# Lecture 9: Deep Learning for Inverse Problems

Haizhao Yang
Department of Mathematics
University of Maryland College Park

2022 Summer Mini Course
Tianyuan Mathematical Center in Central China

# Outline

- End-to-end methods using neural networks
- Learning to regularize: first learn, then reconstruct
- Unrolled and learned optimization
- Learning to sense

# End-To-End DNN Methods

# Learning Mathematical Operators

## Notations

- Function spaces $\mathcal{X}$ and $\mathcal{Y}$, e.g., $\mathbb{R}$-valued over domain $\Omega \subset \mathbb{R}^D$
- Operator $\Psi : \mathcal{X} \to \mathcal{Y}$
- Data samples $\mathcal{S} = \{u_i, v_i\}_{i=1}^{2n}$ with

$$v_i = \Psi(u_i) + \epsilon_i,$$

where $u_i \overset{\text{i.i.d.}}{\sim} \gamma$ and $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mu$

## Goal

- Learn $\Psi$ from samples $\mathcal{S}$

## Method

- Deep neural networks $\Psi^n(u; \theta)$ as parametrization
- Supervised learning to find $\Psi^n(\cdot; \theta^*) \approx \Psi(\cdot)$

# Learning to Regularize: First Learn, then Reconstruct

# First Learn then Optimize

- Inverse problem:

$$\hat{\beta} = \arg\min \|y - X\beta\|_2^2 + R(\beta)$$

- $R(\beta)$: regularization
  - Classical regularization: Tikhonov regularization (ridge regression)

$$R(\beta) = \|\beta\|_2^2$$

  - Geometric regularization: e.g., sparsity, patch redundancy, total variation
  - Learned regularization: e.g.,

$$R(\beta) = 0, \text{ if } \beta \text{ is on the image manifold}$$

$$R(\beta) = \infty, \text{ otherwise}$$

Naive idea: Given an image set $B$, find a positive DNN $r(\cdot\,; \theta^*)$ such that

$$\theta^* = \arg\min \sum_{\beta \in B} |r(\beta; \theta)|^2$$

Advanced idea: GAN

# Unrolled and learned optimization

# Deep proximal gradient

$$y \longrightarrow \boxed{\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|_2^2 + r(\beta)} \longrightarrow \hat{\beta}$$

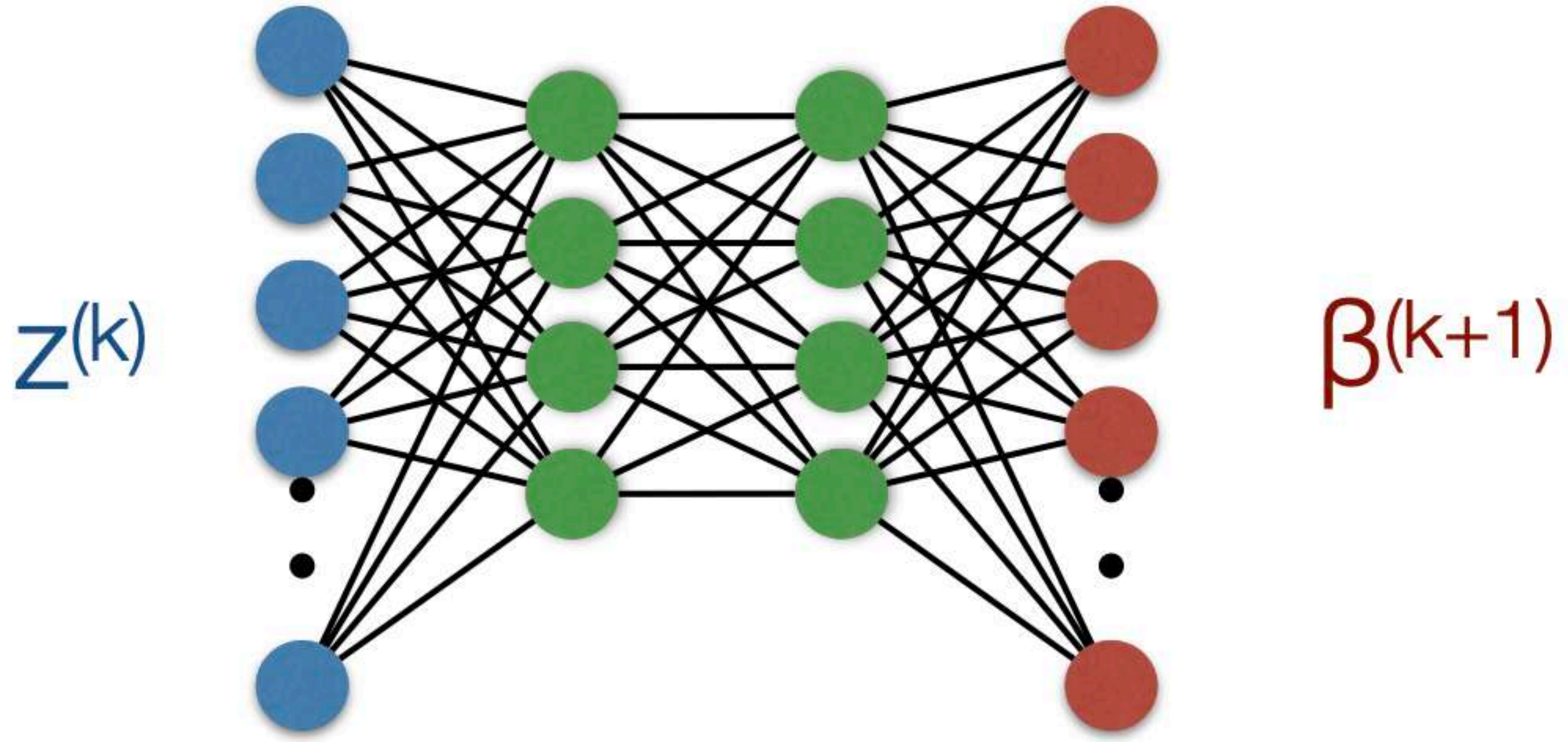
set $\hat{\beta}^{(1)}$ and stepsize $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = \hat{\beta}^{(k)} + \eta X^\top (y - X\hat{\beta}^{(k)})$$ gradient descent

$$\hat{\beta}^{(k+1)} = \boxed{\arg\min_{\beta} \|z^{(k)} - \beta\|_2^2 + \eta r(\beta)}$$ denoising

Replace with learned neural network




Slide from Rebecca Willett

*Chang, Li, Poczos, Kumar, & Sankaranarayanan 2017*

# “Unrolled” gradient descent

Assume $r(\beta)$ differentiable.

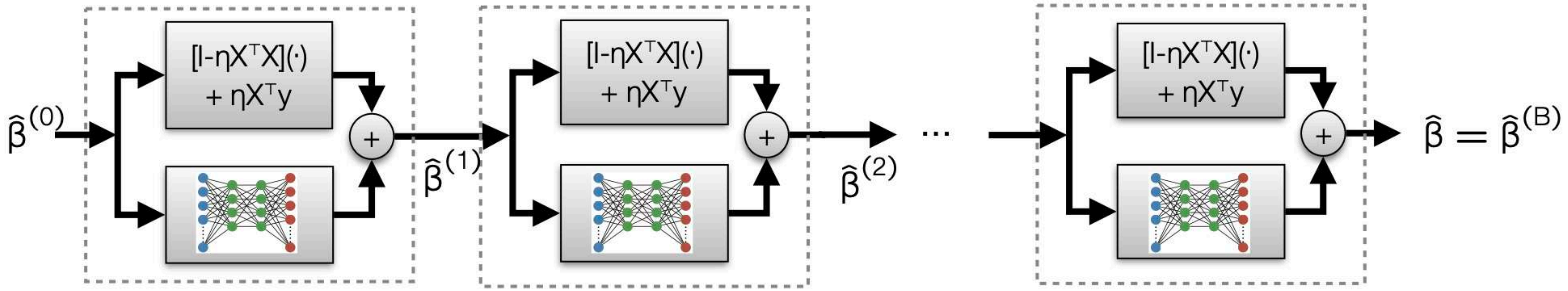
$$\widehat{\beta} = \arg\min_{\beta} \|y - X\beta\|_2^2 + r(\beta)$$

set $\widehat{\beta}^{(1)}$ and stepsize $\eta > 0$

for $k = 1, 2, \ldots$

$$\widehat{\beta}^{(k+1)} = \widehat{\beta}^{(k)} + \eta X^{\top}(y - X\widehat{\beta}^{(k)}) + \boxed{\eta \nabla r(\widehat{\beta}^{(k)})}$$

Replace with learned neural network




“Unrolled” optimization framework **trained end-to-end**

Slide from Rebecca Willett

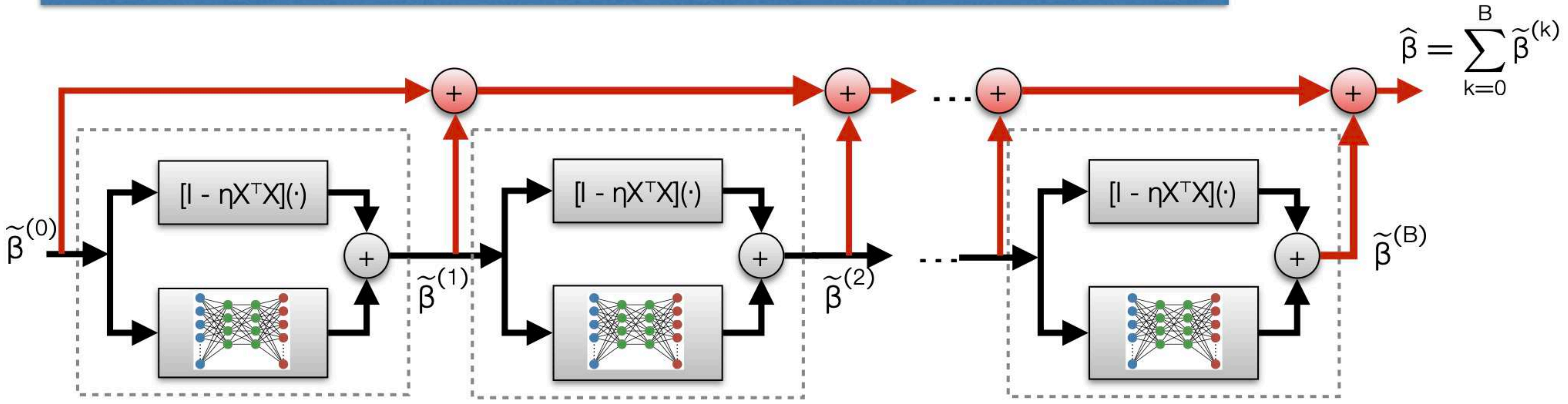
# Neumann networks

Assume $r(\beta)$ differentiable.

$$
\begin{aligned}
\hat{\beta} &= \arg\min_{\beta} \|y - X\beta\|_2^2 + r(\beta) \\
&= (X^\top X + \nabla r)^{-1} X^\top y \\
&\approx \sum_{k=1}^{B} (I - \eta X^\top X - \boxed{\eta \nabla r})^k \eta X^\top y
\end{aligned}
$$

Replace with learned neural network

**Neumann network (parallel pipelines + skip connections):**



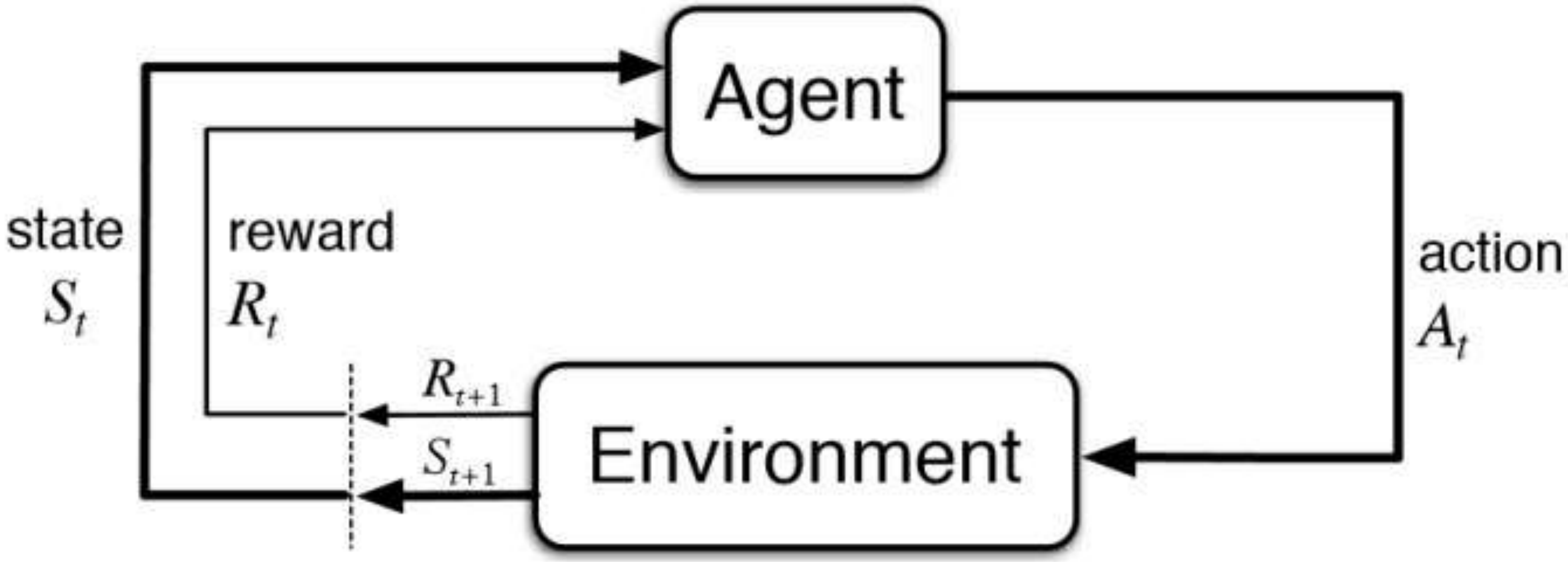

Slide from Rebecca Willett

# Learning to Sense

# Learning to Scan

Ziju Shen, Yufei Wang, Dufan Wu, Xu Yang, Bin Dong, IPI, 2022

# Learning to Discretize




By Richard S. Sutton and Andrew G. Barto.

- **Goal:** Learn how to select angles and assign does in CT scanning
- **Ideas:**

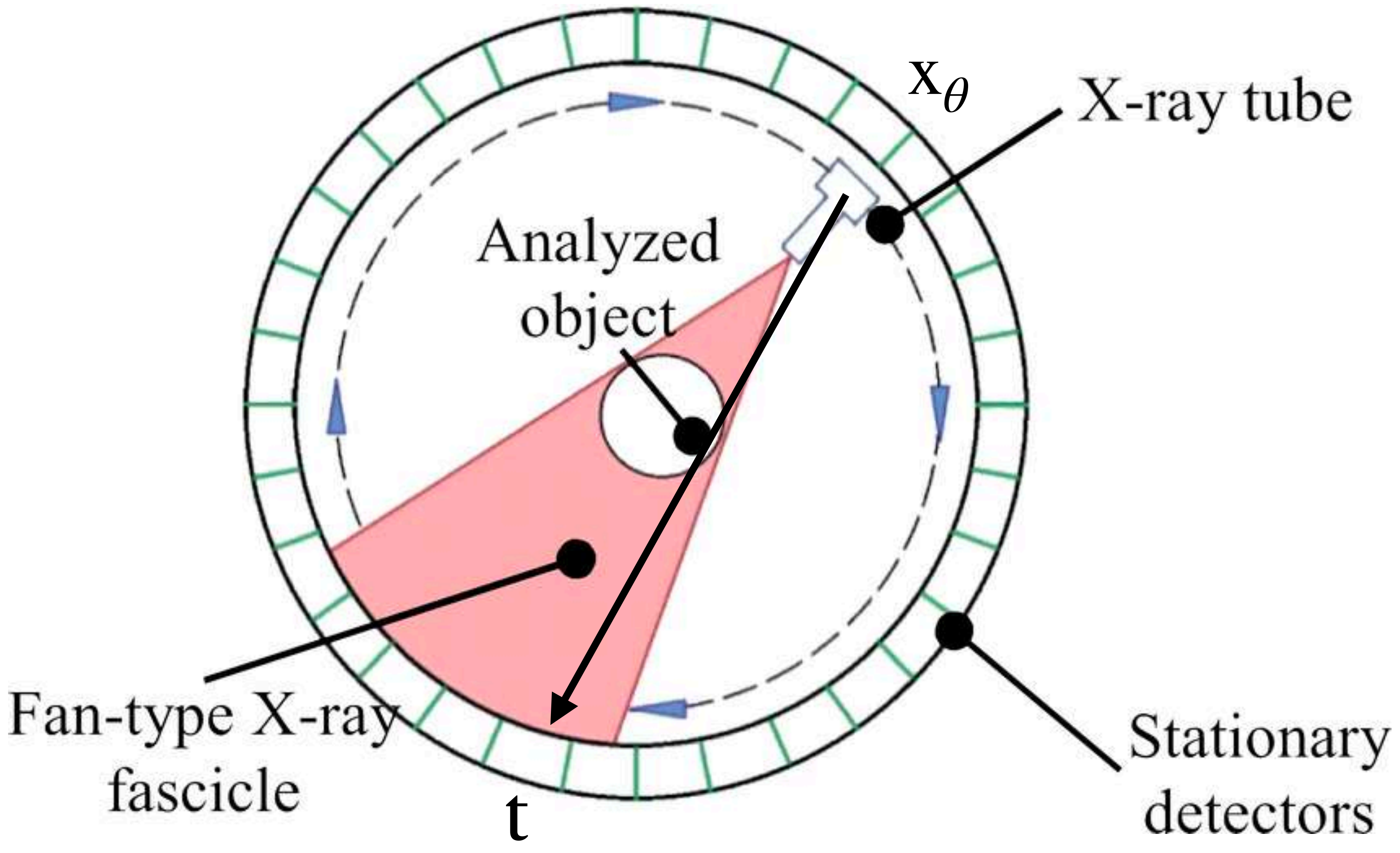
1. Reformulate the sequential (selection, assignment, reconstruction) procedure as a sequence of (action, state, reward)
2. Reformulate the decision strategy for selection and assignment as the policy to take actions
3. "Efficiently improving" the reconstruction quality as the reward
4. Learning to balance short-term noise reduction and long-term reconstruction quality

# Learning to Scan

## CT Imaging Background

Problem statement in 2D:

- Projection operator: $A^{\theta,r}[f](t) = \int_0^{L(t)} f(\mathrm{x}_\theta + \mathrm{n}s)ds$
- $f$ : unknown image
- $\theta$ : angle to determine the source at $\mathrm{x}_\theta = (x_\theta, y_\theta)$
- $r$ : X-ray beamlet with a direction $\mathrm{n} = (n_x, n_y)$
- $t$ : coordinate of the imager
- $L(t)$ : the distance from source at $\mathrm{x}_\theta$ to the imager location $t$



# Learning to Scan

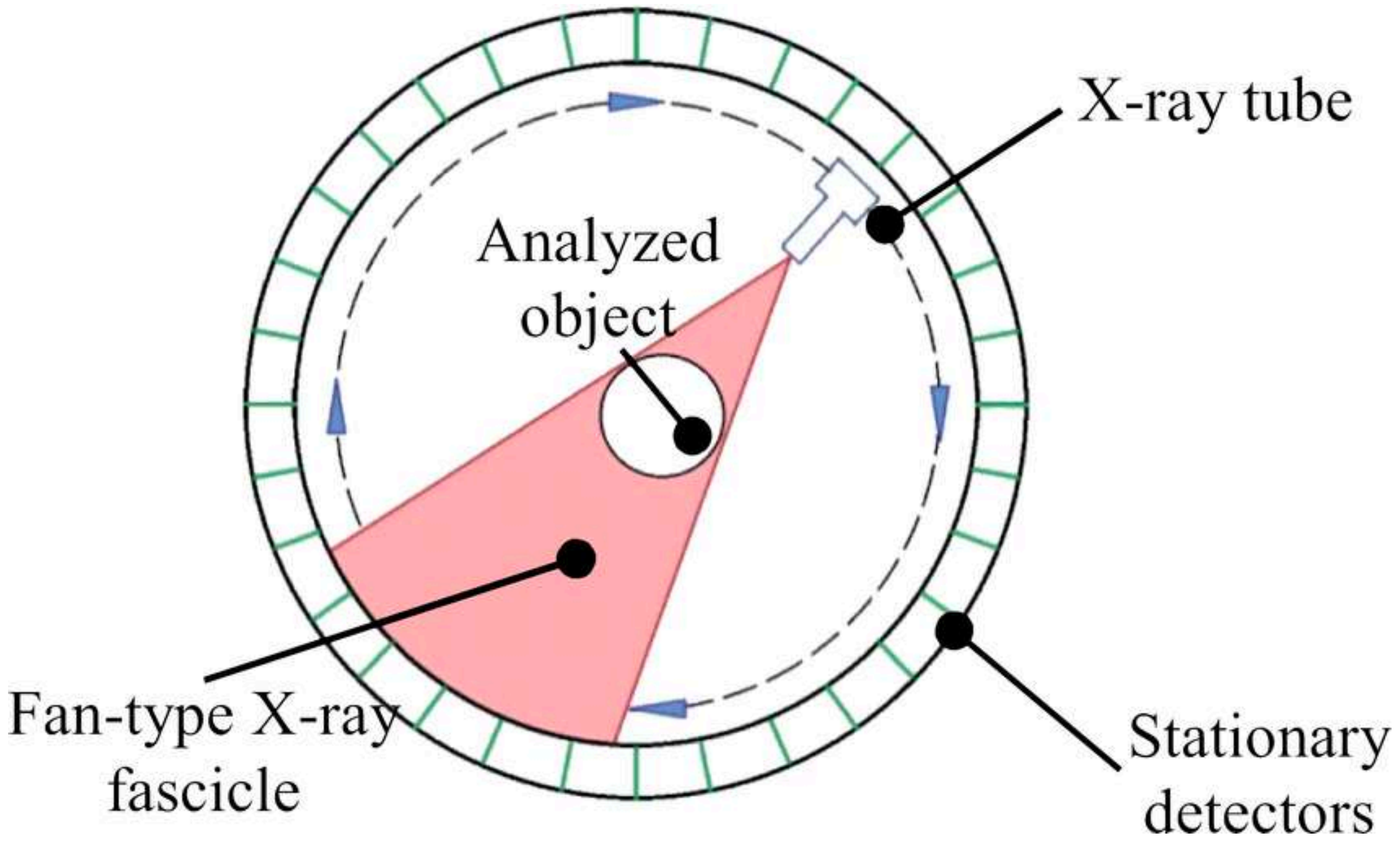
## CT Imaging Background

Problem statement in 2D:

- Projection operator: $A^{\theta,r}[f](t) = \int_0^{L(t)} f(\mathrm{x}_\theta + \mathrm{n}s)ds$
- CT image reconstruction: $p = Af + \epsilon$
- Measurement noise and dose $\epsilon \sim \mathcal{N}(0,\sigma)$

$$\sigma \propto \frac{1}{\sqrt{n_{\max} d \exp(-P)}}$$

- $n_{\max}$ : the maximum number of photons generated by source
- $d$ : the X-ray dose
- $P$ : the average intensity of measurement
- Need to balance between the noise level and the dose we use, which is actually the balance between accuracy and safety



# Learning to Scan

## CT Imaging Background

Problem statement in 2D:

- Projection operator: $A^{\theta,r}[f](t) = \int_0^{L(t)} f(\mathrm{x}_\theta + \mathrm{n}s)ds$
- **ILL POSED** CT image reconstruction:

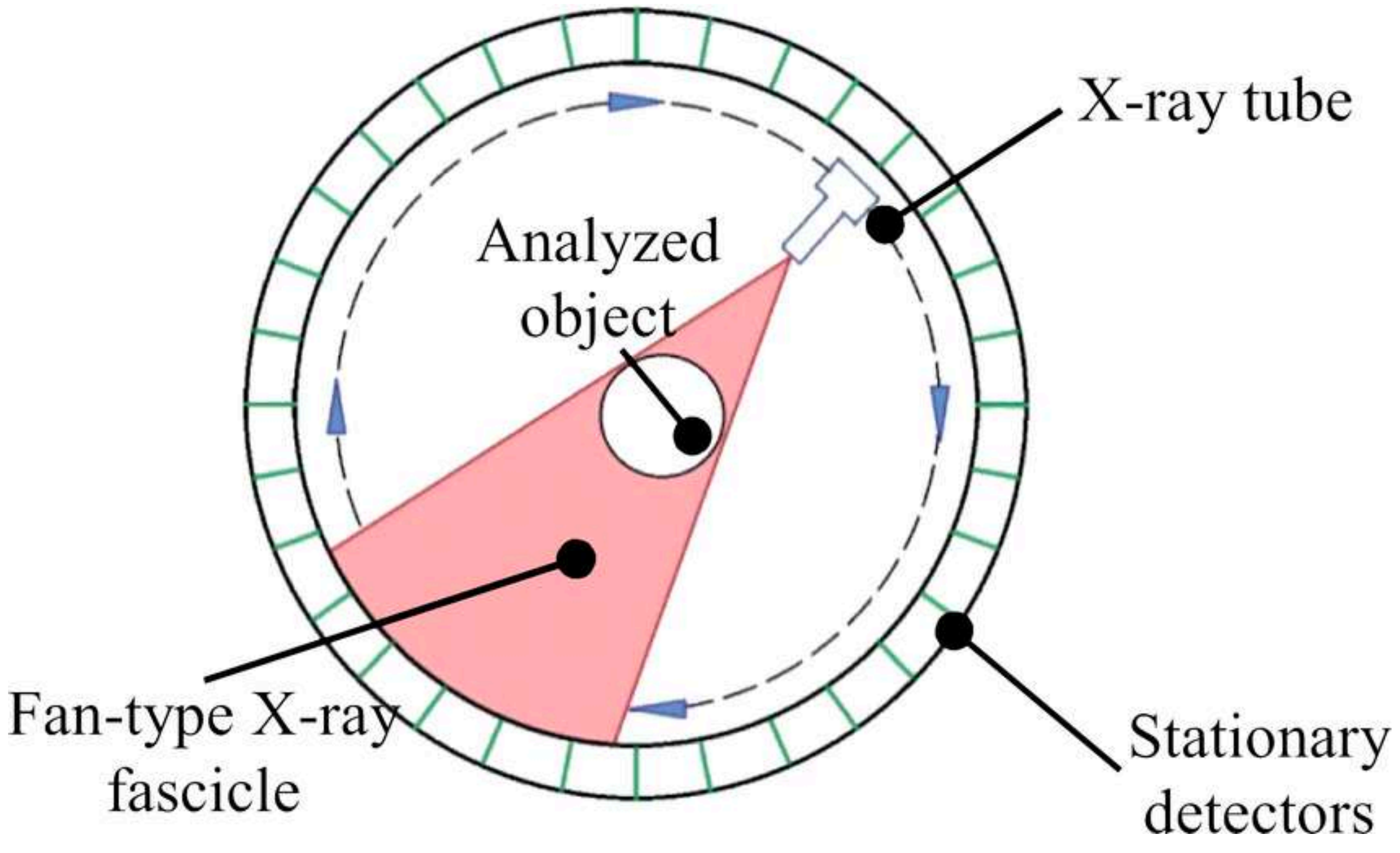
$$p = Af + \epsilon$$

- Numerical solution:

$$\min_f \frac{1}{2}\|p - Af\|_2^2 + \lambda R(f)$$

  with appropriate regularization $R(f) = \|\nabla f\|_1$ (TV) or $R(f) = \|Wf\|_1$ (wavelet transform W)

- Optimization solved by ADMM



# Learning to Scan

**Key question: How can we improve the imaging efficiency and safety simultaneous?**

## Existing effort:

- Faster optimization
- Better regularization
- Compressive sensing to reduce the total dose of X-ray

## Attention not paid

- Uniform dose per direction v.s. personalized dose assignment

# Learning to Scan

## Personalized Dose Assignment

- Problem statement:
  - Fix the total dose of X-ray
  - Obtain the optimized imaging quality
- Assumptions:
  - Flexibility to put sources wherever needed
  - Immediate imaging result available
- Method:
  - Choose source locations and the dose for each location adaptive to imaging targets
  - A combinatorial optimization problem (**NP-Hard**)
  - This is a sequential decision making process

# Learning to Scan

## Markov Decision Process Formulation of Personalized Scanning

- **State** at time t:

$$\overrightarrow{s}_t = (s_1, s_2, \ldots, s_t)$$

where $s_t = (p_t, d_t^{ac}, d_t^{rest})$

- $p_t$ : the collected measurement at time t
- $d_t^{ac} \in \mathbb{R}^N$ : the used dose distribution up to time t
- $d_t^{rest} \in \mathbb{R}$ : the remaining dose we can use
- $N$ : the number of possible source locations

# Learning to Scan

## Markov Decision Process Formulation of Personalized Scanning

- **Action** at time t:

$$a_t = (a_t^{angle}, a_t^{dose})$$

- $a_t^{angle} \in \mathbb{R}^N$ : the angle chosen at time t as a one-hot vector
- $a_t^{dose} \in [0,1]$ : the fraction of dose applied at time t
- Terminate state applied when the total used does exceeds the total allowed dose.

# Learning to Scan

## Markov Decision Process Formulation of Personalized Scanning

- **Reward** at time t:

$$r_t = \text{PSNR}(I_t, I) - \text{PSNR}(I_{t-1}, I)$$

- $I$ : the groundtruth image
- $I_t$ : the reconstructed image at time t
- $\text{PSNR}(\hat{I}, I)$ : the Peak Signal to Noise Ration of the reconstructed image $\hat{I}$
- $I_t$ computed via SART (Algebraic reconstruction techniques, Gordon et al. 1970) without hyperparameter tuning

# Learning to Scan

**Markov Decision Process Formulation of Personalized Scanning**

## ○ The transition model:

- $\overrightarrow{s}_{t+1}$ is the concatenation of $\overrightarrow{s}_t$ and $s_{t+1} = (p_{t+1}, d^{ac}_{t+1}, d^{rest}_{t+1})$
- Simulated measurement $p_{t+1} = p^{true}_{t+1} + \epsilon$ with $\epsilon \sim \mathcal{N}(0,\sigma)$ and $\sigma \propto \dfrac{1}{\sqrt{n_{\max} d \exp(-P)}}$, where $P$ is the average of $p^{true}_{t+1}$
- $d^{ac}_{t+1} = d^{ac}_t + 1_{\mathrm{a}^{\mathrm{angle}}_{\mathrm{t}}} \cdot \mathrm{a}^{\mathrm{dose}}_{\mathrm{t}}$
- $d^{rest}_{t+1} = d^{rest}_t - a^{dose}_t$

# Learning to Scan

## Deep RL Algorithm

Notations:

- Goal of the task: maximize accumulated rewards $G(\pi) = \sum_{t=1}^{\infty} \gamma^t r_t$
- A policy $\pi_\theta(s, a)$ is parametrized by a network $\theta$
- Value functions $V^{\pi_\theta}(s) = \mathrm{E}_{\pi_\theta}[\mathrm{G} \mid \mathrm{s}_0 = \mathrm{s}]$ and $Q^{\pi_\theta}(s, a) = \mathrm{E}_{\pi_\theta}[\mathrm{G} \mid \mathrm{s}_0 = \mathrm{s}, \mathrm{a}_0 = \mathrm{a}]$
- Goal of RL: $\max_\theta \eta(\pi_\theta) := \mathrm{E}_{\mathrm{s}_0 \sim \rho(\mathrm{s})}[\mathrm{V}^{\pi_\theta}(\mathrm{s})]$
- Policy gradient method: $\nabla_\theta \eta(\pi_\theta) = \mathrm{E}_{(\mathrm{s},\mathrm{a}) \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(\mathrm{a} \mid \mathrm{s}) \cdot \mathrm{Q}^{\pi_\theta}(\mathrm{s}, \mathrm{a})]$ but variance is high
- Proximal policy optimization method: reduced variance but heavily on the effectiveness of its exploratory policy search

# Learning to Scan

## Deep RL Algorithm

Proximal policy optimization method:

- Andvantage function: $A^{\pi_\theta}(s, a) = Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$
- Proximal control function: $\dfrac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}$
- Proximal objective function in each step:

$$J^{PPO}(\theta) = \mathrm{E}_{(\mathrm{s,a})\sim\pi_{\theta_{\mathrm{old}}}}\left[\min\left(\mathrm{b}_\theta(\mathrm{s, a})\mathrm{A}^{\pi_{\theta_{\mathrm{old}}}}(\mathrm{s, a}), \quad \mathrm{clip}(\mathrm{b}_\theta(\mathrm{s, a}), 1-\epsilon, 1+\epsilon)\mathrm{A}^{\pi_{\theta_{\mathrm{old}}}}(\mathrm{s, a})\right)\right]$$

where $\mathrm{clip(x, a, b)} = \mathrm{max(min(x, b), a)}$.

# Learning to Scan

$$\pi_\theta(a \mid \vec{s}) = \pi_\theta(a^{dose}, a^{angle} \mid \vec{s})$$

- $= \pi_\theta(a^{angle} \mid \vec{s})\pi_\theta(a^{dose} \mid \vec{s}, a^{angle})$
- $a_t^{dose} \sim \mathcal{N}(\mu^{dose}, \sigma^{dose})$

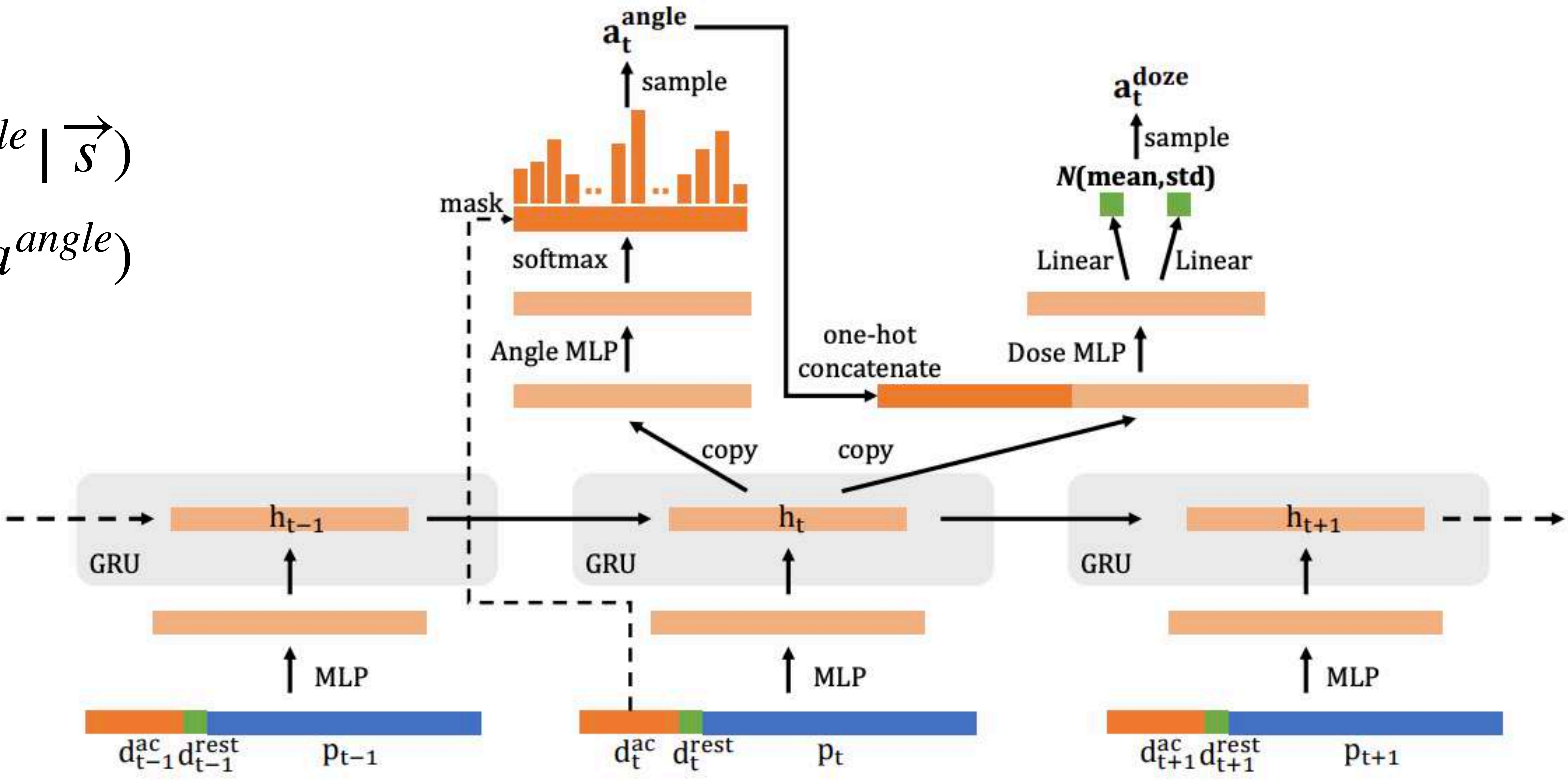



FIGURE 1. Policy network architecture. Each MLP contains two hidden layers with 512 neurons. We use a multi-layer GRU which contains 3 recurrent layers and each layer has 256 neurons. The Angle MLP has one hidden layer of 512 neurons, and the Dose MLP has 2 hidden layers with 512 neurons.

# Learning to Scan

## Experiments

| Reconstruction Method | | RL-AD | DS-ED | UF-AEC |
|---|---|---|---|---|
| Noise 1 | | | | |
| SART | PSNR | **23.48(0.47)** | 23.30(0.64) | 23.01(0.64) |
| | SSIM | **0.424(0.020)** | 0.403(0.023) | 0.391(0.022) |
| TV | PSNR | **23.85(0.42)** | 23.75(0.41) | 23.63(0.38) |
| | SSIM | **0.582(0.030)** | 0.579(0.030) | 0.578(0.028) |
| WF | PSNR | **25.14(0.40)** | 25.05(0.42) | 24.91(0.39) |
| | SSIM | **0.659(0.027)** | 0.652(0.027) | 0.649(0.026) |
| PD-net | PSNR | **30.87(0.64)** | 30.44(0.51) | 30.23(0.46) |
| | SSIM | **0.776(0.036)** | 0.771(0.029) | 0.773(0.028) |

# Learning to Scan

## Experiments

| Noise 2 | | | | |
|---|---|---|---|---|
| SART | PSNR | **23.15(0.48)** | 22.91(0.53) | 22.60(0.64) |
| | SSIM | **0.413(0.020)** | 0.390(0.024) | 0.378(0.024) |
| TV | PSNR | **23.74(0.40)** | 23.50(0.36) | 23.27(0.40) |
| | SSIM | **0.580(0.030)** | 0.576(0.030) | 0.573(0.028) |
| WF | PSNR | **24.98(0.29)** | 24.84(0.41) | 24.68(0.39) |
| | SSIM | **0.657(0.027)** | 0.649(0.026) | 0.646(0.026) |
| PD-net | PSNR | **30.78(0.64)** | 30.35(0.51) | 30.15(0.77) |
| | SSIM | **0.774(0.037)** | 0.769(0.030) | 0.771(0.029) |

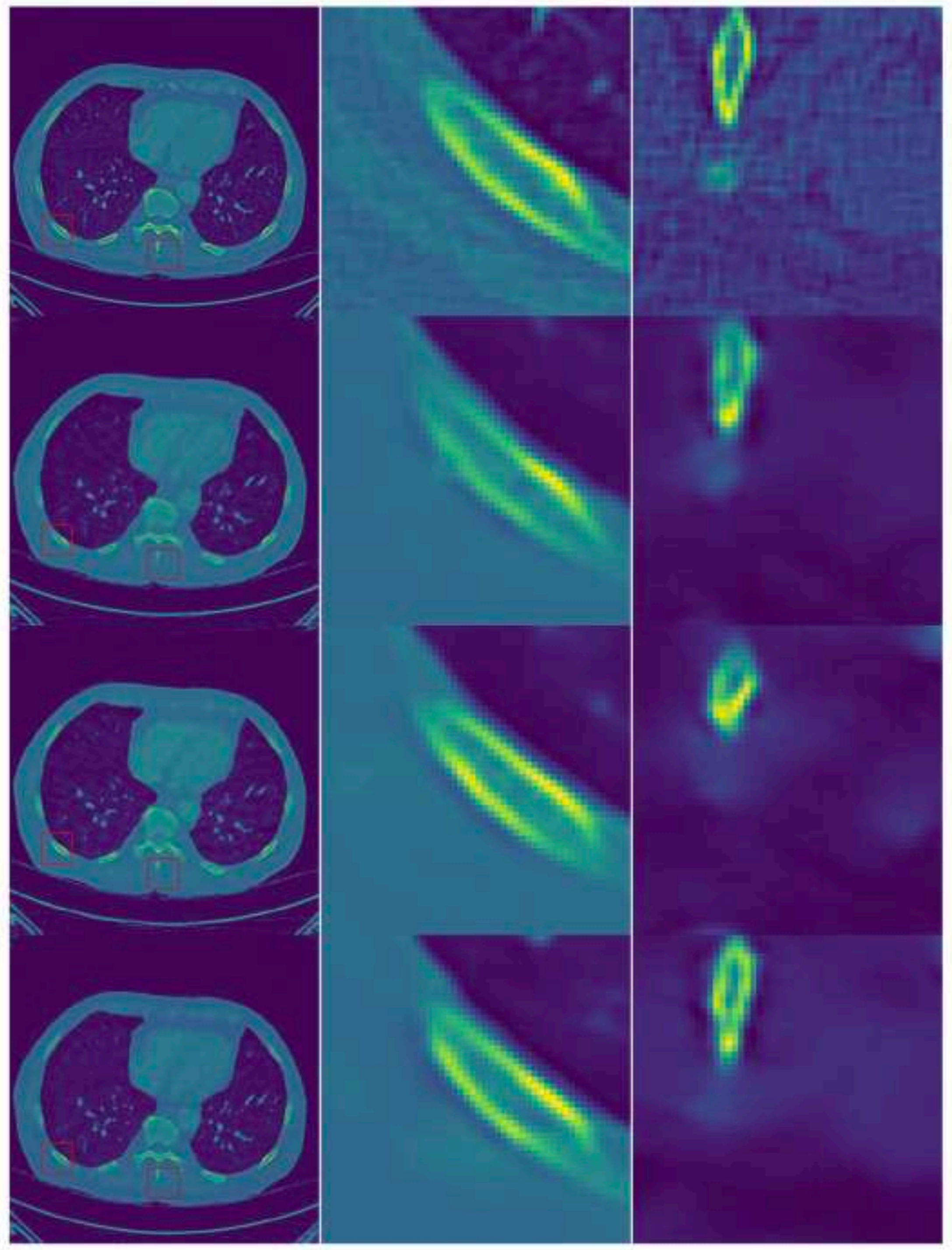
# Learning to Scan

## Experiments

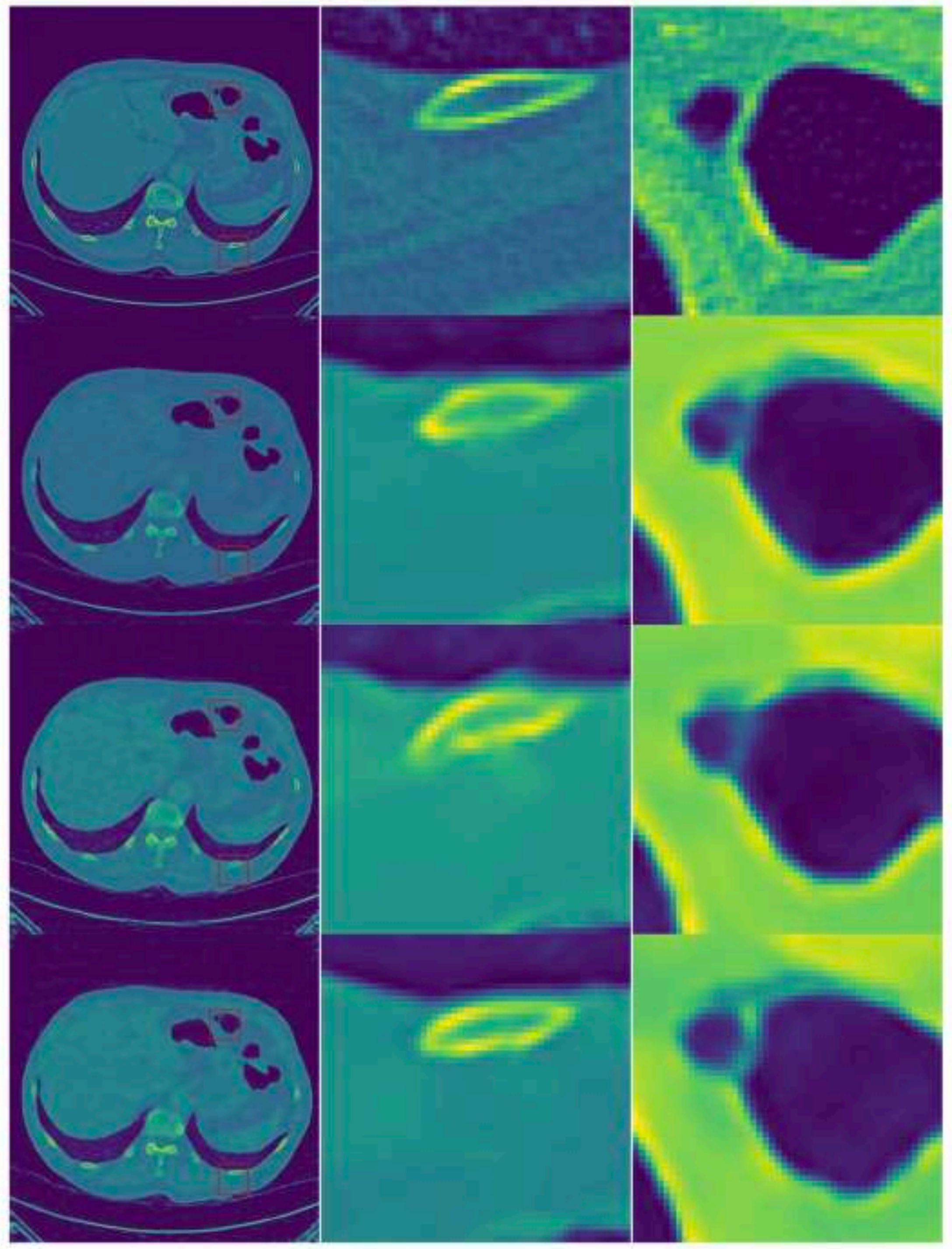
| | | | | |
|---|---|---|---|---|
| SART | PSNR | **20.71(0.55)** | 20.26(0.72) | 19.83(0.66) |
| | SSIM | **0.334(0.026)** | 0.304(0.030) | 0.291(0.029) |
| TV | PSNR | **21.73(0.57)** | 21.43(0.48) | 21.08(0.47) |
| | SSIM | **0.568(0.027)** | 0.555(0.026) | 0.545(0.026) |
| WF | PSNR | **23.35(0.48)** | 23.05(0.51) | 22.72(0.55) |
| | SSIM | **0.636(0.0326)** | 0.616(0.027) | 0.605(0.028) |
| PD-net | PSNR | **29.97(0.66)** | 29.56(0.51) | 29.36(0.47) |
| | SSIM | **0.753(0.038)** | 0.746(0.032) | 0.747(0.031) |
| Inference Time (s) | | 0.46(0.02) | 0.21(0.008) | 0.20(0.001) |

# Learning to Scan

## Experiments



(a)

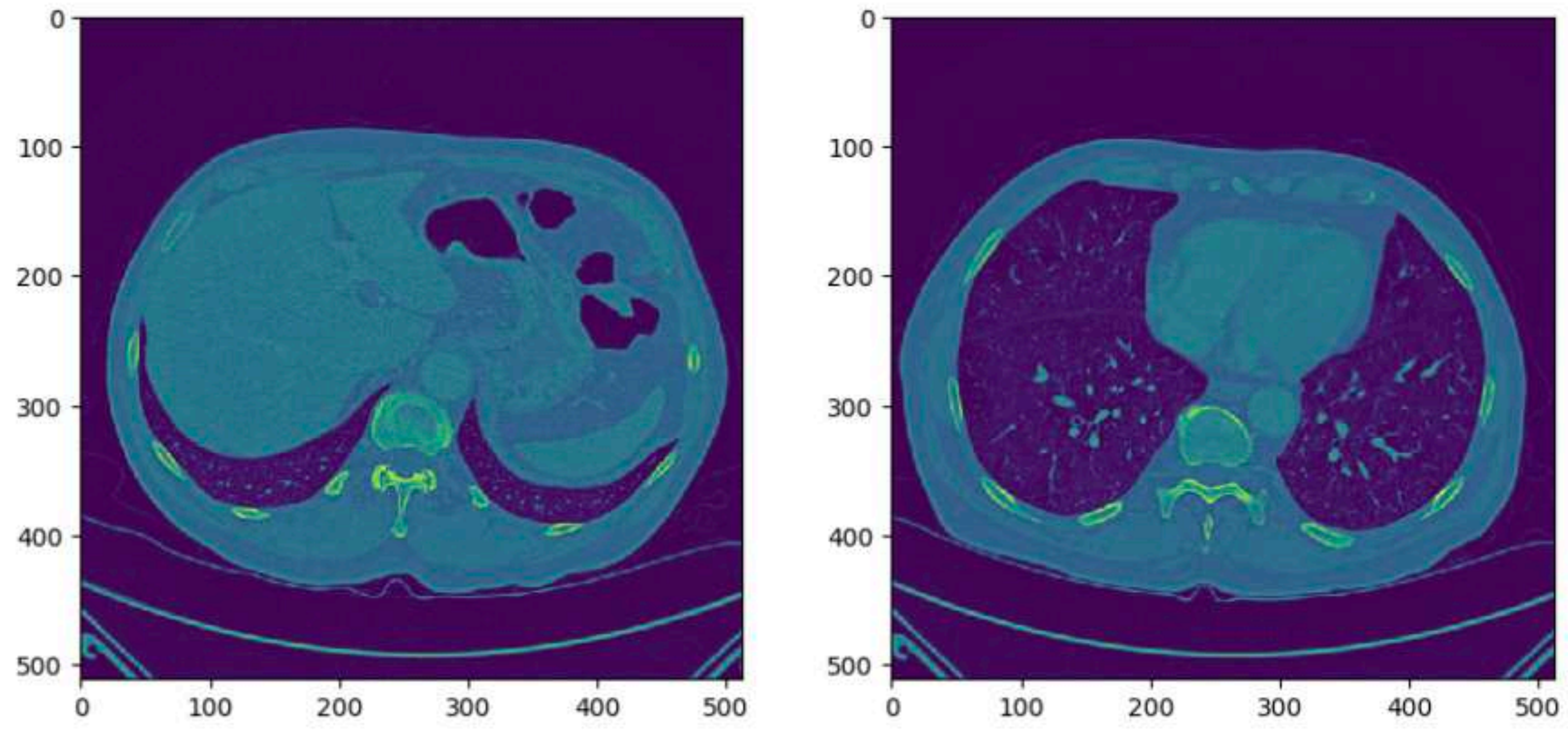


(b)

FIGURE 3. Two examples of the reconstructed images. The top row contains the ground truth images and their zoom-in views. The second through the fourth row contain results from UF-AEC, DS-ED and RL-AD respectively, and combined with PD-net's reconstruction. Note that RL-AD selects 65 measurement angles for the subject in (a) and 54 measurement angles for the subject in (b).

# Learning to Scan

## Experiments




(a) (b)

FIGURE 5. (a): an example image that takes 54 measurements. (b): an example image that takes 64 measurements. We can see that the images for which RL selects more measurement angles contains more structures.

# Learning to Scan

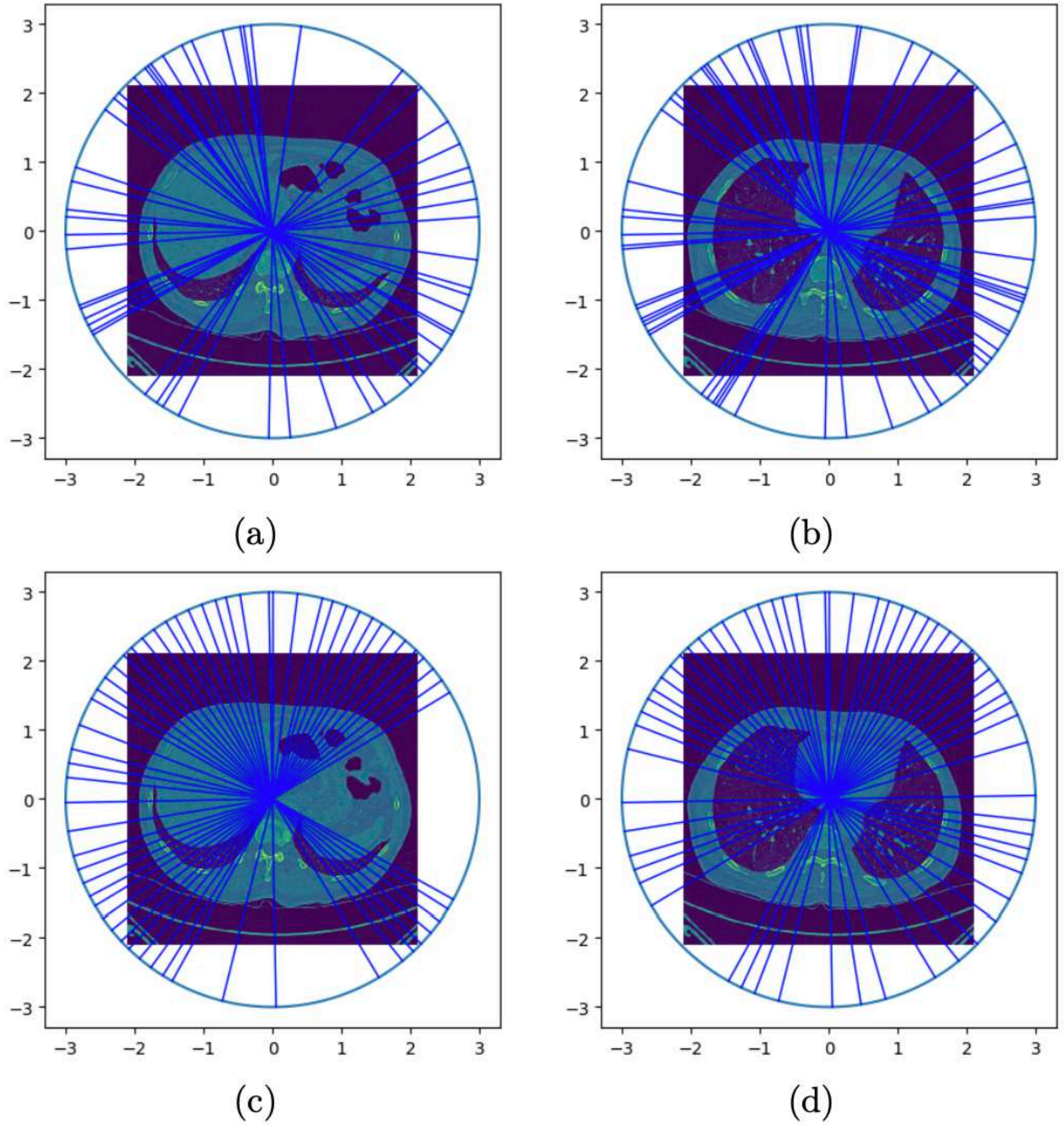



FIGURE 6. The angle selection of the CT image in Figure 5. Top row: RL-AD, bottom row: DS-ED. The lines show the selected angles.

# Reinforced Inverse Scattering

Hanyang Jiang, Yuehaw Khoo, Haizhao Yang, arxiv:2206.04186

# Inverse Scattering

- Applications: non-destructive testing, medical imaging, seismic imaging, etc.
- Helmholtz equation: $$Lu(x) := \left( -\nabla - \frac{\omega^2}{c(x)^2} \right) u(x) = 0$$

  with a given frequency $\omega$ and unknown speed $c(x)$
- Introduce $$\frac{\omega^2}{c(x)^2} = \frac{\omega^2}{c_0(x)^2} + \eta(x) \qquad \text{and} \qquad L_0 := -\nabla - \frac{\omega^2}{c_0(x)^2}$$

  with a background speed $c_0(x)$ given in applications
- A parametric PDE : $$\left( -\nabla - \frac{\omega^2}{c(x)^2} \right) u(x) = (L_0 - \eta(x)) u(x) = 0$$

  with parameter $\eta(x)$
- Goal: gather information of $u(x)$ using sensors to reconstruct $\eta(x)$

# Sensing Setup in Inverse Scattering

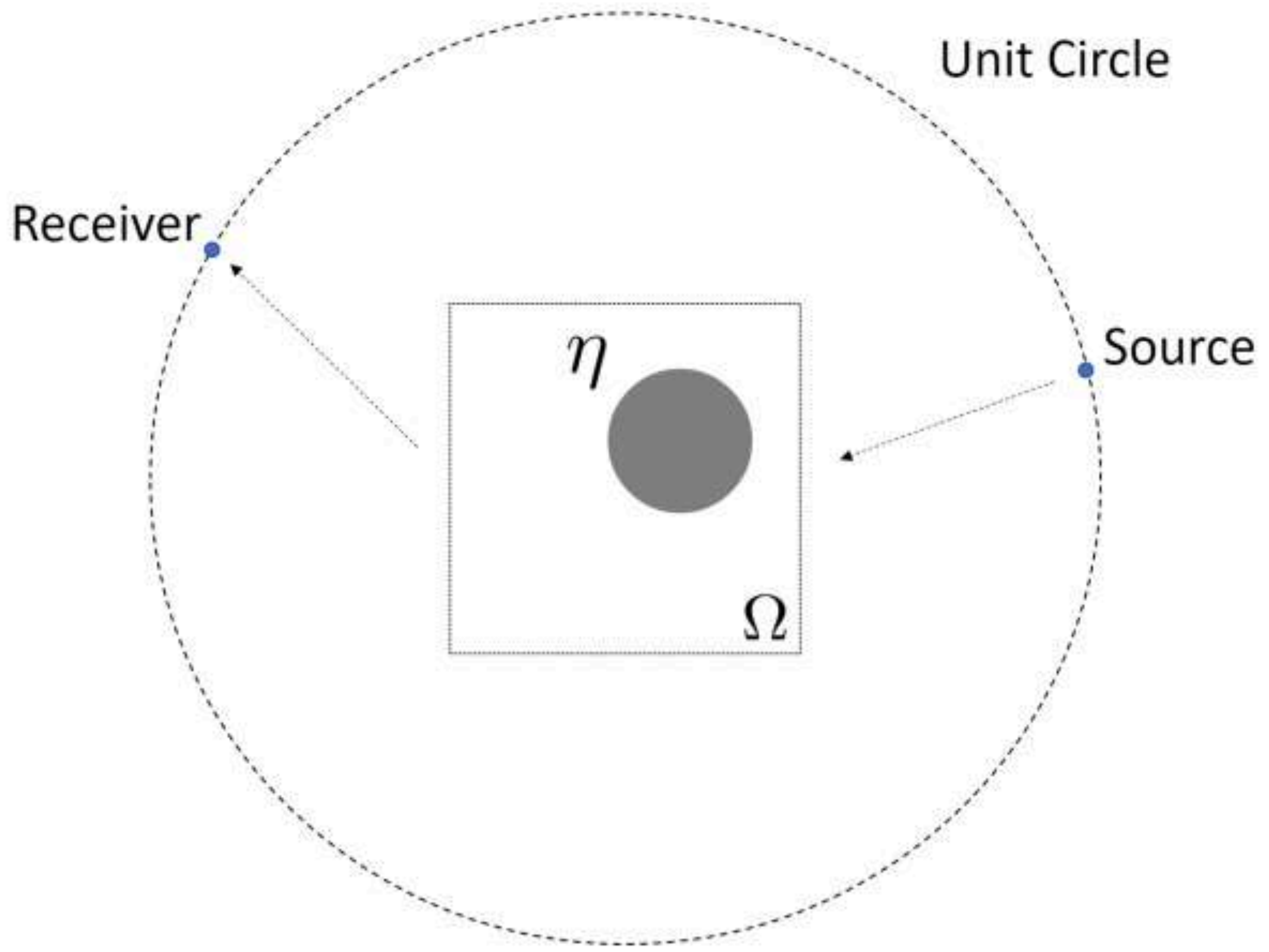
Problem statement in 2D:

- Notation: $G_0 = L_0^{-1}$ (known) and $E = \mathrm{diag}(\eta)$ (unknown)
- Observed data at a source location $\sigma_1$ and a receiver location $\sigma_2$:

$$d(\sigma_1, \sigma_2) = \sum_{x \in \Omega} \sum_{y \in \Omega} e^{-i\omega\sigma_2 \cdot x} \left(E + EG_0E + \cdots\right)(x, y) e^{i\omega\sigma_1 \cdot y}$$

$$:= F_{\omega, \mathscr{R}, \Sigma}(\eta)(\sigma_1, \sigma_2)$$

- $\omega$ : incident wave frequency
- $\mathscr{R}$ : the set of receiver locations
- $\Sigma$ : the set of source locations




The sensing process for
a far-field pattern problem.

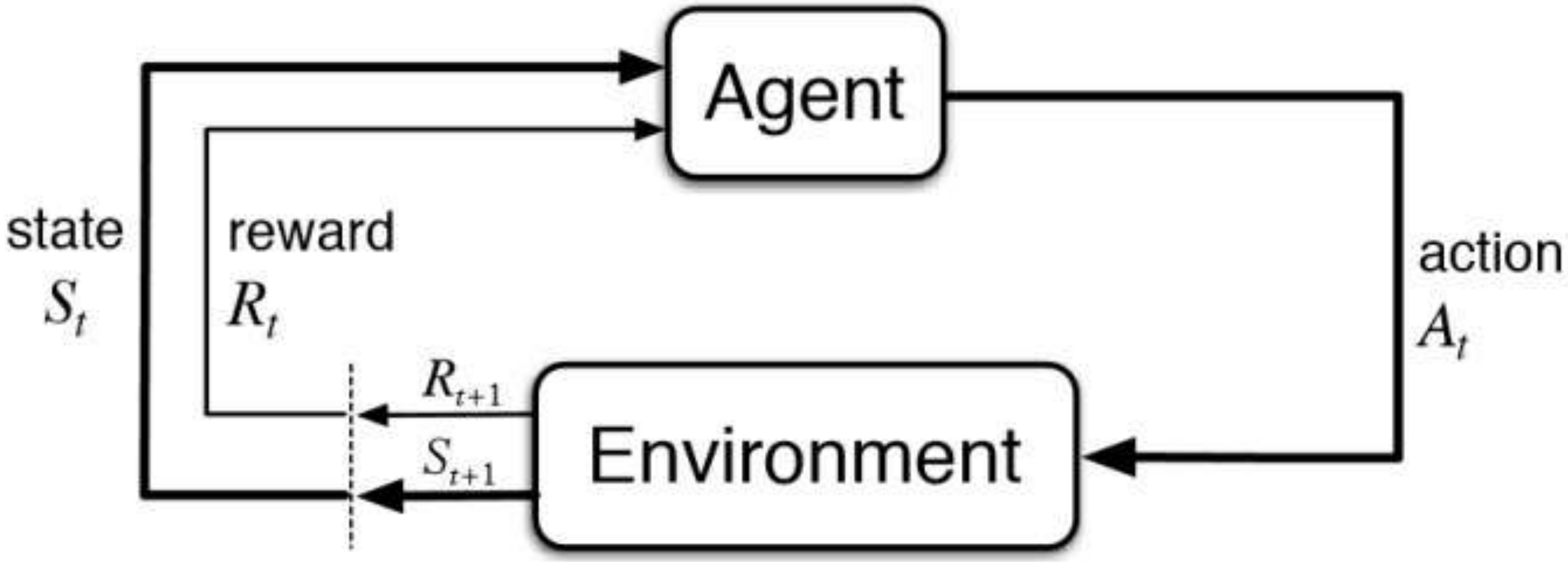
# Reconstruction in Inverse Scattering

- Regularized optimization for ill-possedness

$$\min_{\eta} \; \|d - F_{\omega,\mathscr{R},\Sigma}(\eta)\|_2^2 + \lambda\|\eta\|_1$$

- $\eta$ : unknown parameter to be reconstructed
- $d$ : observed data
- $\omega$ : incident wave frequency
- $\mathscr{R}$ : the set of receiver locations
- $\Sigma$ : the set of source locations
- **Literature**: better regularization, faster optimization, multi-frequency
- **Our angle**: "personalized" sensing strategy to improve observed data for precision imaging

# Learning for Sensing




By Richard S. Sutton and Andrew G. Barto.

- **Goal:** Learning how to select sensor positions and wave frequencies in inverse scattering
- **Ideas:**
  1. Reformulate the sequential (selection, observation, reconstruction) procedure as a sequence of (action, state, reward)
  2. Reformulate the decision strategy for selection and observation as the policy to take actions
  3. "Efficiently improving" the reconstruction quality as the reward
  4. Learning to balance local and global reconstruction quality

# Learning for Sensing

## Personalized Strategy for Precision Imaging

- Problem statement:
  - Fix the total number of sensors and incident waves
  - Obtain the optimized imaging quality
- Assumptions:
  - Flexibility to put sources wherever needed and choose arbitrary frequencies
  - Immediate imaging result available
  - Several ground truth images available
- Method:
  - Choose sensor locations and frequencies for each location adaptive to imaging targets
  - A combinatorial optimization problem (**NP-Hard**)
  - This is a sequential decision making process

# Learning for Sensing

## Markov Decision Process Formulation of Personalized Scanning

- **State** at time t:

$$\overrightarrow{s}_t = (s_1, s_2, \ldots, s_t)$$

where $s_t = (d_t, u_t, T + 1 - t)$

- $d_t$ : the collected measurement at time t
- $u_t$ : the sensor position set up to time t
- $T$ : the total number of sensing operations

# Learning for Sensing

**Markov Decision Process Formulation of Personalized Scanning**

- **Action** at time t:

$$a_t = (\sigma^{a_t}, \omega^{a_t})$$

- $\sigma^{a_t}$ : the position of sensors chosen at time t
- $\omega^{a_t}$ : the frequency applied at time t
- Terminate state applied when the total number of sensing operations has been reached

# Learning for Sensing

## Markov Decision Process Formulation of Personalized Scanning

- **Reward** at time t:

$$r_t = \mathrm{PSNR}(\eta_t, \eta) - \mathrm{PSNR}(\eta_{t-1}, \eta)$$

- $\eta$ : the groundtruth image
- $\eta_t$ : the reconstructed image at time t
- $\mathrm{PSNR}(\hat{\eta}, \eta)$ : the Peak Signal to Noise Ration of the reconstructed image $\hat{\eta}$
- $\eta_t$ computed via regularized reconstruction without hyperparameter tuning

# Learning for Sensing

## Markov Decision Process Formulation of Personalized Scanning

### ○ The transition model:

- $\overrightarrow{s}_{t+1}$ is the concatenation of $\overrightarrow{s}_t$ and $s_{t+1} = (d_{t+1}, u_{t+1}, T - t)$
- $u_{t+1} = u_t + \omega^{a_t}\sigma^{a_t}$
- $d_{t+1} = F_{\omega^{a_t}, \mathscr{R}^{a_t}, \Sigma^{a_t}}(\eta)$

# Learning for Sensing

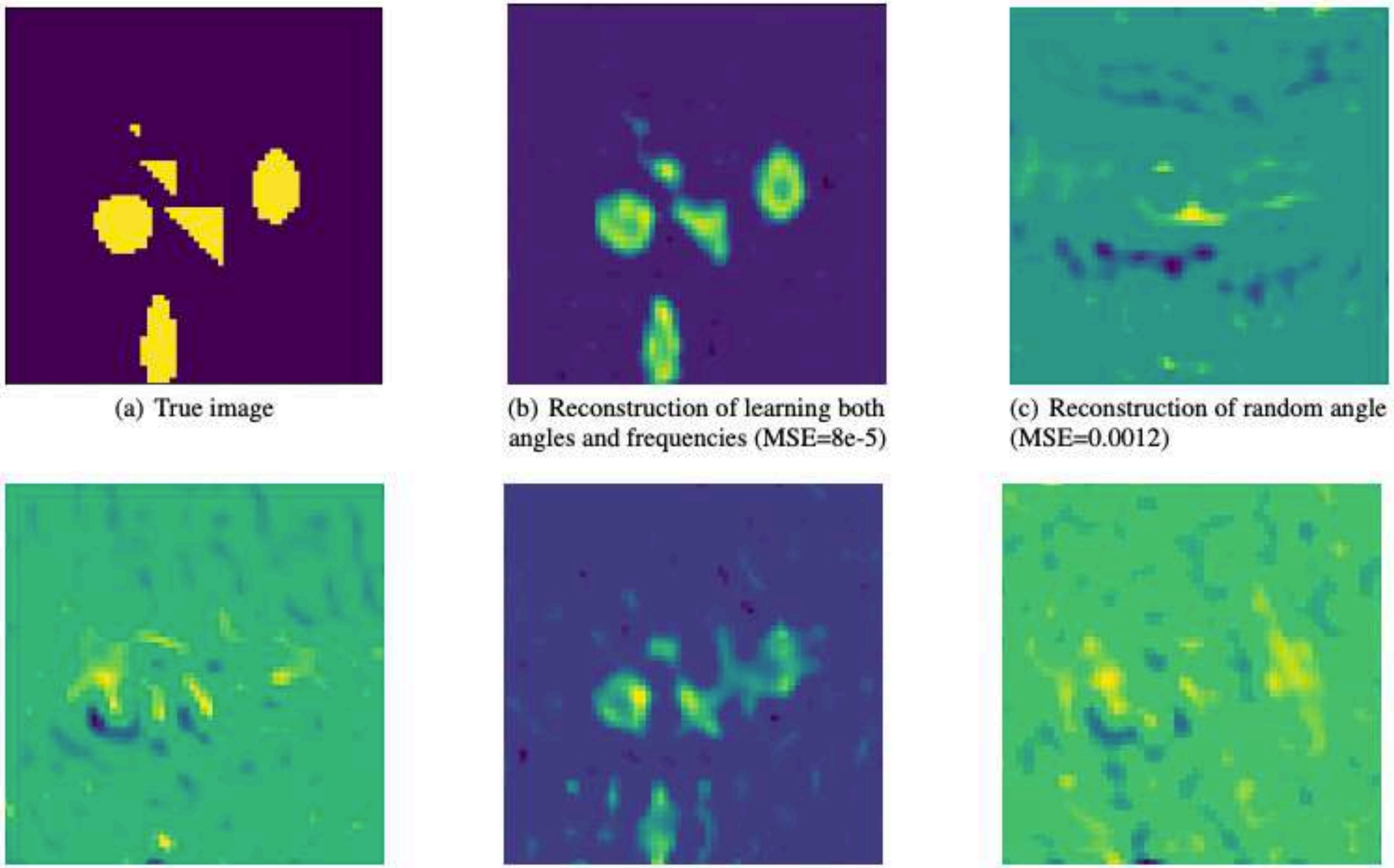
**Deep RL Algorithm**

Notations:

- Goal of the task: maximize accumulated rewards $G(\pi) = \sum_{t=1}^{\infty} \gamma^t r_t$
- A policy $\pi_\theta(s, a)$ is parametrized by a network $\theta$
- Value function $V^{\pi_\theta}(s) = \mathrm{E}_{\pi_\theta}[\mathrm{G} \,|\, \mathrm{s}_0 = \mathrm{s}]$
- Goal of RL: $\max_\theta \mathscr{L}(\theta) := \mathrm{E}_{\mathrm{s}_0 \sim \rho(\mathrm{s})}[\mathrm{V}^{\pi_\theta}(\mathrm{s})]$

# Reinforced Inverse Scattering

## Experiments



(a) True image

(b) Reconstruction of learning both angles and frequencies (MSE=8e-5)

(c) Reconstruction of random angle (MSE=0.0012)

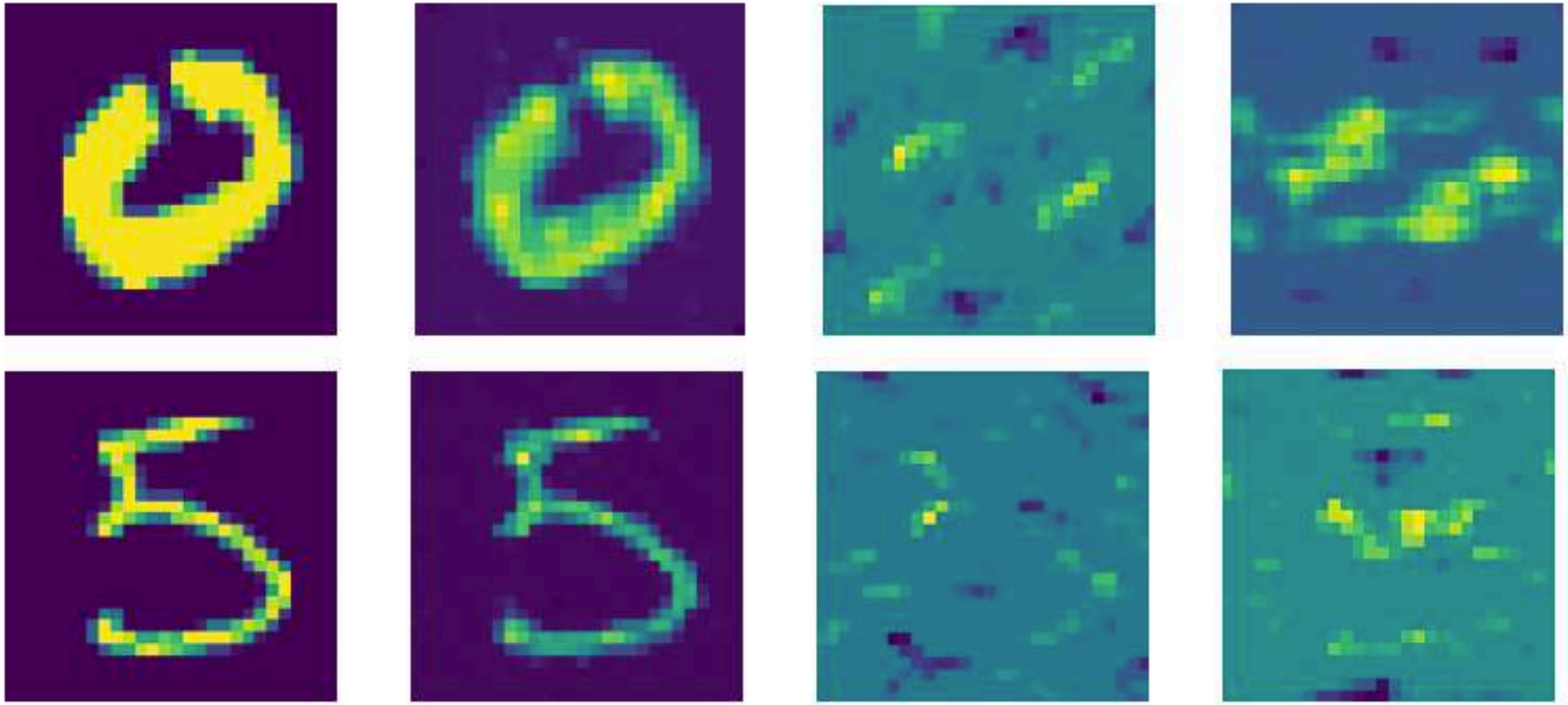
(d) Reconstruction of uniform angle (MSE=0.0017)

(e) Reconstruction of learning angles only (MSE=3e-4)

(f) Reconstruction of learning frequencies only (MSE=2e-3)

# Reinforced Inverse Scattering

## Experiments



(a) True image

(b) Reconstruction of learning both angles and frequencies (MSE=2e-4)

(c) Reconstruction of random angle (MSE=5e-3)

(d) Reconstruction of uniform angle (MSE=3e-3)