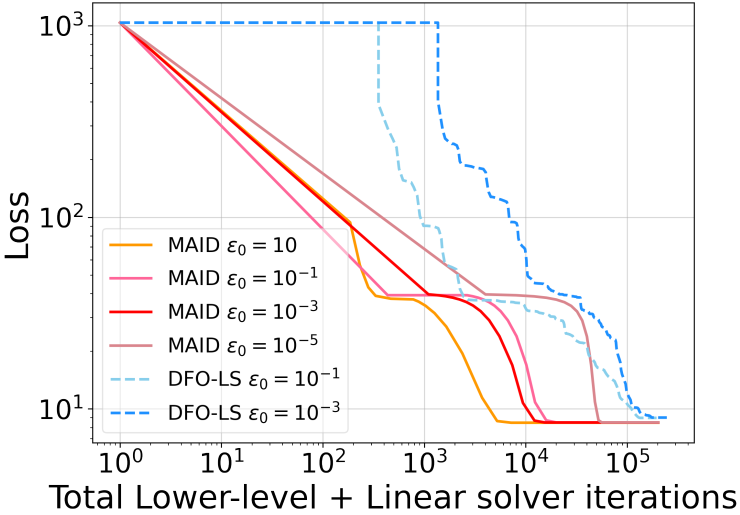
Noisy, PSNR=20.0

DFO-LS, 26.7

MAID, 26.9

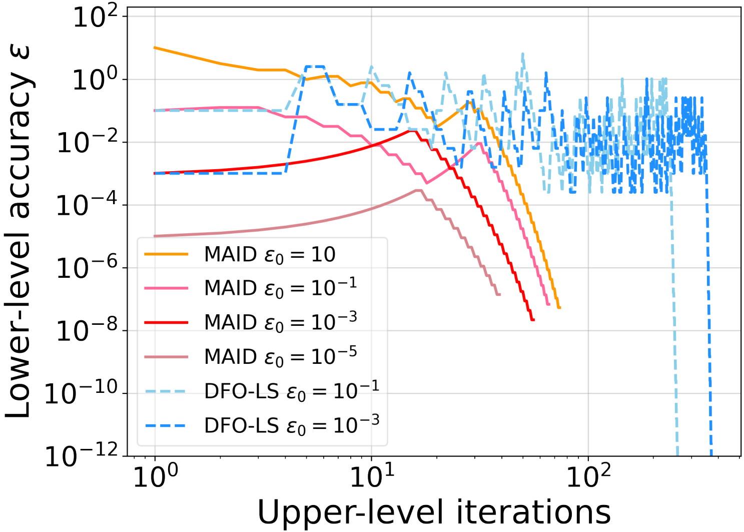
- similar image quality

TV denoising: MAID vs DFO-LS (2 parameters)



- Robustness to initial accuracy ε_0
- MAID particularly initially faster

TV denoising: MAID vs DFO-LS (2 parameters)



- MAID adapts accuracy, converge to same values in similar trend

FoE denoising: MAID vs HOAG ($\approx$ 2.5k parameters)

$$h(x,\theta) = \frac{1}{2}\|x-y\|^2 + e^{\theta[0]} \sum_{k=1}^{K} e^{\theta[k]} \|c_k * x\|_{\theta[K+k]}$$

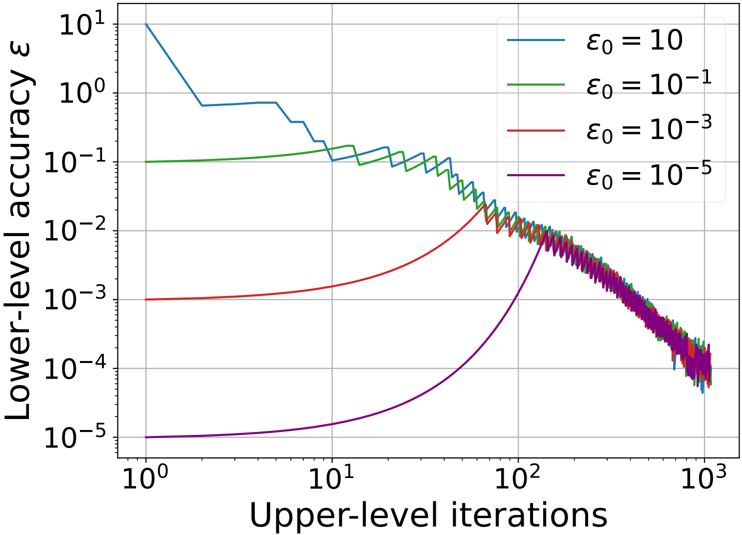
Noisy, PSNR=20.3

HOAG^2, 28.8

MAID, 29.7

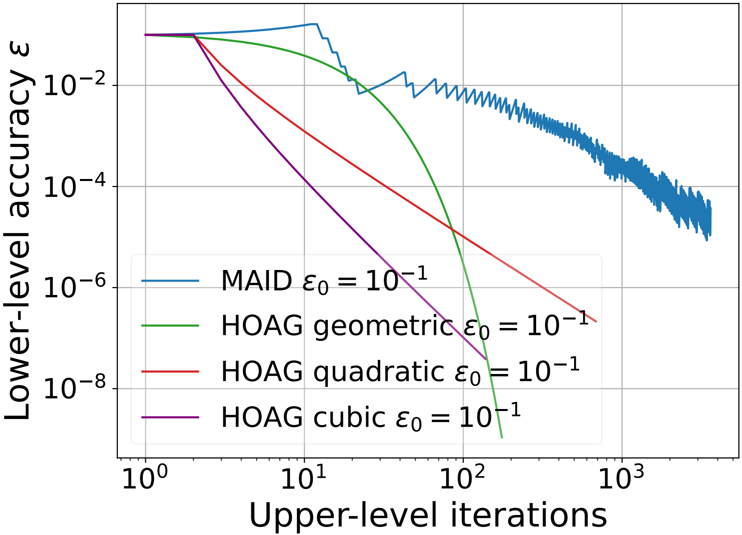
- MAID learns better regularizer than all HOAG variants; here best quadratic $\varepsilon_k = C/k^2$

FoE denoising: MAID vs HOAG ($\approx$ 2.5k parameters)



- MAID automatically tunes best accuracy schedule

FoE denoising: MAID vs HOAG ($\approx$ 2.5k parameters)



- accuracy schedule important; here slower decay better

Inexact Piggyback for Bilevel learning

Upper level: $\min_{\theta} \mathcal{L}(\theta) := \ell_1(\hat{x}(\theta)) + \ell_2(\hat{y}(\theta))$

Lower level: $\hat{x}(\theta), \hat{y}(\theta) := \arg\min_x \max_y \langle \theta x, y \rangle + g(x) - f^*(y)$

If g and f^* are regular enough, gradients can be computed via

$$\nabla \mathcal{L}(\theta) = \hat{y}(\theta) \otimes \hat{X}(\theta) + \hat{Y}(\theta) \otimes \hat{x}(\theta)$$

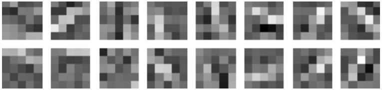
where $\hat{X}(\theta), \hat{Y}(\theta)$ solve another saddle-point problem of a similar form involving $\nabla^2 g(\hat{x}(\theta))$, $\nabla^2 f^*(\hat{y}(\theta))$, $\nabla \ell_1(\hat{x}(\theta))$ and $\nabla \ell_2(\hat{x}(\theta))$

Idea: this is of the same form as for MAID.

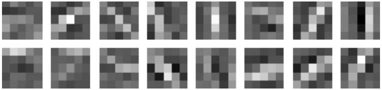
Problems of this form:

- learning discretisations of TV Chambolle and Pock '21
- training ICNNs after primal-dual reformulation Wong et al. '24

Learning TV discretisations



non-adaptive



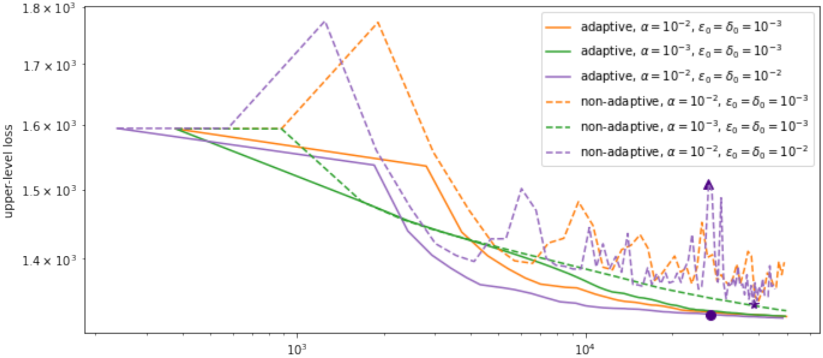
adaptive

standard TV
PSNR = 25.82 dB

non-adaptive
PSNR = 26.63 dB

adaptive
PSNR = 26.90 dB

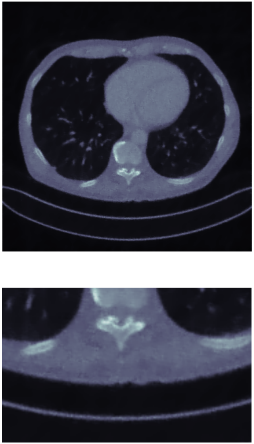
Learning TV discretisations II



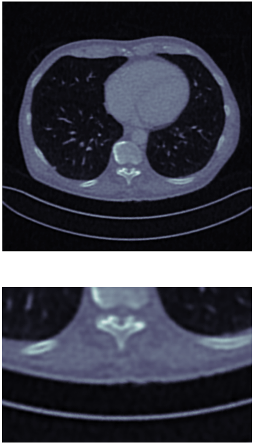
compute

- results still depend on parameters
- sensitivity much reduced

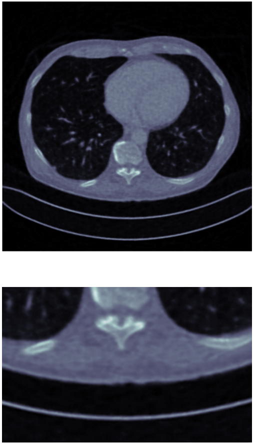
CT Reconstruction



ICNN-AR, PSNR=29.3



ICNN-Bilevel, 31.4



LPD, 34.2

Conclusions & Future Work

Conclusions

- **Bilevel learning**: supervised learning for variational regularization; computationally very hard
- **Accuracy** in the optimization algorithm is important; stability and efficiency
- **MAID** is a first-order algorithm with adaptive accuracies for descent and backtracking
- **High-dimensional** parametrizations can be learned; e.g. FoE, ICNN (a few thousand parameters)

Future work

- **Smart accuracy** schedule; can we disentangle accuracies ε, δ and step size α
- **Stochastic** variants for training from large data