Learning to See the Unseen: Data-Driven Regularization for Inverse Problems

Cong Shi

Johann Radon Institute for Computational and Applied Mathematics

Networking seminar for MLA2S
9 June, 2026.

Outline

Our institute

RICAM – the Johann Radon Institute for Computational and Applied Mathematics
Johann Karl August Radon:
16 December 1887 – 25 May 1956.
Achievements:

- Radon–Nikodym theorem;
- the Radon measure;
- the Radon transform;
- In 1954/55, he was rector of the University of Vienna.
- From 1948 to 1950, he was president of the Austrian Mathematical Society.
- ....

In 2003, the Austrian Academy of Sciences founded an Institute for Computational and Applied Mathematics and named it after Johann Radon.

Our institute

- **RICAM** was founded in 2003 by Heinz W. Engl.
- **Heinz W. Engl**: Won the Wilhelm Exner Medal (1998), Member of Academia Europaea (2013), Golden Medal of Honor (Goldenes Ehrenzeichen) of the Upper Austria (2023), Great Golden Medal (Großes Goldenes Ehrenzeichen) of Austria (2023), Great Silver Medal (Großes Silbernes Ehrenzeichen) of Vienna (2025)...

Working groups in RICAM:

- Computational Methods for PDEs, Inverse Problems and Mathematical Imaging, Mathematical Data Science, Mathematical Methods in Medicine and Life Sciences, Optimization and Optimal Control, Symbolic Computation, Transfer Group...

AI in our institute

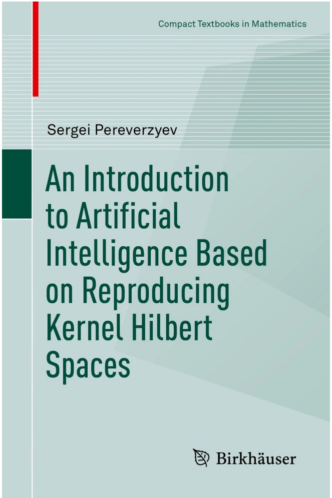
Mathematical Data Science (Group Leader: Philipp Grohs):
Designs and analyzes algorithms for problems in signal processing, the computational sciences, or computational finance.

- Machine learning based numerical methods for high dimensional PDEs.
- Robustness and accuracy of deep learning algorithms.

Inverse Problems and Mathematical Imaging (Group Leader: Otmar Scherzer):
Develops mathematical theory, computational methods, and data-driven approaches for extracting hidden information from indirect, incomplete, or noisy observations.

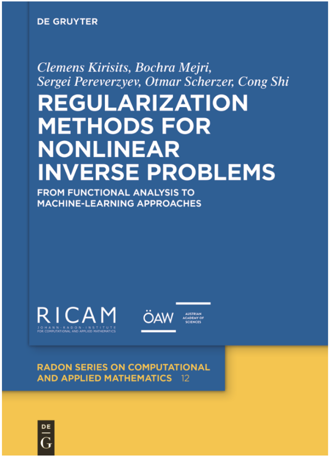
- Regularization theory for inverse problems.
- Mathematical imaging.
- Learning-based methods.

AI in our institute



- Adaptivity can be achieved at the level of observations (e.g., varying signals, image illumination), as well as at the level of domains (e.g., changing cameras, patient cohorts).
- The obtained results are presented in the book, which is reviewed in Zentralblatt MATH with the words: a very beautiful book...everyone interested in learning theory and regularization should, or rather must, read this book.

AI in our institute



- Nonlinear inverse problem
- Classical regularization theory.
- Neural networks functions.
- Function, functional and operator learning.
- Sequential learning.
- Training of neural networks.
- Hybrid regularization.

Outline

Biomedical imaging

Aim: To explore the body beneath the surface.
Examples:

- Diagnosis of disease.
- Help surgeons to plan the operation, and to make sure everything unwanted is removed.
- Perform biopsy in the operating room.

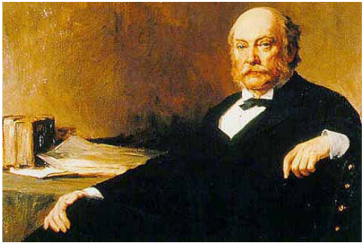

Figure: Ultrasound and biopsy.

Biomedical imaging

Waves interact with the tissue, e.g., by reflection or scattering.

Key concepts:

- Resolution: smallest detail that can be resolved.
- Robustness: stability of the image formation with respect to model uncertainty and electronic noise.
- Contrast: physical nature (eg. benign or malignant for tumors).



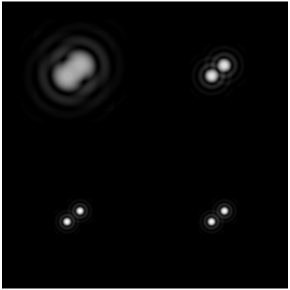


Figure: Left: Lord Rayleigh; Right: Rayleigh criterion for imaging resolution

Examples of biomedical imaging

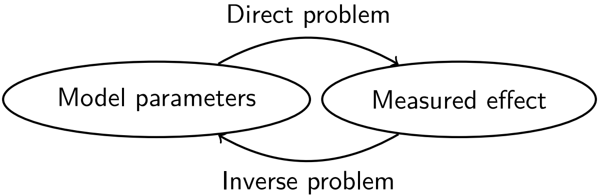
- Single-modality imaging, eg. computer tomography (CT), ultrasound imaging, optical coherence tomography (OCT), electric impedance tomography (EIT),...
- Hybrid imaging, eg. photoacoustic tomography (PAT), acousto-electric tomography (AET), ...

High Res. / High Con.	Ultrasound	Magnetic resonance imaging
Optical - OT	Photoacoustic tomography	
Electrical - EIT	UMEIT, AET	MREIT
Elastic	Transient elastography	Magnetic resonance elastography

Inverse problem

All the imaging process was created by solving a **mathematical puzzle**.
The puzzle has a name: **"inverse problem"**.

Inverse problem: from the seen effect to the unseen cause — like a patient's organs.



$$F[\underbrace{\mathrm{x}}_{\text{unseen}}] = \underbrace{\mathrm{y}^{\delta}}_{\text{seen}}.$$

Our focus:
Medical imaging + Inverse problems + data-driven methods + mathematical interpretation.

Example: Computer Tomography

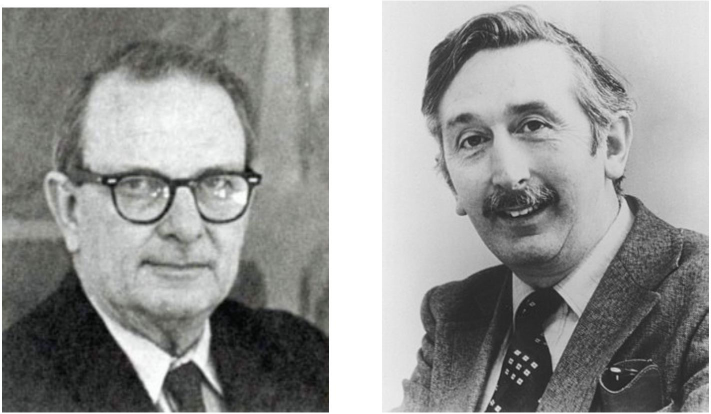


Figure: Allan M. Cormack and Godfrey N. Hounsfield.

- In 1972, the first CT setup was produced in England.
- In 1979, South African American physicist Allan M. Cormack and British electrical engineer Godfrey N. Hounsfield was awarded the Nobel Prize in Physiology or Medicine.

Example: Computer Tomography

Figure: The CT equipment invented by Hounsfield.

- In 1972, the first CT setup was produced in England.
- In 1979, South African American physicist Allan M. Cormack and British electrical engineer Godfrey N. Hounsfield was awarded the Nobel Prize in Physiology or Medicine.

Example: Computer Tomography

Figure: The current rotating CT equipment.

Example: Computer Tomography

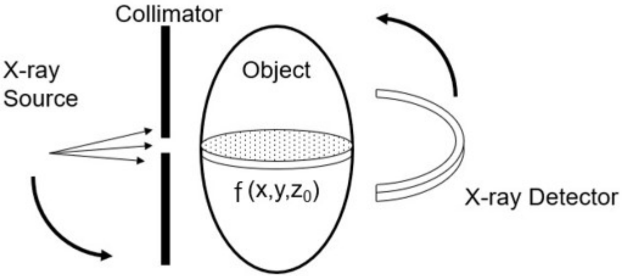
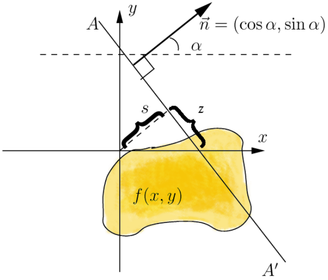


Figure: CT Imaging principle.

- Uses X-rays, but exposure is limited to a slice (or "a couple of" slices) by a collimator.
- Source and detector rotate around object – projections from many angles gives enough information.
- The desired image is computed from the projections.

Mathematical modelling for Computed Tomography (CT)



Let the desired image on a slice be $f(x, y)$. Then the CT measurements are given by line integrals:

$$\begin{aligned} m(s,\alpha) &= \int_{\mathbb{R}} f(x(z), y(z))\, dz \\ &= \int_{\mathbb{R}} f(z\sin\alpha + s\cos\alpha, z\cos\alpha - s\sin\alpha)\, dz \end{aligned}$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x, y)$ from measurements of $m(s, \alpha)$.

Mathematical modelling for Computed Tomography (CT)


Figure: Radon transform example 1.

Let the desired image on a slice be $f(x,y)$. Then the CT measurements are given by line integrals:

$$m(s,\alpha) = \int_{\mathbb{R}} f(x(z), y(z))\, dz$$
$$= \int_{\mathbb{R}} f(z\sin\alpha + s\cos\alpha, z\cos\alpha - s\sin\alpha)\, dz$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x,y)$ from measurements of $m(s,\alpha)$.

Mathematical modelling for Computed Tomography (CT)

Figure: Radon transform example 1.

Let the desired image on a slice be $f(x, y)$. Then the CT measurements are given by line integrals:

$$\begin{aligned} m(s, \alpha) &= \int_{\mathbb{R}} f(x(z), y(z))\, dz \\ &= \int_{\mathbb{R}} f(z \sin\alpha + s\cos\alpha, z\cos\alpha - s\sin\alpha)\, dz \end{aligned}$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x, y)$ from measurements of $m(s, \alpha)$.

Mathematical modelling for Computed Tomography (CT)

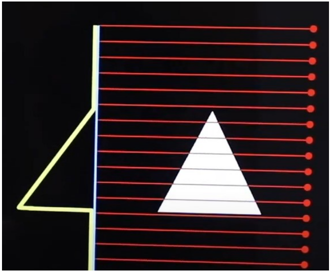


Figure: Radon transform example 1.

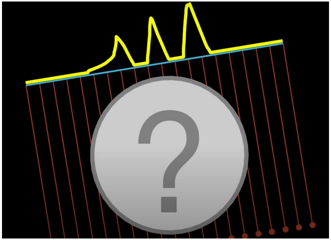
Let the desired image on a slice be $f(x, y)$. Then the CT measurements are given by line integrals:

$$\begin{aligned} m(s, \alpha) &= \int_{\mathbb{R}} f(x(z), y(z))\, dz \\ &= \int_{\mathbb{R}} f(z \sin\alpha + s\cos\alpha, z\cos\alpha - s\sin\alpha)\, dz \end{aligned}$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x, y)$ from measurements of $m(s, \alpha)$.

Mathematical modelling for Computed Tomography (CT)



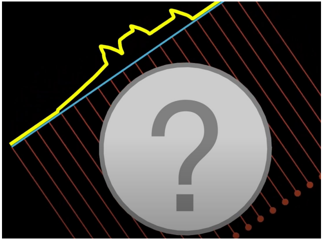


Figure: Radon transform example 2.

Let the desired image on a slice be $f(x, y)$. Then the CT measurements are given by line integrals:

$$\begin{aligned} m(s, \alpha) &= \int_{\mathbb{R}} f(x(z), y(z))\, dz \\ &= \int_{\mathbb{R}} f(z \sin \alpha + s \cos \alpha, z \cos \alpha - s \sin \alpha)\, dz \end{aligned}$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x, y)$ from measurements of $m(s, \alpha)$.

Mathematical modelling for Computed Tomography (CT)

Figure: Radon transform example 2.

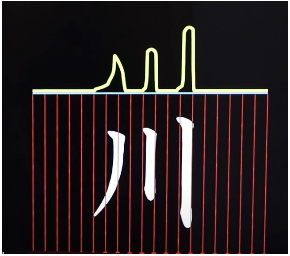
Let the desired image on a slice be $f(x, y)$. Then the CT measurements are given by line integrals:

$$
\begin{aligned}
m(s, \alpha) &= \int_{\mathbb{R}} f(x(z), y(z))\, dz \\
&= \int_{\mathbb{R}} f(z \sin\alpha + s\cos\alpha, z\cos\alpha - s\sin\alpha)\, dz
\end{aligned}
$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x, y)$ from measurements of $m(s, \alpha)$.

Mathematical modelling for Computed Tomography (CT)



Let the desired image on a slice be $f(x, y)$. Then the CT measurements are given by line integrals:

$$\begin{aligned} m(s, \alpha) &= \int_{\mathbb{R}} f(x(z), y(z))\, dz \\ &= \int_{\mathbb{R}} f(z \sin\alpha + s\cos\alpha, z\cos\alpha - s\sin\alpha)\, dz \end{aligned}$$

This is called *Radon transform*. This idea was first invented by Johann Radon in 1917.

- Inverse problem: calculate $f(x, y)$ from measurements of $m(s, \alpha)$.

Outline

What did we do in the book?

Book focus: Guideline for (non)linear inverse problem and data driven methods.
Use CT as an example:

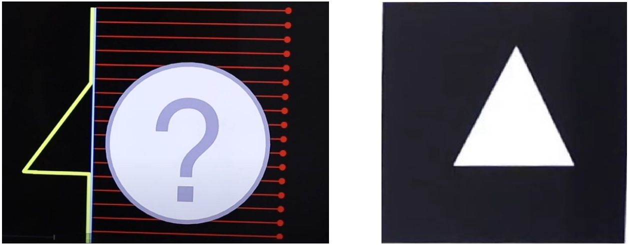


Figure: Left: measurements y^δ, Right: model parameters x

Formulate it mathematically:

$$F[\mathrm{x}] = \mathrm{y}^\delta,$$

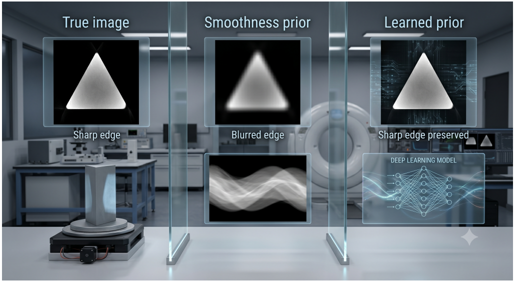
where F is the Radon transform.

Aim: $\min_{\mathrm{x}\in\mathrm{X}} \|F[\mathrm{x}] - \mathrm{y}^\delta\|_{\mathrm{Y}} + \alpha R(\mathrm{x})$.

Why neural networks?

In practice, CT images actually have sharp boundaries (bone/muscle), fine textures (blood vessels), sudden contrast changes.

Problem:
The mismatch: Smoothness priors penalize sharp edges as "unsmooth".



[1]

[1] Generated by Gemini.

Why neural networks?

1. New tools – Advantages for inverse problems

- Expressivity (edges, textures);
- Combine model and data information;
- Speed (one forward pass).

2. Inverse problem theory interprets NNs

The minimization functional in inverse problems:

$$\min_{\mathtt{x}\in\mathtt{X}} \|F[\mathtt{x}] - \mathtt{y}^\delta\|^2 + \alpha R(\mathtt{x}).$$

Inverse problems:

- Data fidelity $\|F[\mathtt{x}] - \mathtt{y}^\delta\|^2$;
- Regularizer $R(\mathtt{x})$;
- Solution $\mathtt{x}$.

Machine learning:

- Loss on training data;
- Regularization / prior;
- Model parameters / output.

NNs are not black boxes. Inverse problems give us the language to **interpret** them.

How to Use AI to Solve CT Reconstruction Problems and Get convergence analysis?

Approaches:

- **End to end learning**:
  Learn the inverse map $\mathtt{x} = F^{-1}(\mathtt{y}^{\delta})$ from the data directly.
- **Learn the regularizer**:
  Replace regularization term $R(\mathtt{x})$ with a learned prior (or learn the regularization parameter α).
- **Learning to learn** (Data-driven iterations):
  Iteration = Model update + Data update.

Outline

Shallow neural networks

Leaving the inverse problem aside for a moment...
Consider one layer (shallow) neural networks on $\mathbb{R}^n$,

$$G(\mathrm{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\vec{w}_j^T \mathrm{x} + \theta_j).$$

Questions: How to guarantee the output is what we want? convergence? convergence rates?

Tools for convergence:

- **Universal approximation Theorems**.

Classical result (Cybenko, 1989):
A single hidden layer neural network can approximate any continuous function arbitrarily well – given enough neurons.

Shallow neural networks

Leaving the inverse problem aside for a moment...
Consider one layer (shallow) neural networks on $\mathbb{R}^n$,

$$G(\mathrm{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\vec{w}_j^T \mathrm{x} + \theta_j).$$

Questions: How to guarantee the output is what we want? convergence? convergence rates?

Tools for convergence:

- **Tauber-Wiener Theorems (1932)**.

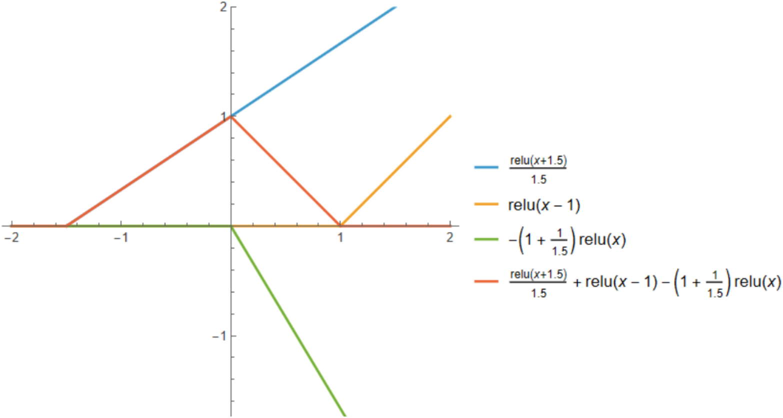
A single hidden layer neural network (without weights) can approximate any function from $L^1(\mathbb{R})$ or $L^2(\mathbb{R})$ arbitrarily well – given enough neurons.

Convergence rates

Tools for convergence rates:

Method 1: Finite element methods

- 3 ReLU can exactly represent a 1D linear spline;



Convergence rates

Method 2: Wavelet theories
To approximate any function (e.g., a CT image), we need basis functions – like Lego pieces – to assemble it.
Option 1: Classical basis (Fourier): Use sine and cosine waves – periodic, smooth, infinite.

- Good for smooth, repeating patterns.
- Bad for sharp edges (organ boundaries).

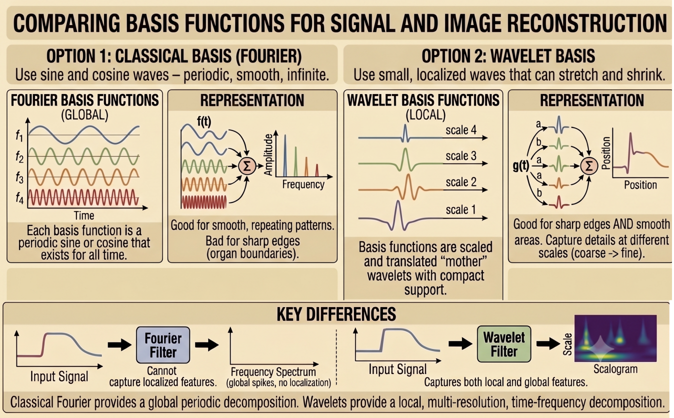
Option 2: Wavelet basis: Use small, localized waves that can stretch and shrink.

- Good for sharp edges and smooth areas.
- Capture details at different scales (coarse $\rightarrow$ fine).

Convergence rates

Method 2: Wavelet theories

To approximate any function (e.g., a CT image), we need basis functions – like Lego pieces – to assemble it.



[2]

[2] Generated by Gemini.

Fun history of these 3 tools

Tools for convergence:

- Universal approximation theorems (eg. Cybenko (1989), Hornik (1989), Deng & Han (2009));
- Tauber-Wiener Theorems (1932).

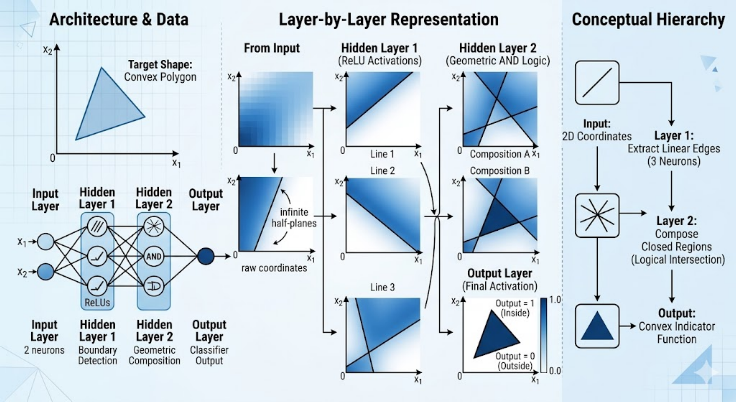
Tools for convergence rates:

- Finite element methods;
- Wavelet theories (eg. Haar wavelet (1909), Daubechies (1987)).

Figure: Historical timeline of important publications.

2 hidden layer neural network

- 1 hidden layer $\rightarrow$ Universal approximation (any function);
- 2 hidden layer $\rightarrow$ Same/ Better accuracy + fewer parameters.



[3]

---

[3]Generated by Gemini.

Outline

Linear case

Aim: Learn the inverse operator F^{-1} from data $(\mathrm{x}^1, \mathrm{y}^1), ..., (\mathrm{x}^{\mathbb{N}}, \mathrm{y}^{\mathbb{N}})$.

Simple approach using linear algebra:

Method	**What it does**
Orthonormalization	Make basis vectors orthonormal.
Bi-orthonormalization	Find two mutually orthogonal bases: for measurements and reconstructions.
Principal Component Analysis (PCA)	Identify directions of largest variance in the data; keep principal components.

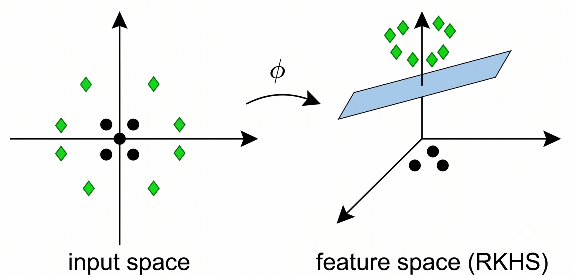
Issue: PCA (linear) fails for high-contrast CT.

Need: Nonlinear framework.

Nonlinear case

Tools: Reproducing Kernel Hilbert Space (RKHS).

- Lift data into a high-dimensional space where the relationship becomes linear;
- Then do PCA / linear algebra there.



Nonlinear case

Tools: Reproducing Kernel Hilbert Space (RKHS).

- Lift data into a high-dimensional space where the relationship becomes linear;
- Then do PCA / linear algebra there.

Short history for RKHS:

- Aronszajn (1930s) – Founded the theory.
- Wahba (1960s) – Brought to statistics (smoothing splines, regularization).
- Vapnik, Schölkopf (1990s) – Kernel methods explode in machine learning (Support Vector Machines, Kernel PCA).

Outline

The Hybrid Regularization – Three Approaches

Approach 1: Learn the prior from data instead of hand-crafting it.

$$\min \|F[\mathrm{x}] - \mathrm{y}^\delta\|_{\mathrm{Y}}^2 + \lambda \underbrace{\|P[\mathrm{x}] - \mathrm{y}^\delta\|^2}_{\text{learned feature}} + \alpha \underbrace{\|\mathrm{x} - \mathrm{x}^0\|^2}_{\text{regularizer}}$$

What the learned prior does:

- Treats sharp edges as normal (present in training data);
- No penalty for "unsmoothness";
- Penalizes only patterns unlikely in real organs.

Result:

- Sharp edges preserved + noise suppressed;
- Guarantee convergence + convergence rates.

The Hybrid Idea – Three Approaches

Tools: Reproducing Kernel Hilbert Space theory (RKHS).
Approach 2 (RKHS): Learn the regularization functional or parameter α using RKHS.

$$\min \|F[\mathrm{x}] - \mathrm{y}^\delta\|_{\mathrm{Y}}^2 + \alpha \underbrace{\|\mathrm{x} - \mathrm{x}^0\|^2}_{\text{learned regularizer}} .$$

Role of RKHS:

- Parameter α is selected using RKHS theory;
- Provides convergence guarantees and optimal rates.

The Hybrid Idea – Three Approaches

Approach 3: Data-driven iterative methods
Classical iterative methods:

- Iteratively regularized Landweber algorithm (IRLI);
- Iteratively regularized Gauss-Newton method (IRGN).

Key idea: Incorporate data-driven methods into the iteration.

Two Components per Iteration:

1. Model-driven update: uses the physical forward model F;
2. Data-driven update: uses a learned network or prior (improves quality, suppresses artifacts).

Conclusion

How to Use AI to Solve CT Reconstruction Problems and convergence analysis?

Approaches:

- **End to end learning**: learn the inverse map $\mathrm{x} = F^{-1}(\mathrm{y}^\delta)$ from data y^δ directly. (Chapter 5).
- **Learn the regularizer**: Replace regularization term $R(\mathrm{x})$ with a learned prior (or learn the regularization parameter α). (Chapter 8)
- **Learning to learn** (Data-driven iterations): model update + data update. (Chapter 9)

Basics: Neural network functions (Chapter 4), examples (Chapter 2), classical regularization theories (Chapter 3), functional analysis (Appendix).

Thank you very much for your attention!