Neural Networks Can Automatically Adapt to Low-Dimensional Structure in Inverse Problems

Suzanna Parkinson, Ph.D. Candidate
University of Chicago
Committee on Computational and Applied Mathematics

Brigham Young University Applied Math Seminar
September 4, 2025

https://arxiv.org/abs/2502.15522

ROBOTICS

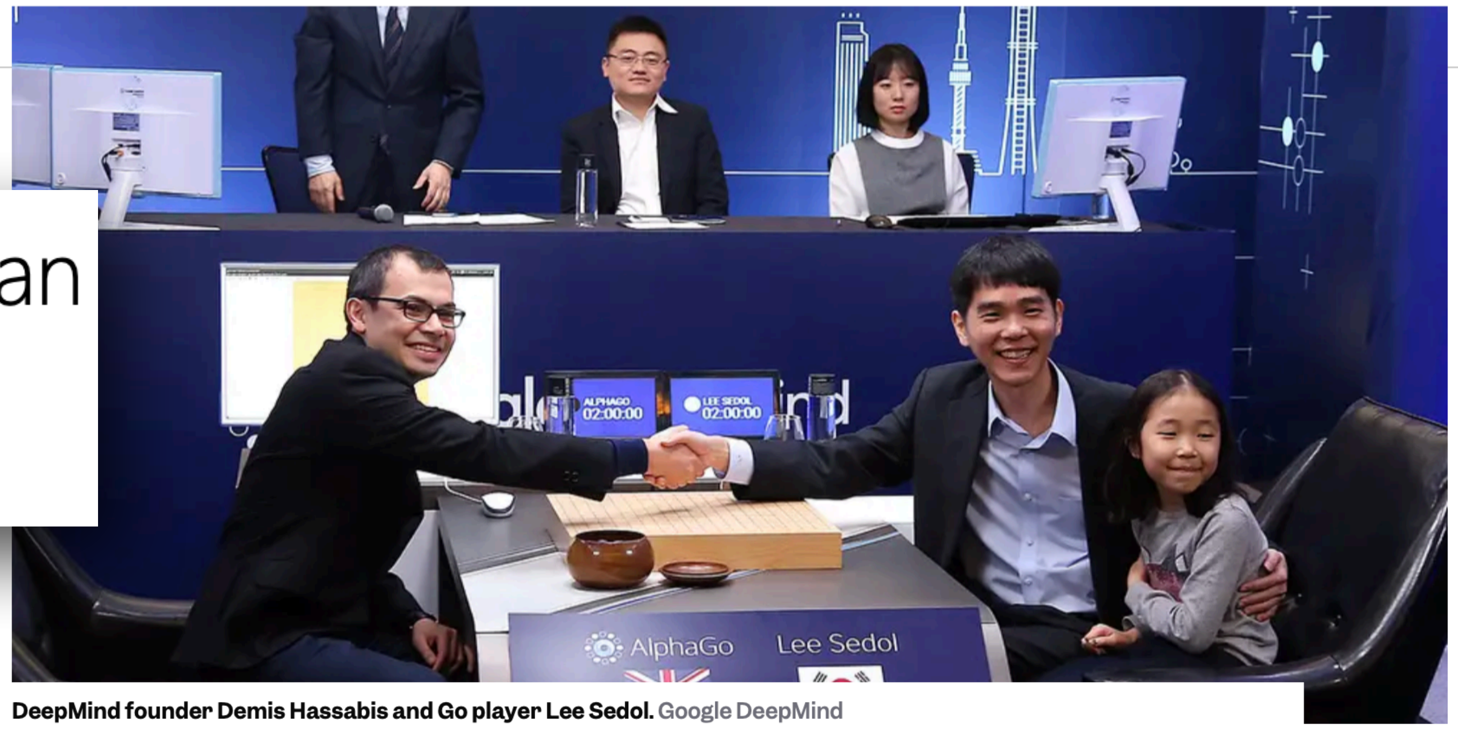
Computer Eyesight Gets a Lot More Accurate

BY JOHN MARKOFF AUGUST 18, 2014 8:01 PM

TECH

Google just made artificial-intelligence history

By Drake Baer



DeepMind founder Demis Hassabis and Go player Lee Sedol. Google DeepMind

Mar 9, 2016, 7:39 AM MT Share Save Add us on G

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

October 18, 2016 | Allison Linn

First supernova detected, confirmed, classified and shared by AI

New artificial intelligence tool removes humans from entire search, discovery process

October 13, 2023 | By Amanda Morris

A year after ChatGPT's release, the AI revolution is just beginning

By Catherine Thorbecke, CNN
8 min read · Updated 10:32 AM EST, Thu November 30, 2023

News & Views | Open access | Published: 25 November 2024

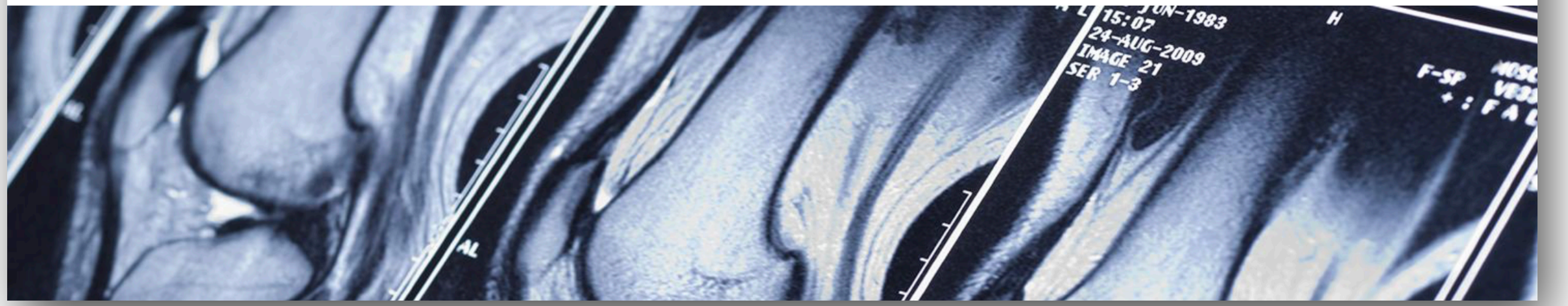
Artificial Intelligence awarded two Nobel Prizes for innovations that will shape the future of medicine

Ben Li & Stephen Gilbert

npj Digital Medicine **7**, Article number: 336 (2024) | Cite this article

RESEARCH, PRESS RELEASES, ARTIFICIAL INTELLIGENCE | AUGUST 18, 2020

New Research Finds FastMRI Scans Generated with Artificial Intelligence Are as Accurate as Traditional MRI



Why do **neural networks** work so well?

ROBOTICS

Computer Eyesight Gets a Lot More Accurate

BY JOHN MARKOFF AUGUST 18, 2014 8:01 PM

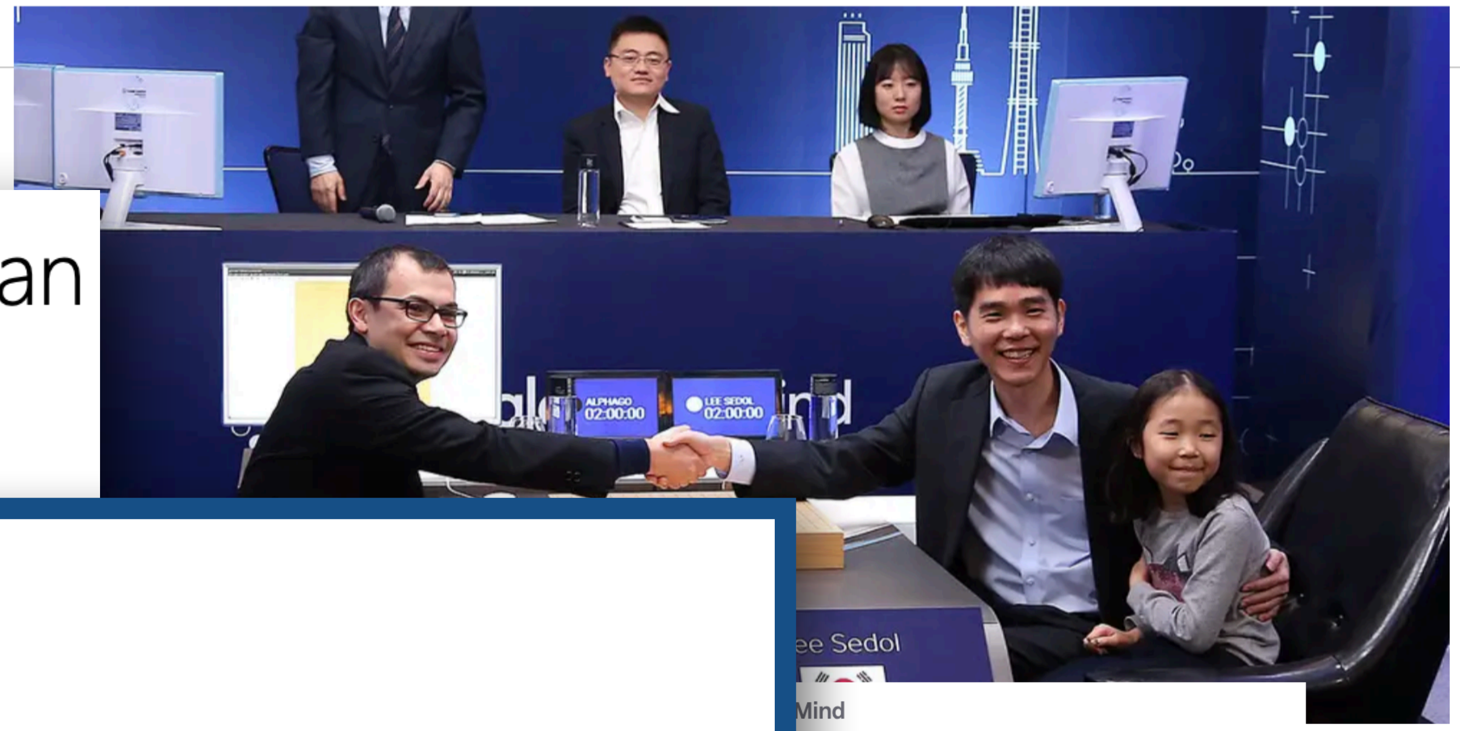
Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

October 18, 2016 | Allison Linn

TECH

Google just made artificial-intelligence history

By Drake Baer



Share Save Add us on

First
confirme

New artificial intel

revolution

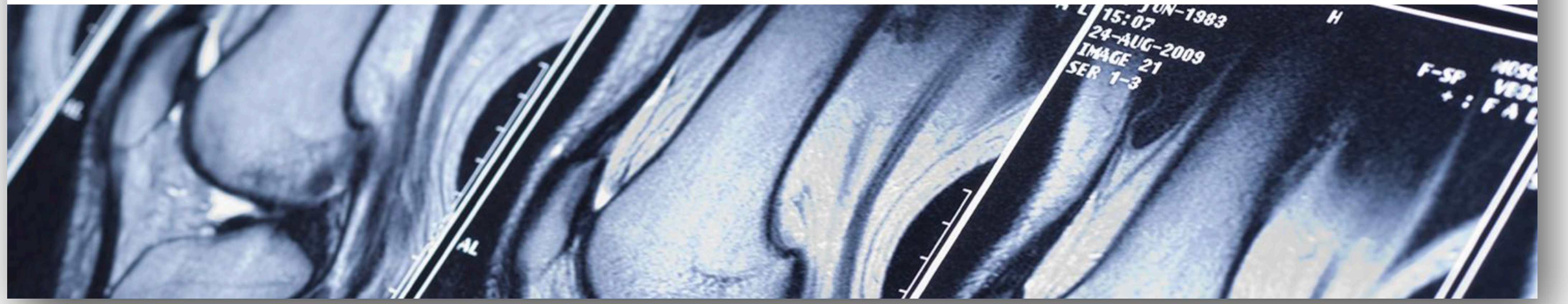
News & Views | Open access | Published: 25 November 2024

Artificial Intelligence awarded two Nobel Prizes for innovations that will shape the future of medicine

Ben Li & Stephen Gilbert

npj Digital Medicine 7, Article number: 336 (2024) | Cite this article

New Research Finds FastMRI Scans Generated with Artificial Intelligence Are as Accurate as Traditional MRI



Why do **neural networks** work so well?

ROBOTICS

Computer Eyesight Gets a Lot More Accurate

BY JOHN MARKOFF AUGUST 18, 2014 8:01 PM

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

TECH

Google just made artificial-intelligence history

By Drake Baer

First
confirme

New artificial intel

revolution

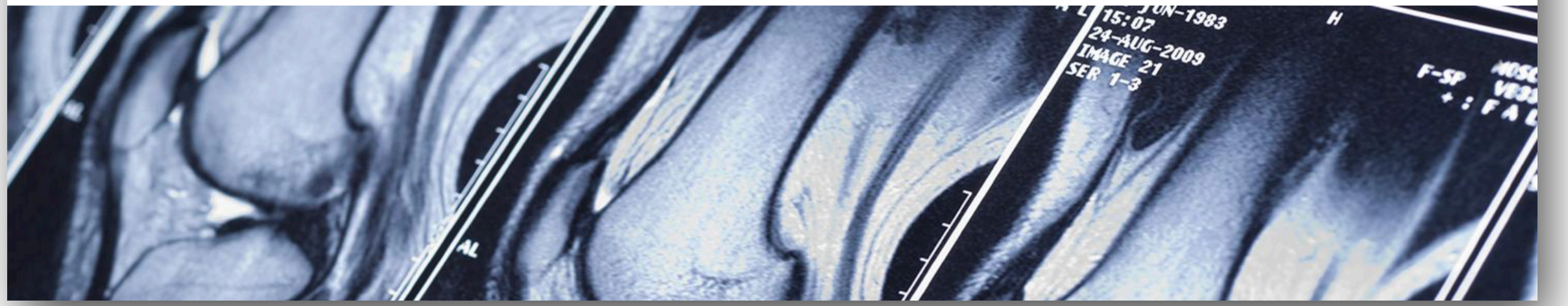
News & Views | Open access | Published: 25 November 2024

Artificial Intelligence awarded two Nobel Prizes for innovations that will shape the future of medicine

Ben Li & Stephen Gilbert

npj Digital Medicine 7, Article number: 336 (2024) | Cite this article

New Research Finds FastMRI Scans Generated with Artificial Intelligence Are as Accurate as Traditional MRI



ROBOTICS

Computer Eyesight Gets a Lot More Accurate

BY JOHN MARKOFF AUGUST 18, 2014 8:01 PM

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

TECH

Google just made artificial-intelligence history

By Drake Baer

Why do **neural networks** work so well for solving **inverse problems**?

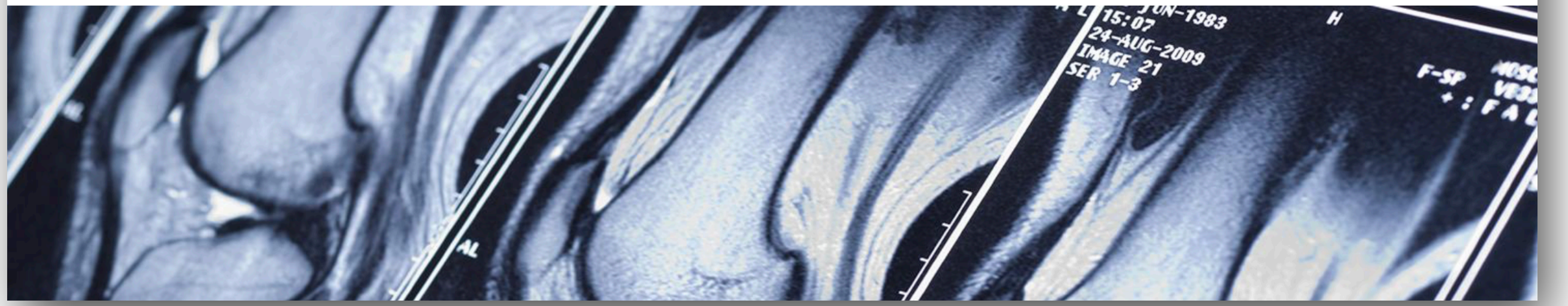
News & Views | Open access | Published: 25 November 2024

Artificial Intelligence awarded two Nobel Prizes for innovations that will shape the future of medicine

Ben Li & Stephen Gilbert

npj Digital Medicine 7, Article number: 336 (2024) | Cite this article

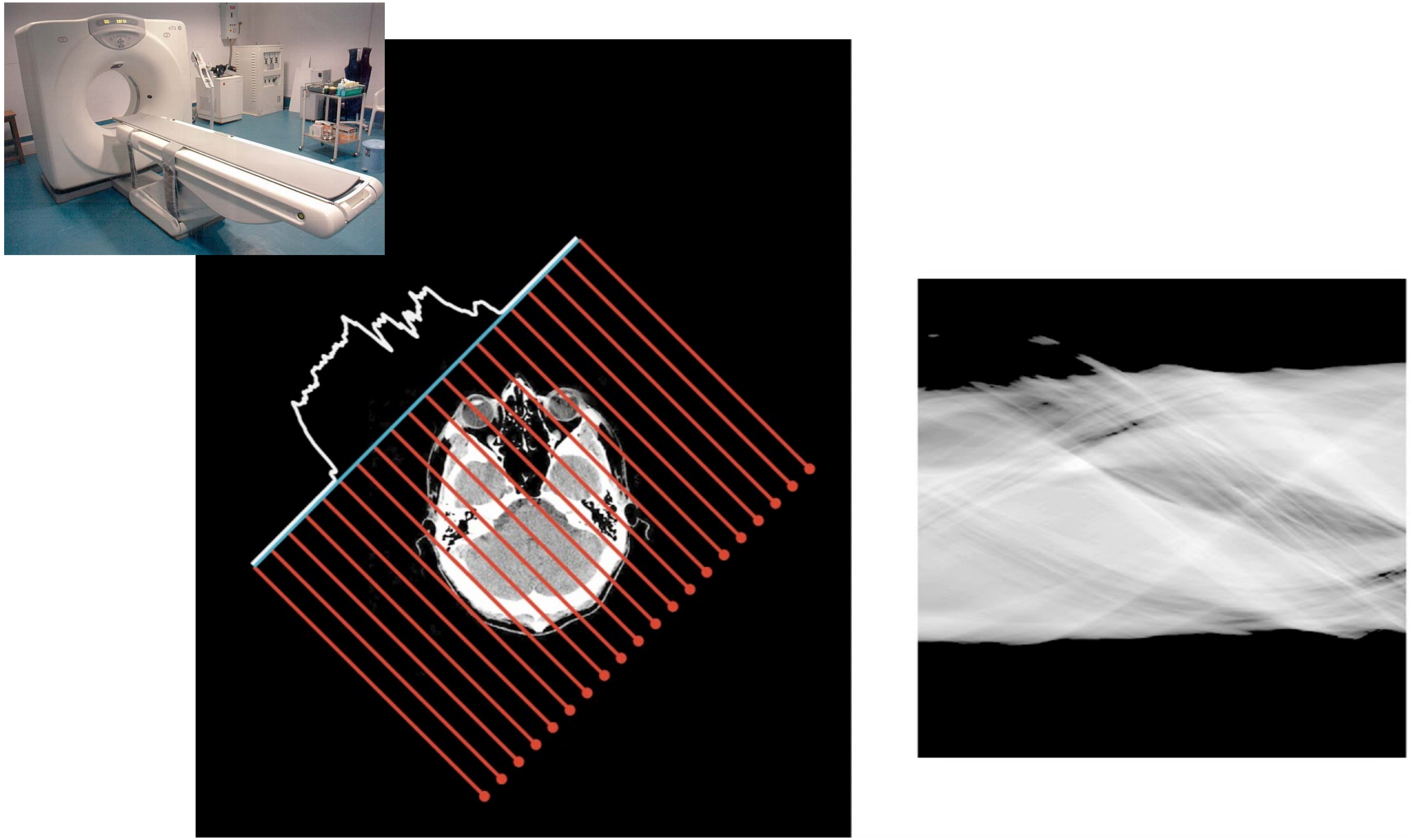
New Research Finds FastMRI Scans Generated with Artificial Intelligence Are as Accurate as Traditional MRI



What is an **inverse problem**?

Video from Samuli Siltanen https://www.youtube.com/watch?v=q7Rt_OY_7tU
CT Scan from Andrew Ciscel

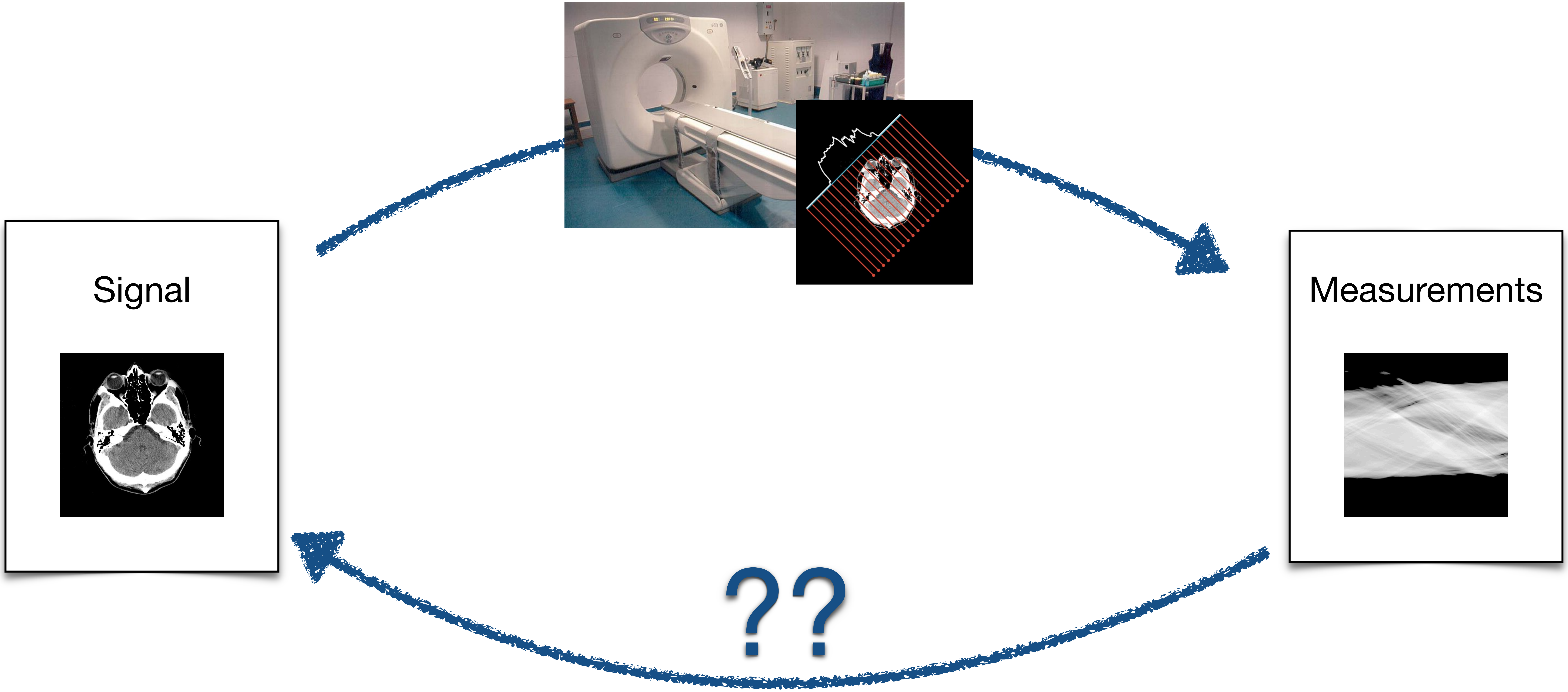
What is an **inverse problem**?



Computed Tomography (CT) Scan

Video from Samuli Siltanen https://www.youtube.com/watch?v=q7Rt_OY_7tU
CT Scan from Andrew Ciscel

What is an **inverse problem**?



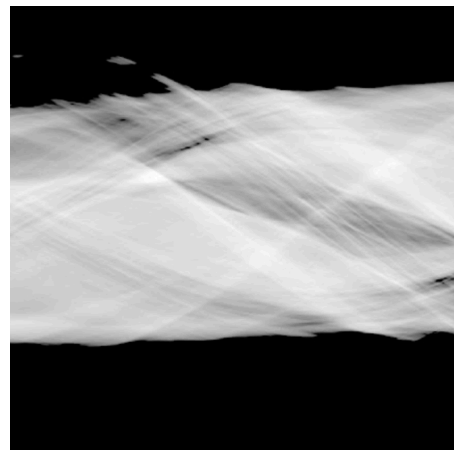
Video from Samuli Siltanen https://www.youtube.com/watch?v=q7Rt_OY_7tU
CT Scan from Andrew Ciscel

What is an **inverse problem**?

$$\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i) \text{ for } i = 1, \ldots, n$$

n examples of this process

Measurements
$\mathbf{y} \in \mathbb{R}^m$



Measurement Operator
$\mathbf{A} : \mathbb{R}^d \to \mathbb{R}^m$

Signal
$\mathbf{x} \in \mathbb{R}^d$

$$\mathbf{y}_{\text{new}} = \mathbf{A}(\mathbf{x}_{\text{new}}) + \varepsilon$$

Measurement Noise
$\varepsilon \in \mathbb{R}^m$

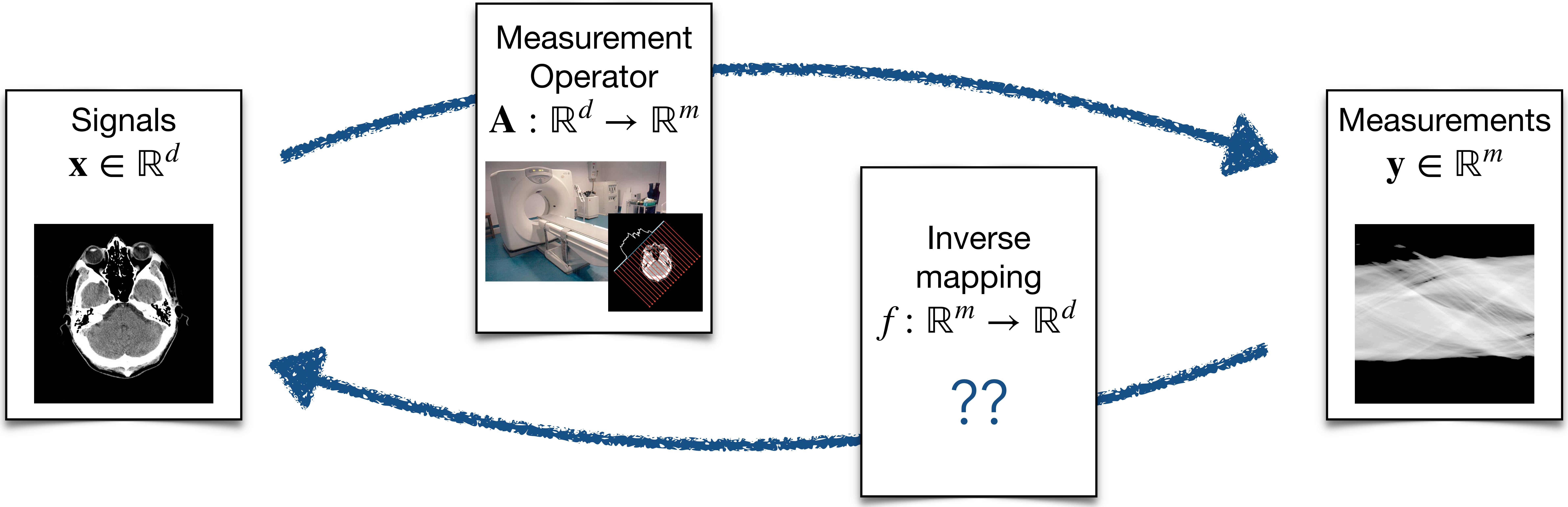
$$m \ll d$$

Given a new observation $\mathbf{y}_{\text{new}}$ our **goal** is to estimate $\mathbf{x}_{\text{new}}$

What is an **inverse problem**?

$$\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i) \text{ for } i = 1, \ldots, n$$

$$\mathbf{y}_{\text{new}} = \mathbf{A}(\mathbf{x}_{\text{new}}) + \varepsilon$$



Given a new observation $\mathbf{y}_{\text{new}}$ our **goal** is to estimate $\mathbf{x}_{\text{new}}$

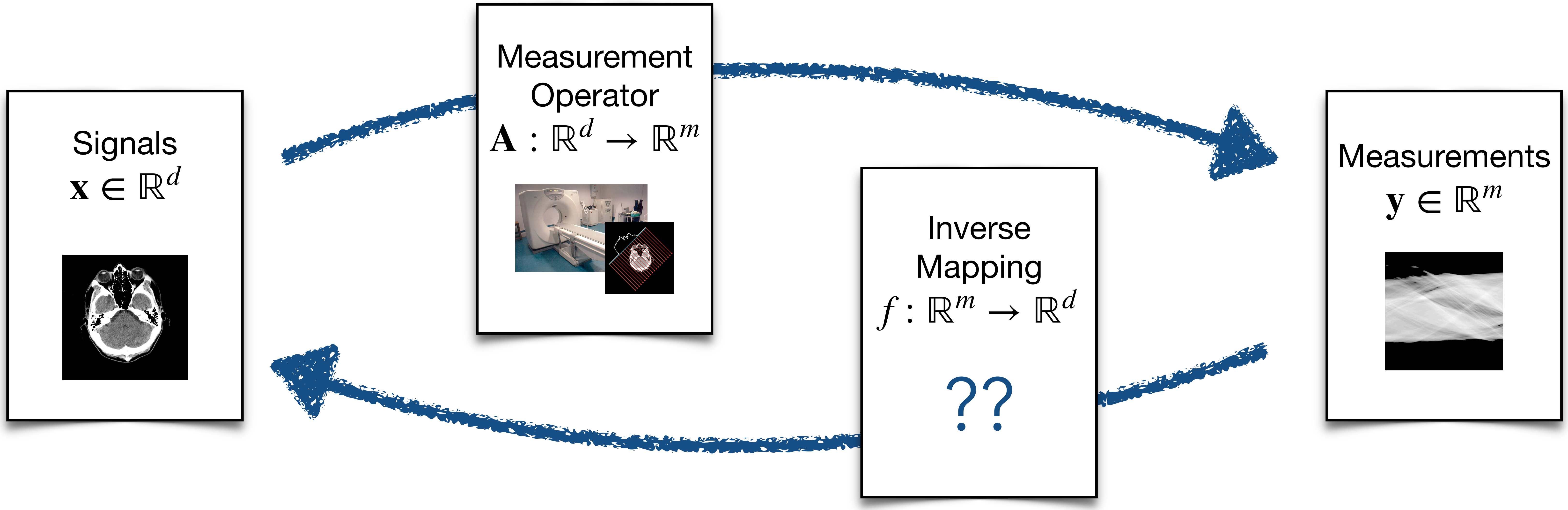
$\iff$

Find a function $f : \mathbf{y}_{\text{new}} \mapsto \mathbf{x}_{\text{new}}$

How did you **traditionally** solve inverse problems?

- Explicitly assume something about the structure of the signal
- Recover the signal as the solution to an optimization problem, regularized according to structural assumptions:

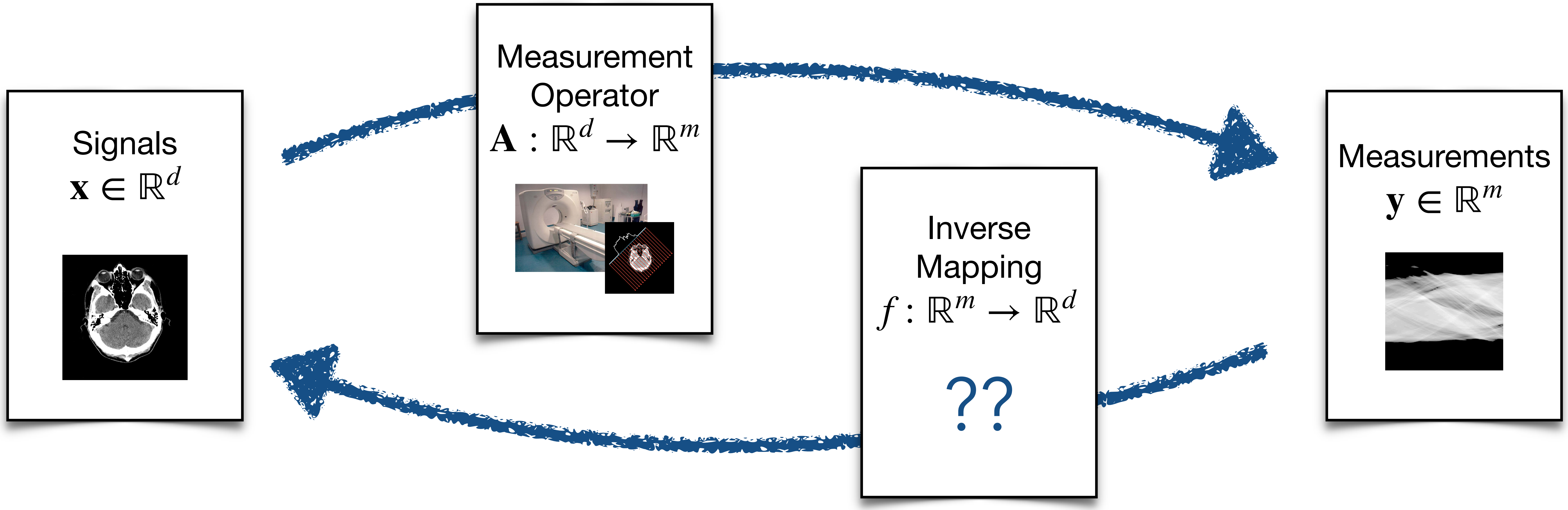
$$\mathbf{x}_{\text{new}} = f(\mathbf{y}_{\text{new}}) = \arg\min_{\mathbf{x}} \|\mathbf{y}_{\text{new}} - \mathbf{A}(\mathbf{x})\|_2^2 + \lambda \|\mathbf{x}\|_2^2$$



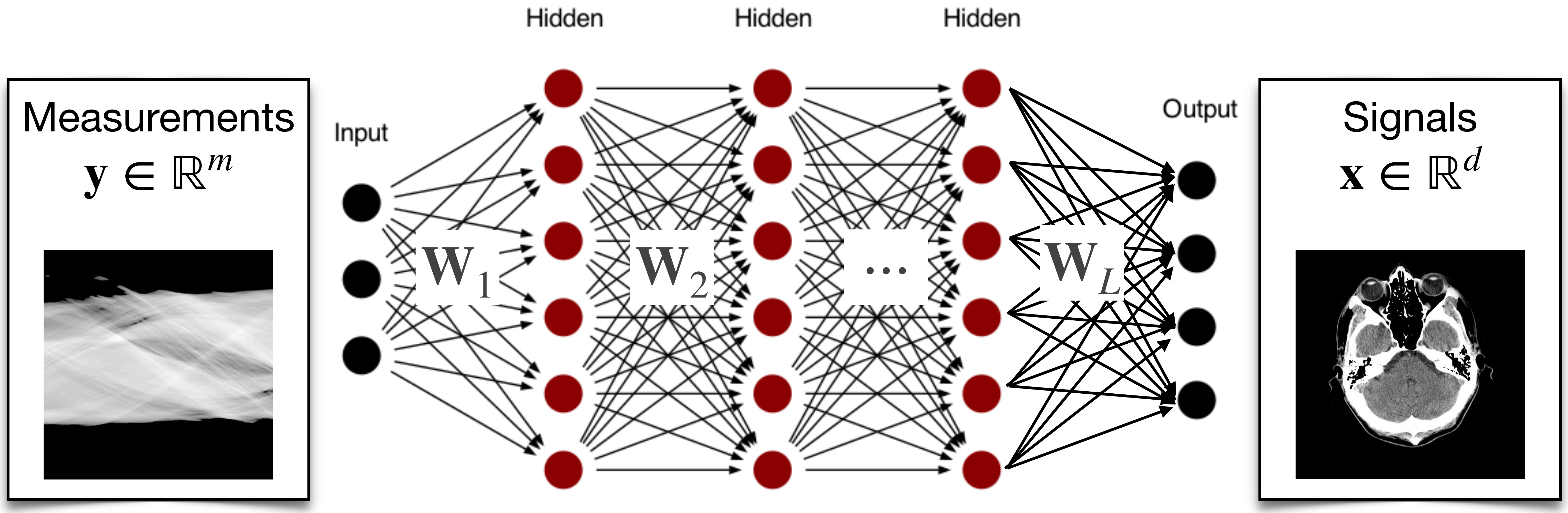
What are **machine learning** approaches to solving an inverse problem?

- If we have access to n training data pairs $\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i)$, can we learn an even better mapping $f : \mathbf{y} \mapsto \mathbf{x}$?
- Pick f in some model class $\mathcal{F}$ that best fits the training data, perhaps plus some regularization

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \; L(f) = \frac{1}{n} \sum_{i=1}^{n} \|f(\mathbf{y}_i) - \mathbf{x}_i\|^2 + R(f)$$



How do you solve inverse problems with a **neural network**?



$$\theta = \left(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L\right)$$

$$f_\theta(\mathbf{y}) = \mathbf{W}_L \sigma\left(\cdots \sigma\left(\mathbf{W}_2 \sigma\left(\mathbf{W}_1 \mathbf{y}\right)\right)\right)$$

Find $\hat{\theta} \in \underset{\theta}{\arg\min} \; L(\theta) = \frac{1}{2} \sum_{i=1}^{n} \|f_\theta(\mathbf{y}_i) - \mathbf{x}_i\|^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$

via **Gradient Descent:** $\theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

How do you solve inverse problems with a **neural network**?

- Machine learning approaches have been surprisingly successful for solving inverse problems
  Ongie et al. (2018), Barbastathis et al. (2019), Knoll et al (2020)

- The success is especially surprising in light of the very high dimensionality of the data

> Why do **neural networks** work so well for solving **inverse problems**?

- Hypothesis: The training signals have **latent low-dimensional structure** that is preserved by the measurement operator, and neural networks are **adapting to that structure**, allowing for improved robustness to noise at test time. **How does this happen?**

Simplified Setting

- Let us assume that the training signals have a simple form of **latent low-dimensional structure** that is preserved by the measurement operator

- Does a simple neural network **adapt** to that structure? Does this improve **robustness**?

Low-dimensional structure that is **preserved** by the measurement operator

$\mathbf{x}_i = \mathbf{R}\mathbf{z}_i \in \mathbb{R}^d$ and $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{R}\mathbf{z}_i \in \mathbb{R}^m$

$n \geq s$ samples of this process

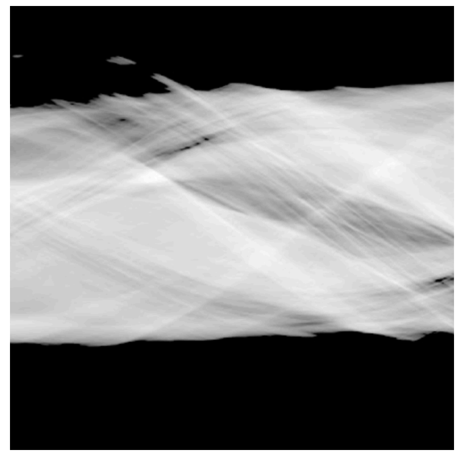
$\mathbf{R} \in \mathbb{R}^{d \times s}$

Signals live in an s-dimensional subspace $\mathbf{x} \in \text{range}(\mathbf{R}) \subseteq \mathbb{R}^d$

Linear measurement operator $\mathbf{A} \in \mathbb{R}^{m \times d}$ preserves the subspace structure

$(1-\delta)\|\mathbf{x}\| \leq \|\mathbf{A}\mathbf{x}\| \leq (1+\delta)\|\mathbf{x}\|$ $\forall \mathbf{x} \in \text{range}(\mathbf{R})$

Measurements live in an s-dimensional subspace $\mathbf{y} \in \text{range}(\mathbf{A}\mathbf{R}) \subseteq \mathbb{R}^m$



$$s \ll m \ll d$$

$$\mathbf{y}_{\text{new}} = \mathbf{A}\mathbf{x}_{\text{new}} + \varepsilon = \mathbf{A}\mathbf{R}\mathbf{z}_{\text{new}} + \varepsilon$$

Measurement Noise $\varepsilon \in \mathbb{R}^m$

Warning: What follows is **not** advice on how to solve this inverse problem!

- If you know *a priori* that your inverse problem has this subspace structure, then there is a known way recover $\mathbf{x}_{\text{new}}$ with high accuracy
- **Oracle solution** using the Moore-Penrose Pseudoinverse:

Stack n samples into matrices $\mathbf{X} \in \mathbb{R}^{d \times n}$ and $\mathbf{Y} \in \mathbb{R}^{m \times n}$

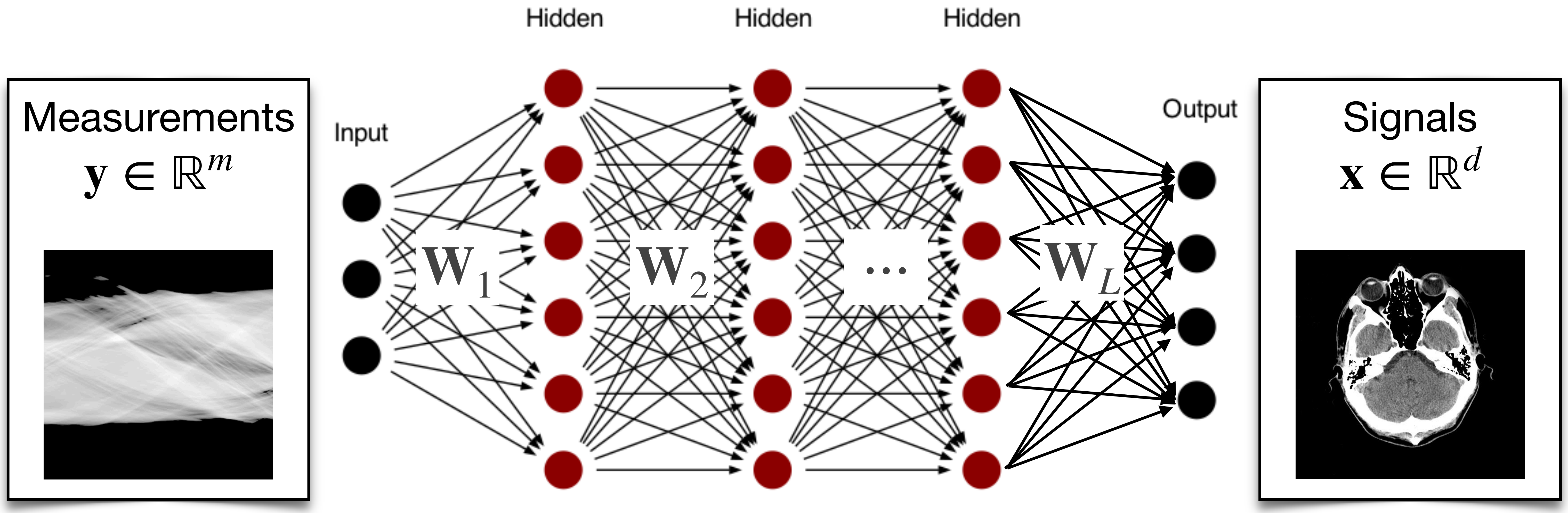
$$\mathbf{x}_{\text{new}} \approx \mathbf{R}(\mathbf{A}\mathbf{R})^{\dagger}\mathbf{y}_{\text{new}} = \mathbf{X}\mathbf{Y}^{\dagger}\mathbf{y}_{\text{new}}$$

- Precisely because the oracle solution exists, we can analyze how close the learned neural network is to doing the "right" thing
- An inverse mapping that takes advantage of the **low-dimensional structure** does much better than one that does not
- What does this simplified setting reveal about the ability of neural networks to **automatically adapt to structure** in data?

$$\mathbf{x}_i = \mathbf{R}\mathbf{z}_i \in \mathbb{R}^d \text{ and } \mathbf{y}_i = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{R}\mathbf{z}_i \in \mathbb{R}^m$$

$$\mathbf{y}_{\text{new}} = \mathbf{A}\mathbf{x}_{\text{new}} + \varepsilon = \mathbf{A}\mathbf{R}\mathbf{z}_{\text{new}} + \varepsilon$$

Neural Network with Linear Activations



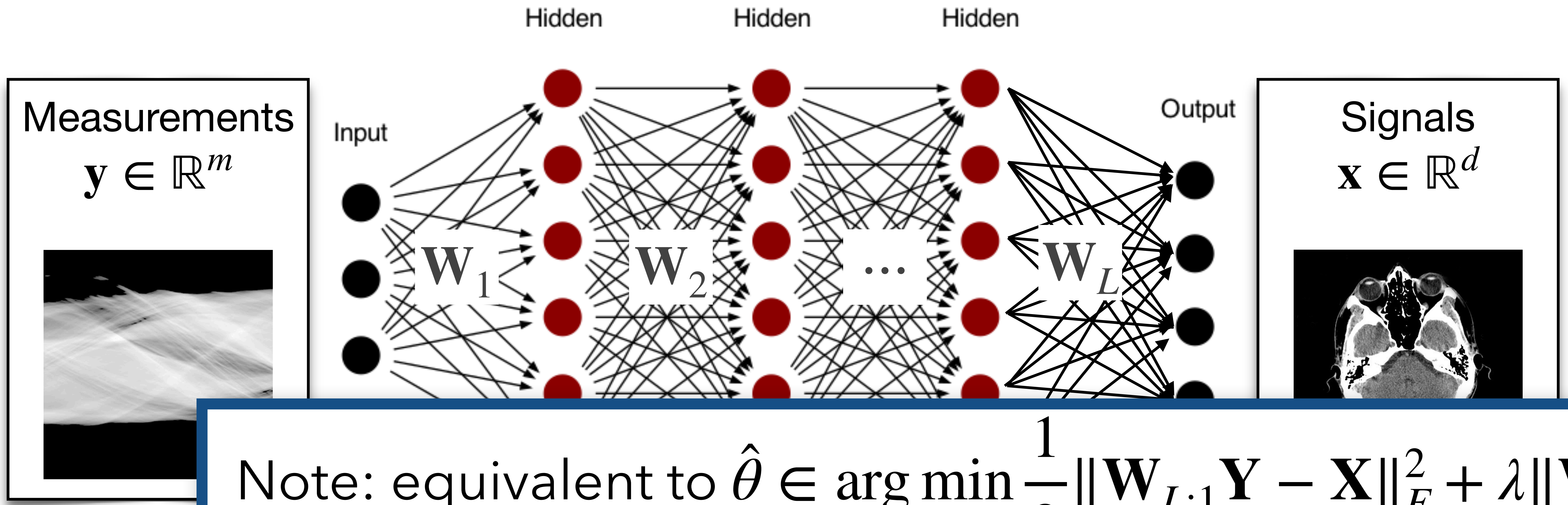
$$\theta = \left(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L\right)$$

$$f_\theta(\mathbf{y}) = \mathbf{W}_L \cdots \mathbf{W}_2 \mathbf{W}_1 \mathbf{y}$$

Find $\hat{\theta} \in \arg\min_{\theta} \; L(\theta) = \frac{1}{2}\|\mathbf{W}_L \cdots \mathbf{W}_1 \mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$

via **Gradient Descent:** $\theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

Neural Network with Linear Activations



Never explicitly imposing low-dimensional structure!

$$\theta = \left(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L\right)$$

$$f_\theta(\mathbf{y}) = \mathbf{W}_L \cdots \mathbf{W}_2 \mathbf{W}_1 \mathbf{y}$$

Note: equivalent to $\hat{\theta} \in \arg\min_{\theta} \frac{1}{2} \|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \|\mathbf{W}_{L:1}\|_{S^{2/L}}^{2/L}$

But gradient descent trajectory may be different!

Find $\hat{\theta} \in \arg\min_{\theta} \; L(\theta) = \frac{1}{2} \|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$

via **Gradient Descent:** $\theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

Previous work on gradient descent in linear neural networks

- Without regularization ($\lambda = 0$) *Du & Hu (2019), Xu et al. (2023)*
- Unrealistic initialization assumptions *Hu et al. (2020), Hu et al. (2022), Arora et al. (2019), Nguegnang et al. (2024), Arora et al. (2018)*
- Step-size η very small *Lewkowycz & Gur-Ari (2020), Gidel et al. (2019), Ji & Telgarsky (2019), Eftekhari (2020), Bah et al. (2019), Pesme et al. (2021), Jacot et al. (2021), Arora et al. (2018),*
- SGD can only ever decrease the rank of a solution, but unclear if it finds a good fit to the data *Wang & Jacot (2024)*

Our analysis

➡With regulariation ($\lambda > 0$)

➡Initialization essentially equivalent to using **PyTorch default**

➡Very **mild** assumptions on stepsize η

➡**Both** adaptation to structure and good fit to the training data

So **what happens** in our simplified setting?

$\mathbf{x}_i = \mathbf{R}\mathbf{z}_i \in \mathbb{R}^d$ and $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{R}\mathbf{z}_i \in \mathbb{R}^m$

$\mathbf{y}_{\text{new}} = \mathbf{A}\mathbf{x}_{\text{new}} + \varepsilon = \mathbf{A}\mathbf{R}\mathbf{z}_{\text{new}} + \varepsilon$

Find $\hat{\theta} \in \arg\min_{\theta} \; L(\theta) = \frac{1}{2}\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$

via **Gradient Descent:** $\theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

We track the evolution of **two main quantities** throughout gradient descent:

$\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F$ and $\|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2$

How well network reconstructs the training data

How well network adapts to low-dimensional structure

$\mathbf{P}_\perp$ = projection onto range$(\mathbf{A}\mathbf{R})^\perp$

Good reconstructions of training data & adaptation to structure
$\Longrightarrow$ robustness to noise at test-time

$$\|\mathbf{W}_{L:1}\mathbf{y}_{\text{new}} - \mathbf{x}_{\text{new}}\|_2 \leq \|\mathbf{W}_{L:1} - \mathbf{X}\mathbf{Y}^\dagger\|_2 \|\mathbf{y}_{\text{new}}\|_2 + \|\mathbf{X}\mathbf{Y}^\dagger\mathbf{y}_{\text{new}} - \mathbf{x}_{\text{new}}\|_2$$

$$\|\mathbf{W}_{L:1} - \mathbf{X}\mathbf{Y}^\dagger\|_2 \leq \|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F \|\mathbf{Y}^\dagger\|_2 + \|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2$$

So **what happens** in our simplified setting?

$\mathbf{x}_i = \mathbf{R}\mathbf{z}_i \in \mathbb{R}^d$ and $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{R}\mathbf{z}_i \in \mathbb{R}^m$

$\mathbf{y}_{\text{new}} = \mathbf{A}\mathbf{x}_{\text{new}} + \varepsilon = \mathbf{A}\mathbf{R}\mathbf{z}_{\text{new}} + \varepsilon$

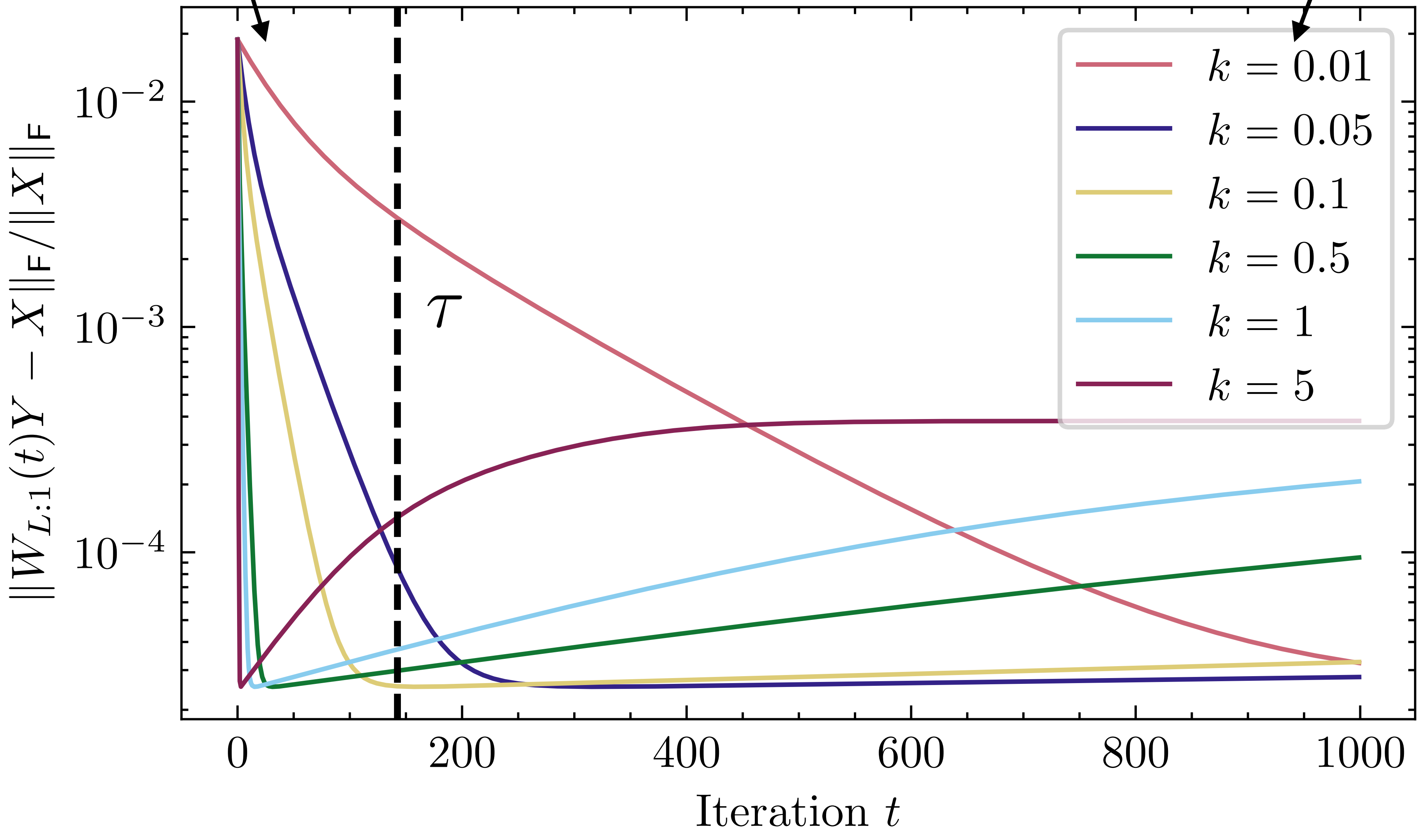
Find $\hat{\theta} \in \arg\min_{\theta} \; L(\theta) = \frac{1}{2}\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$

via **Gradient Descent:** $\quad \theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

How well network reconstructs the training data

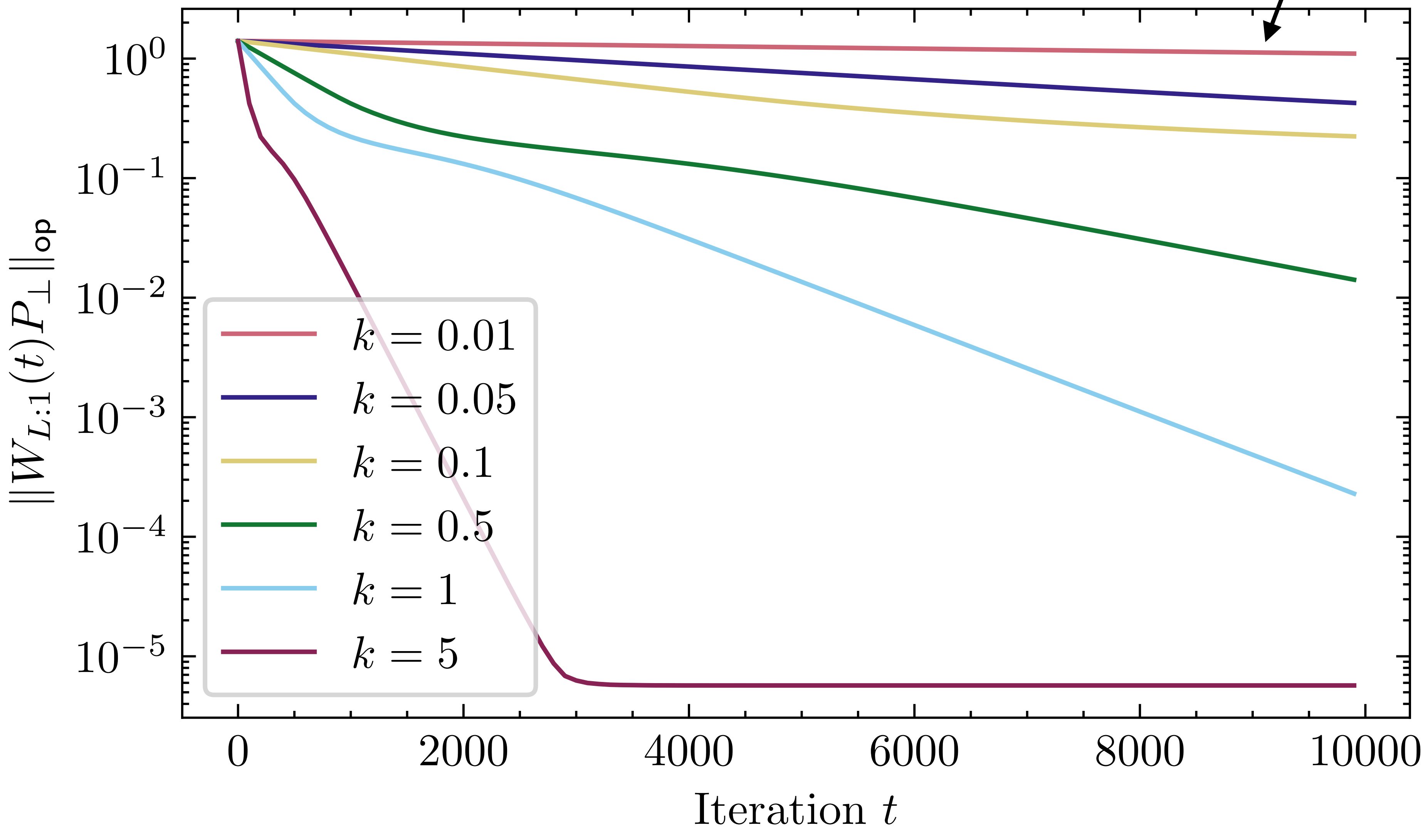
$\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F$

Training with different stepsizes $\eta = k \cdot \hat{\eta}$ where $\hat{\eta}$ is the stepsize prescribed by our theory



$\|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2$

How well network adapts to low-dimensional structure



Two phases:

1. **Rapid** improvement in **reconstructions** of the training samples in first τ iterations
2. **Slow** recovery of the latent low-dimensional **structure**

So **what happens** in our simplified setting?

$$\mathbf{x}_i = \mathbf{R}\mathbf{z}_i \in \mathbb{R}^d \text{ and } \mathbf{y}_i = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{R}\mathbf{z}_i \in \mathbb{R}^m$$

$$\mathbf{y}_{\text{new}} = \mathbf{A}\mathbf{x}_{\text{new}} + \varepsilon = \mathbf{A}\mathbf{R}\mathbf{z}_{\text{new}} + \varepsilon$$

$$\text{Find } \hat{\theta} \in \arg\min_{\theta} \; L(\theta) = \frac{1}{2}\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$$

via **Gradient Descent:** $\theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

Two phases:

1. **Rapid** improvement in reconstructions of the **training** samples

$$\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F = O\left(\frac{\lambda}{L}\right)$$

after $\tau = O\left(\frac{1}{\eta L}\log\left(\frac{L}{\lambda}\right)\right)$ iterations

2. **Slow** recovery of the latent low-dimensional **structure**

$$\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F = O(\lambda) \text{ and } \|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2 = O\left(\frac{1}{d_w^C}\right)$$

after $T = O\left(\frac{\log(d_w)}{\eta\lambda}\right)$ iterations

Good reconstructions of training data & adaptation to structure

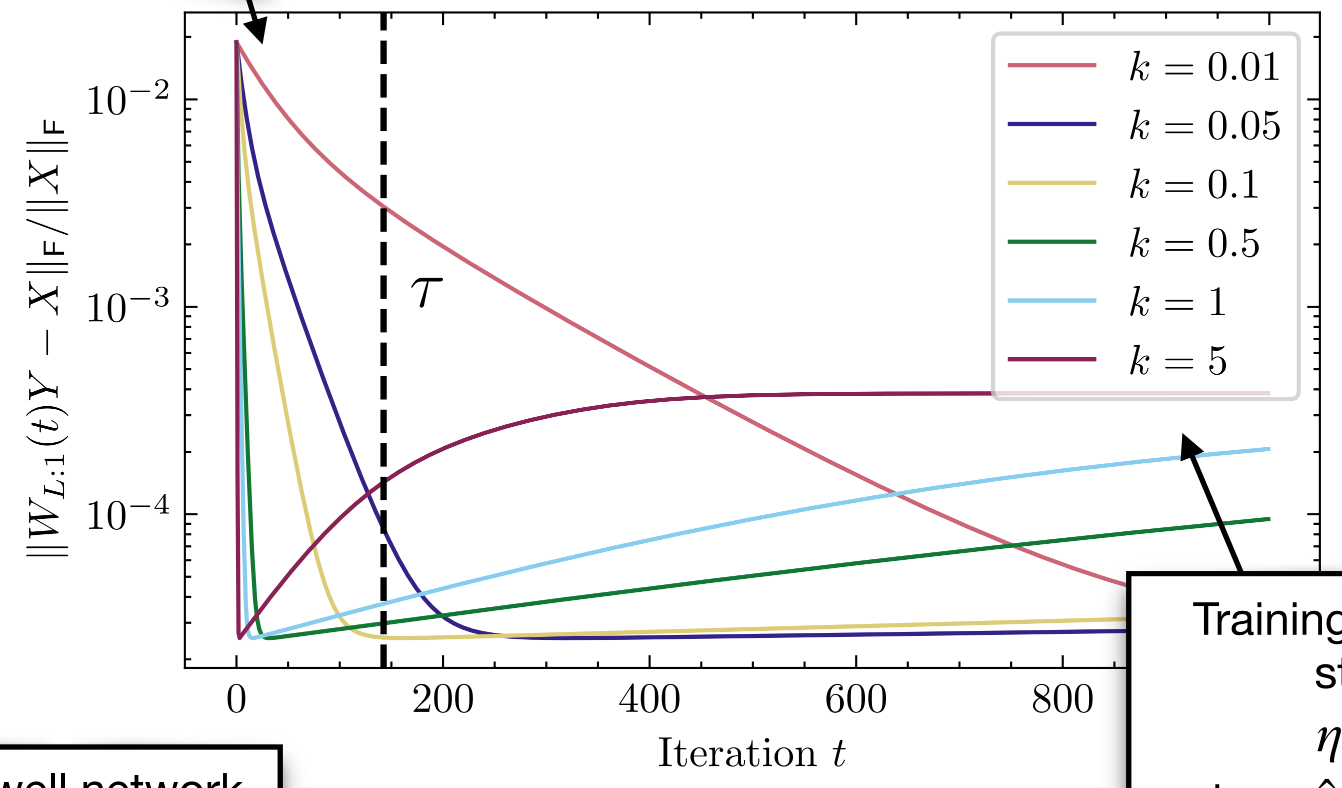
$\Longrightarrow$ robustness to noise at test-time

$$\|\mathbf{W}_{L:1} - \mathbf{X}\mathbf{Y}^\dagger\|_2 \le \|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F \|\mathbf{Y}^\dagger\|_2 + \|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2$$

Distance to **oracle** solution is small at the end of Phase 2

How well network reconstructs the training data

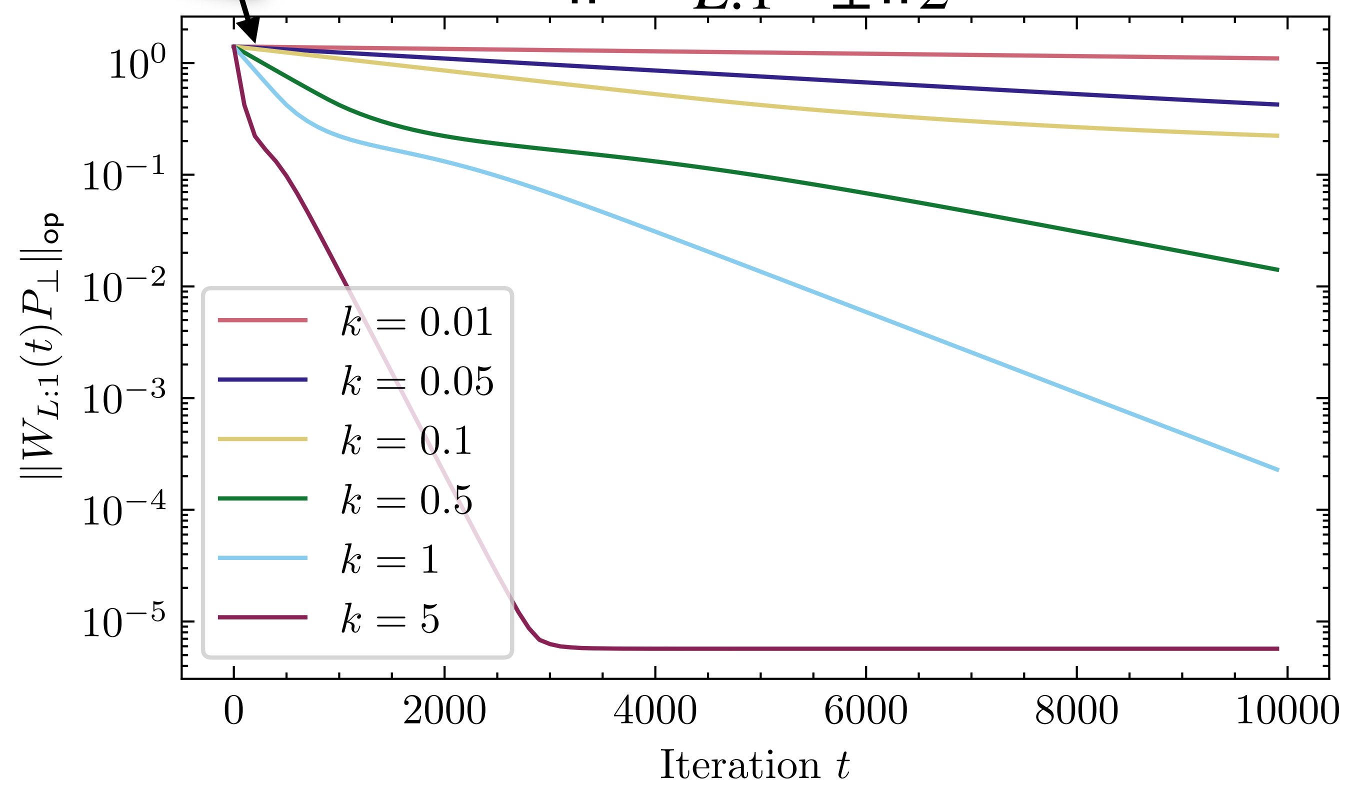
$\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F$



Training with different stepsizes $\eta = k \cdot \hat{\eta}$ where $\hat{\eta}$ is the stepsize prescribed by our theory

How well network adapts to low-dimensional structure

$\|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2$



So **what happens** in our simplified setting?

$\mathbf{x}_i = \mathbf{R}\mathbf{z}_i \in \mathbb{R}^d$ and $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{R}\mathbf{z}_i \in \mathbb{R}^m$

$\mathbf{y}_{\text{new}} = \mathbf{A}\mathbf{x}_{\text{new}} + \varepsilon = \mathbf{A}\mathbf{R}\mathbf{z}_{\text{new}} + \varepsilon$

Find $\hat{\theta} \in \arg\min_{\theta} \; L(\theta) = \frac{1}{2}\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda \sum_{\ell=1}^{L} \|\mathbf{W}_\ell\|_F^2$

via **Gradient Descent:** $\theta^{t+1} = \theta^t - \eta \nabla L(\theta^t)$

Two phases:

1. **Rapid** improvement in reconstructions of the **training** samples

$$\|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F = O\left(\frac{\lambda}{L}\right)$$

afte

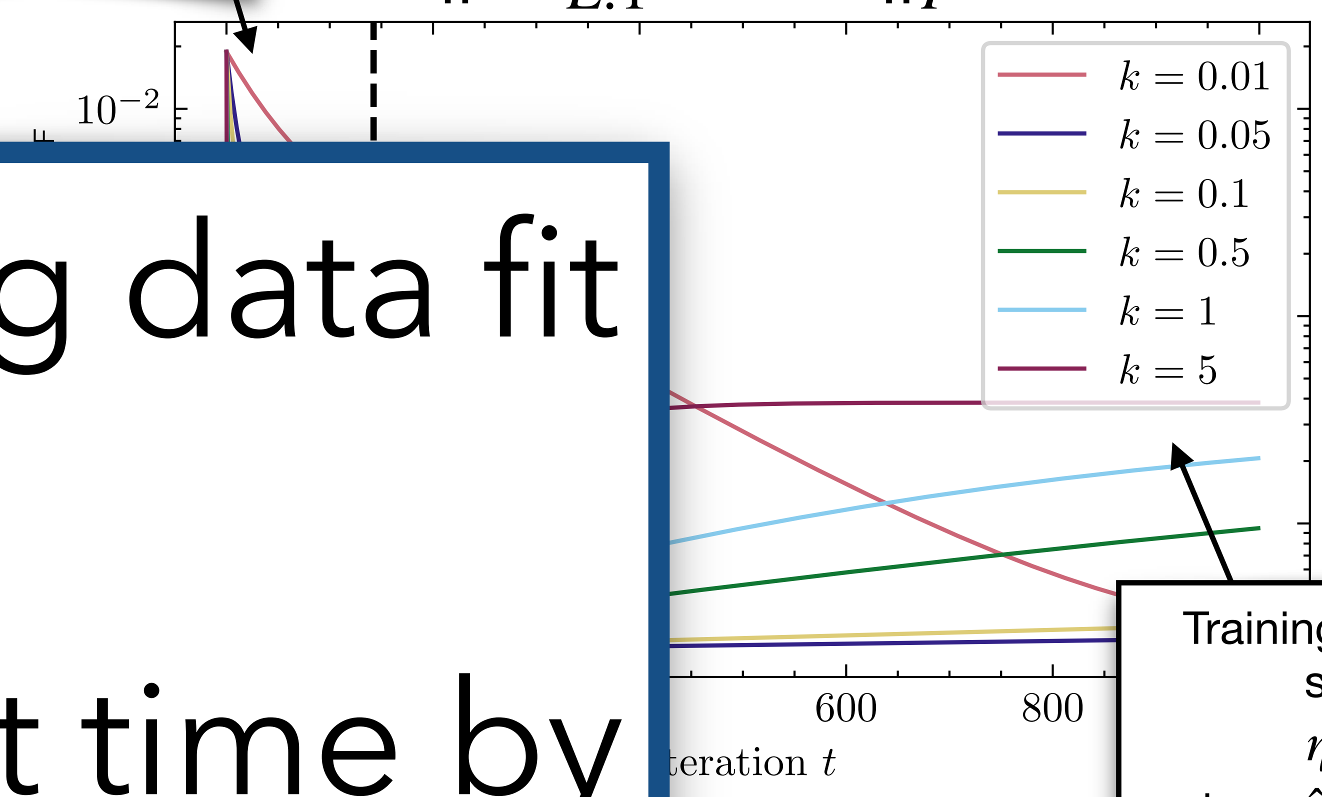
2. **Slow** recovery of the la

$\|\mathbf{W}_{L:1}\mathbf{Y} -$

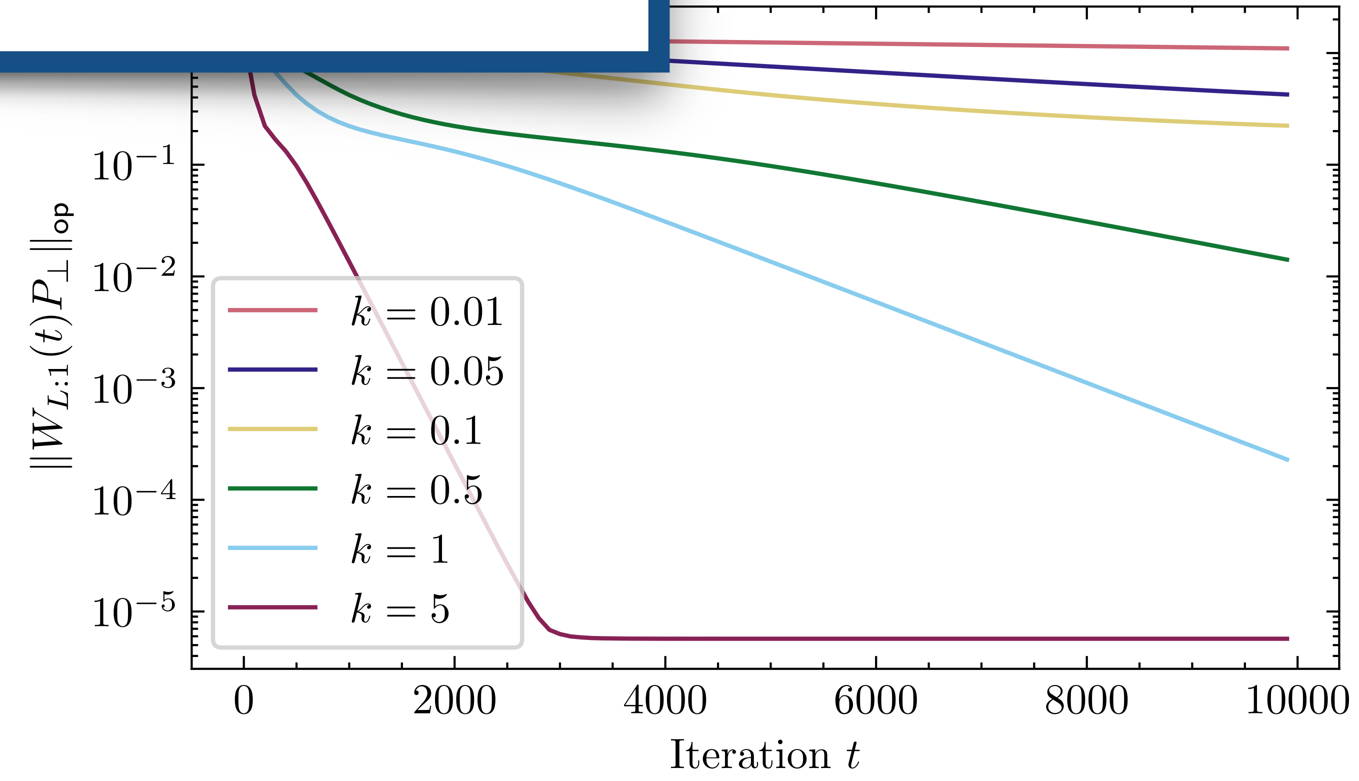
a

Continuing to train after training data fit stops improving →
Network that does better at test time by adapting to low-dimensional structure

How well network reconstructs the training data



Training with different stepsizes $\eta = k \cdot \hat{\eta}$ where $\hat{\eta}$ is the stepsize prescribed by our theory



Good reconstructions of training data & adaptation to structure
$\Longrightarrow$ robustness to noise at test-time

$$\|\mathbf{W}_{L:1} - \mathbf{X}\mathbf{Y}^\dagger\|_2 \leq \|\mathbf{W}_{L:1}\mathbf{Y} - \mathbf{X}\|_F \|\mathbf{Y}^\dagger\|_2 + \|\mathbf{W}_{L:1}\mathbf{P}_\perp\|_2$$

Distance to **oracle** solution is small at the end of Phase 2

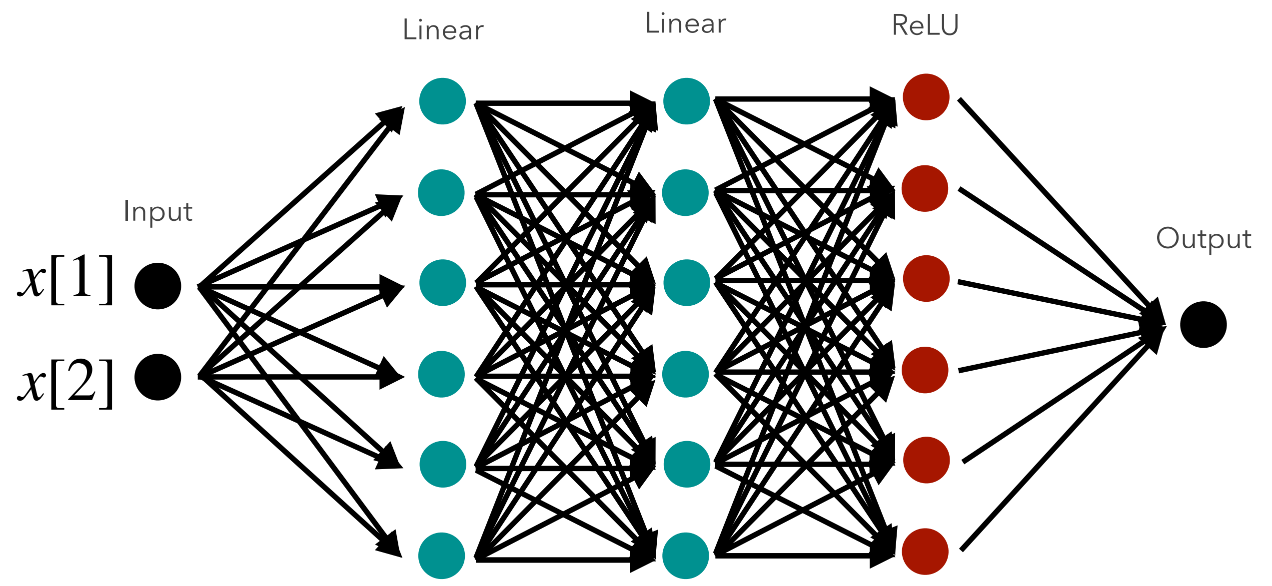
One reason **neural networks** work well for solving **inverse problems** is because they can **automatically adapt** to structure in data

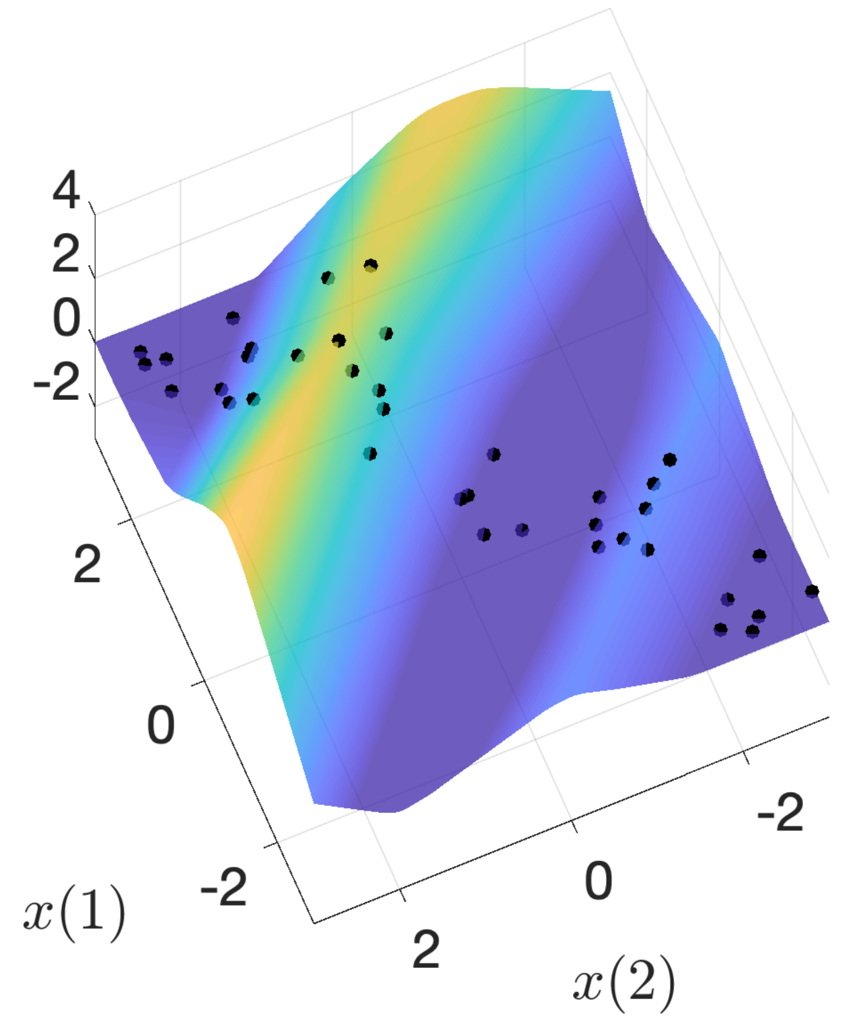
What’s next?

- What about more complex forms of low-dimensional structure in data?
- What about more complex neural networks?
- How does depth affect the ability to adapt to low-dimensional structure?
- What about stochastic variants of gradient descent? Adam, etc.?

What else?

- Studying how nonlinear neural network architectures adapt to low-dimensional structure
  - Adding linear layers to a ReLU network yields a trained network that mostly **only varies in a few directions** in the input space *Parkinson, Ongie & Willett (2025)*
  - Functions that can be represented by a deep ReLU network with small **norm** will have low-dimensional structure *Jacot (2023)*
  - Similar behavior can be induced with only a few ReLU layers and many linear layers
- What can **deeper** networks do that **shallower** networks can't? *Parkinson, Ongie, Willett, Shamir & Srebro (2024)*





Thank you!

https://arxiv.org/abs/2502.15522

Hannah Laus
Technical University
of Munich

Vasileios Charisopoulos
University
of Washington

Rebecca Willett
University
of Chicago

Felix Krahmer
Technical University
of Munich

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2140001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.