Applied Mathematics for Modern Challenges
Vol. 5, 2025, pp. 36-63
doi:10.3934/ammc.2025010

DYNAMIC ANGLE SELECTION IN X-RAY CT: A REINFORCEMENT LEARNING APPROACH TO OPTIMAL STOPPING

Tianyuan Wang✉*[1,2], Felix Lucka✉[1], Daniël M. Pelt✉[2],
K. Joost Batenburg✉[1,2] and Tristan van Leeuwen✉[1,3]

[1]Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, 1098 XG, The Netherlands

[2]Leiden Institute of Advanced Computer Science, Leiden Universiteit, Leiden, The Netherlands

[3]Mathematics Institute, Utrecht University, Campus-Boulevard 30,
Utrecht, 3584 CD, The Netherlands

(Communicated by Tatiana A. Bubba)

Abstract. In industrial X-ray Computed Tomography (CT), the need for rapid in-line inspection is critical. Sparse-angle tomography plays a significant role in this by reducing the required number of projections, thereby accelerating processing and conserving resources. Most existing methods aim to balance reconstruction quality and scanning time, typically relying on fixed scan durations. Adaptive adjustment of the number of angles is essential; for instance, more angles may be required for objects with complex geometries or noisier projections. The concept of optimal stopping, which dynamically adjusts this balance according to varying industrial needs, remains overlooked. Building on our previous work, we integrate optimal stopping into sequential Optimal Experimental Design (sOED) and Reinforcement Learning (RL). We propose a novel method for computing the policy gradient within the Actor–Critic framework, enabling the development of adaptive policies for informative angle selection and scan termination. Additionally, we evaluate whether policies trained in simulation transfer to experimental X-ray CT data and provide initial evidence on laboratory data. Trained on synthetic data, the model shows consistent behavior on experimental scans. This supports flexible CT operation and expands the applicability of sparse-view tomography in industrial settings.

1. **Introduction.**

X-ray Computed Tomography (CT) enables inline industrial inspection through three-dimensional reconstruction. However, fast and adaptive CT scanning is essential to make its widespread industrial application feasible. Previous studies have shown that not all projections are equally informative for certain objects [19, 33]. Most research has focused on sequentially selecting informative angles to enhance efficiency. This sequential Optimal Experimental Design (sOED) is often described within a Bayesian framework [22, 9, 25], where angles are chosen

2020 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. X-ray CT, optimal stopping, sequential optimal experimental design, policy gradient, reinforcement learning, simulation-to-real-world.

Tianyuan Wang is supported by the European Union H2020-MSCA-ITN-2020 under grant agreement no. 956172 (xCTing).

*Corresponding author: Tianyuan Wang.

to maximize information gain. Information gain is typically quantified by comparing the prior and posterior distributions of the reconstruction or by assessing the similarity between the reconstructed image and the ground truth.

Batenburg *et al.* [4] and Dabravolski *et al.* [11] used a set of template images composed of Gaussian blobs to represent samples from the prior distribution. They introduced an upper bound [3] to approximate the information gain, which reflects the diameter of the solution set. Burger *et al.* [8] employed classical "alphabetic criteria" for OED, such as "A-" and "D-" optimality, using the trace or determinant of the covariance matrix of the posterior distribution as summary statistics [18]. Additionally, they used a Gaussian distribution as a prior, updating the posterior after selecting each angle. Building on this approach, Helin *et al.* [16] introduced a Total Variation (TV) prior to enhance edges in reconstructions. The non-Gaussian TV prior was approximated as a Gaussian distribution using lagged diffusivity iteration. Furthermore, Barbano *et al.* [5] utilized a deep image prior as the reconstruction method and linearized the network to approximate the posterior distribution as a Gaussian. To avoid the need for a closed-form solution, Elata *et al.* [13] proposed using a diffusion model for CT reconstruction, sampling from the posterior to approximate the posterior covariance matrix.

Recently, policy-based methods from the reinforcement learning community have been introduced into sOED [28, 7, 14]. In the medical CT field, Shen *et al.* [29] trained a gated recurrent unit as a policy network on simulated medical data to map projections to probabilities over the angle space. In previous work, we explored industrial CT applications with very few angles [34]. We trained a policy that maps the current reconstruction to probabilities over the angle space. Additionally, we addressed a specific task—defect detection—by incorporating an extra reward for defect detectability and prior information about defects, enabling the trained policy to identify informative angles to aid in defect detection [35].

Although extensive work has been devoted to selecting informative projection angles, the question of how many angles to acquire is often neglected. This choice can be cast as an optimal stopping problem, a topic well studied in financial mathematics. The optimal expected reward is given by the Snell envelope, the smallest super-martingale that dominates the reward process. The earliest (respectively, latest) optimal stopping time is the first instant at which the immediate reward equals (respectively, exceeds) the continuation value [24, 6, 12].

Additionally, most studies have focused on simulated data rather than experimental X-ray CT data. This is particularly evident in learning-based methods, which rely on training with simulated data, leaving their generalizability to experimental X-ray CT data uncertain.

The contributions of this work include the development of an optimal stopping method to balance the trade-off between experimental costs and experimental goals, such as reconstruction quality. This approach enables both adaptive selection of informative angles and optimal scan termination based on experimental costs. By incorporating a terminal policy into the Actor–Critic framework, we proposed a novel method for computing the policy gradient, jointly optimizing angle selection and the terminal policy. Additionally, we investigated the gap between simulation and real-world applications by evaluating the trained model on experimental X-ray CT data.

The structure of the paper is as follows: The Background section provides an overview of the fundamental concepts and notations related to inverse problems and sOED using reinforcement learning. The Method section introduces our novel approach for computing the policy gradient. The Results section presents the findings from both simulation and experimental X-ray CT data experiments. Finally, the paper concludes with a discussion of the key findings and their implications.

2. Background.

2.1. Forward and inverse problems and optimal experimental design (OED).

Measurements $\boldsymbol{y}(\boldsymbol{\theta})$ are obtained from the ground-truth image (underlying parameters) $\bar{\boldsymbol{x}}$ using a forward operator $\boldsymbol{A}(\boldsymbol{\theta})$, which is determined by the design parameters $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_M\}$. In the case of X-ray CT, the forward operator $\boldsymbol{A}(\boldsymbol{\theta})$ corresponds to the Radon transform for angles $\boldsymbol{\theta}$ [15]. Since the measurement process is subject to noise, we also incorporate a noise term $\boldsymbol{\epsilon}(\boldsymbol{\theta})$ and model the projections as follows:

$$\boldsymbol{y}(\boldsymbol{\theta}) = \boldsymbol{A}(\boldsymbol{\theta})\bar{\boldsymbol{x}} + \boldsymbol{\epsilon}(\boldsymbol{\theta}). \tag{1}$$

The inverse problem involves using the projections $\boldsymbol{y}(\boldsymbol{\theta})$ and the forward operator $\boldsymbol{A}(\boldsymbol{\theta})$ to compute a reconstruction $\widehat{\boldsymbol{x}}(\boldsymbol{\theta})$ as an estimate of the ground-truth image $\bar{\boldsymbol{x}}$. Figure (1, a) illustrates the forward and inverse processes. However, solving inverse problems is often challenging due to their ill-posedness [23]. The accuracy of the reconstructed ground-truth image is influenced by the design parameters $\boldsymbol{\theta}$, making their optimal selection crucial. OED is employed to select the most informative angles to acquire the corresponding projections. To quantify the accuracy of the reconstruction, a utility function is defined within the OED framework. The optimal design $\boldsymbol{\theta}^*$ is then obtained by maximizing the expected value of this utility function over the design space, taking into account the projections $\boldsymbol{y}(\boldsymbol{\theta})$ and the ground truth image $\bar{\boldsymbol{x}}$ [22, 27]. As shown in Figure (1, b), we plot reconstruction quality—measured by PSNR—versus the number of projection angles M (up to 50). Reconstructions $\widehat{\boldsymbol{x}}(\boldsymbol{\theta})$ are computed with the Simultaneous Iterative Reconstruction Technique (SIRT) under a nonnegativity constraint (150 iterations). The orange curve corresponds to a selection that, at each step, exhaustively evaluates all remaining angles and chooses the most informative one; the blue curve uses uniformly spaced angles. For the triangle phantom (left), the selection consistently outperforms the uniform strategy, especially at small M. The uniform strategy occasionally picks informative directions, which explains the oscillations in its performance. For the Shepp–Logan phantom (right), which is approximately rotationally symmetric and lacks a preferred direction, the gap between the two strategies is small. Across both phantoms we observe diminishing returns: beyond a moderate number of angles, further increases in M yield only marginal PSNR gains. When acquisition is costly, such small gains may not justify additional projections, motivating an optimal stopping rule to balance reconstruction quality against experimental cost.

2.2. Sequential optimal experimental design (sOED) and reinforcement learning.

Traditional experimental design is typically performed a-priori, with all optimal design parameters selected simultaneously. Because it ignores the feedback obtained after each parameter choice, this approach cannot support a-posteriori selection, potentially overlooking information that could refine subsequent decisions [28].

The sOED extends the concept of traditional OED by allowing the design parameters to be determined sequentially, based on the data acquired from previous

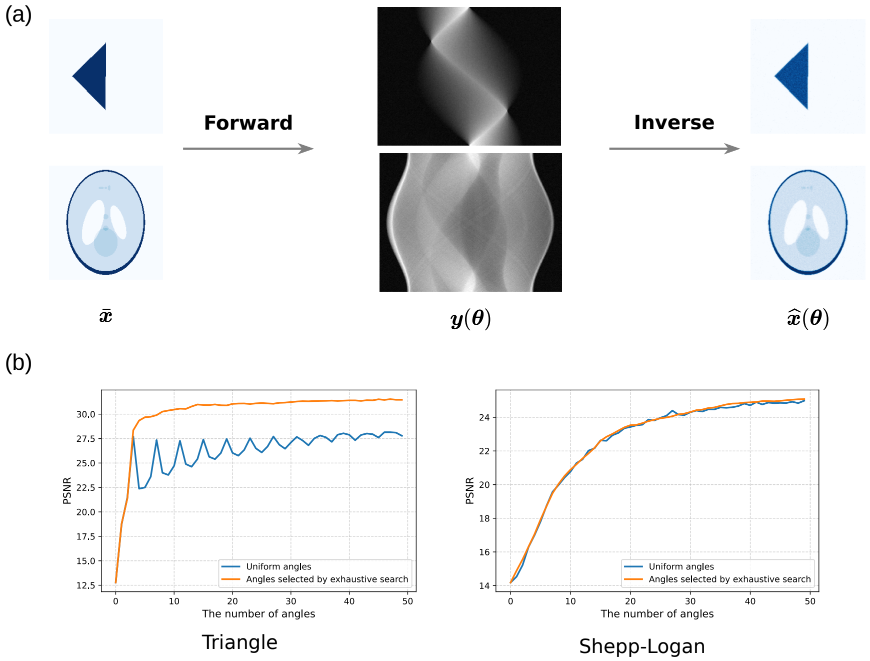


FIGURE 1. Triangle phantom example. (a) Forward process: Noisy parallel-beam projections at 180 equally spaced angles; additive zero-mean Gaussian noise (5% level) is applied to the noise-free sinogram. Inverse process: the image is reconstructed from these noisy projections. (b) PSNR as a function of the number of angles for the triangle phantom. The number of angles increases in increments of one. The orange curve represents angles selected via exhaustive search, while the blue curve corresponds to uniformly spaced angles. Left: triangle phantom; right: Shepp–Logan phantom.

projections [28, 25]. For example, the design parameter θ_k can be chosen based on the previous projections $\boldsymbol{y}(\{\theta_1, \ldots, \theta_{k-1}\})$. This approach enables the design process to adapt dynamically to changes in the underlying parameters (ground-truth image) by adjusting the design parameters iteratively.

Solving OED problems in image reconstruction is inherently a bi-level optimization challenge: the lower level reconstructs the image, while the upper level minimizes the reconstruction error relative to the ground truth. The joint optimization problem is characterized by nonconvexity, nonlinearity, and high dimensionality, particularly in imaging applications [26]. The sequential approach further complicates this optimization, as the bi-level optimization problem would need to be solved in real time. Reinforcement Learning (RL) [30], a machine learning technique designed for fast sequential decision-making, facilitates the resolution of this sOED by training through interaction with the environment. RL is grounded in the framework of Markov Decision Processes (MDPs), which consist of a state space, action

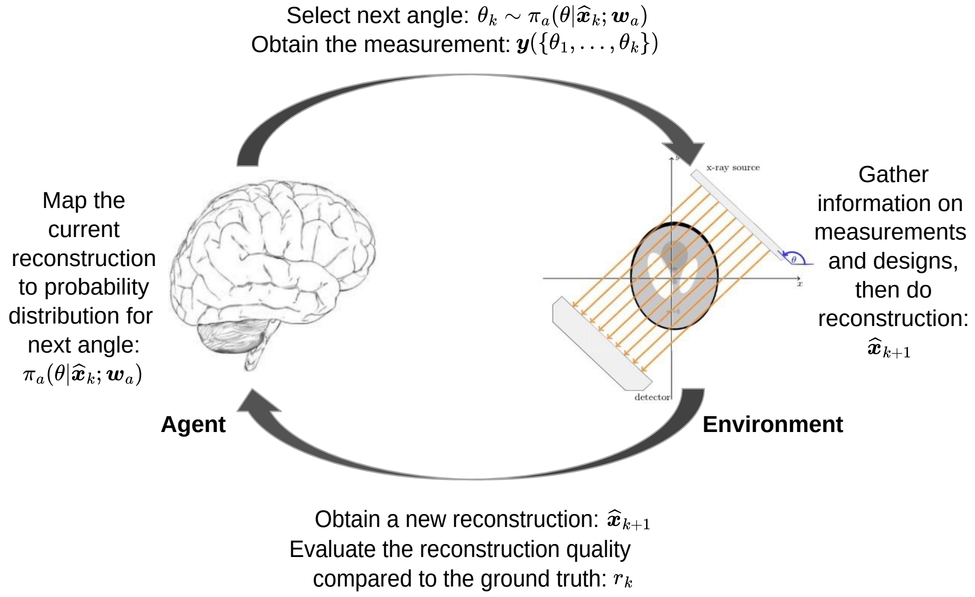

Select next angle: $\theta_k \sim \pi_a(\theta|\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$
Obtain the measurement: $\boldsymbol{y}(\{\theta_1, \ldots, \theta_k\})$
Map the current reconstruction to probability distribution for next angle: $\pi_a(\theta|\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$
x-ray source
θ
x
y
detector
Gather information on measurements and designs, then do reconstruction: $\widehat{\boldsymbol{x}}_{k+1}$
Agent
Environment
Obtain a new reconstruction: $\widehat{\boldsymbol{x}}_{k+1}$
Evaluate the reconstruction quality compared to the ground truth: r_k

FIGURE 2. Acquisition–reconstruction loop at step k. **(1)** The current reconstruction (belief state) $\widehat{\boldsymbol{x}}_k$ is fed to the policy, which either terminates or selects the next angle θ_k. **(2)** If continuing, acquire the projection $\boldsymbol{y}(\theta_k)$. **(3)** Update the reconstruction with the new data to obtain the *updated* estimate $\widehat{\boldsymbol{x}}_{k+1}$. **(4)** Compute the reward r_k from the improvement in utility (PSNR to the ground truth $\bar{\boldsymbol{x}}$). Time flows left-to-right with the transition $k \to k{+}1$; no feedback "back to k" occurs. Notation: $\bar{\boldsymbol{x}}$ denotes the ground truth used solely for computing the training utility.

space, transition model, and reward function. In practice, MDPs are sometimes extended to Partially Observable MDPs (POMDPs) when the underlying parameters are not fully observable, necessitating the reconstruction of the belief state from measurements. The goal of RL is to learn a parameterized policy that maps the current state to the next action, maximizing the expectation of the cumulative rewards [30].

Figure (2) illustrates how the RL framework integrates with the sOED process. In the context of sOED, at the k^{th} step, the action space consists of the possible values that the design parameter θ_k can take. After selecting θ_k, the reconstructed underlying parameters $\widehat{\boldsymbol{x}}_{k+1}$, inferred from all previous projections $\boldsymbol{y}(\{\theta_1, \ldots, \theta_k\})$, serve as the belief state. The utility function used for accuracy estimation acts as the reward function R. For example, R can be defined as PSNR, which estimates the reconstruction quality by comparing the reconstruction with the ground-truth image [34]. Consequently, the optimization problem of sOED is reformulated as learning a parameterized policy $\pi_{\mathrm{a}}(\theta_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$, where $\boldsymbol{w}_a$ represents the policy parameters. This policy, with the optimal parameters $\boldsymbol{w}_a^*$, maximizes the expectation of the cumulative rewards obtained from the experiments.

By employing RL, we optimize *policy parameters* rather than the design parameters directly. We consider a finite-horizon return $\sum_{k=1}^{M} r_k$. The state-value function $V^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}_1)$ is the expected cumulative reward from the initial state $\widehat{\boldsymbol{x}}_1$ under the trajectory distribution π_{chain}, which factorizes into the action policy and the data-driven transition:

$$V^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}_1) = \mathbb{E}_{\boldsymbol{\tau}\sim\pi_{\mathrm{chain}}}\left[\sum_{k=1}^{M} r_k \mid \widehat{\boldsymbol{x}}_1\right], \tag{2}$$

where the trajectory distribution is

$$\pi_{\mathrm{chain}}(\boldsymbol{\tau}; \boldsymbol{w}_a) = \prod_{k=1}^{M} \pi_{\mathrm{a}}(\theta_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a) \, \cdot \, \pi_t(\widehat{\boldsymbol{x}}_{k+1} \mid \widehat{\boldsymbol{x}}_k, \theta_k), \tag{3}$$

and π_t denotes the acquisition+reconstruction state transition (unknown analytically; we sample it by taking a projection at θ_k and running one reconstruction update).

A trajectory $\boldsymbol{\tau}$ of M steps, $\{\widehat{\boldsymbol{x}}_1, (\theta_1, \widehat{\boldsymbol{x}}_2, r_1), \ldots, (\theta_M, \widehat{\boldsymbol{x}}_{M+1}, r_M)\}$, is thus generated by iterating: choose $\theta_k \sim \pi_{\mathrm{a}}(\cdot \mid \widehat{\boldsymbol{x}}_k)$, obtain $\widehat{\boldsymbol{x}}_{k+1} \sim \pi_t(\cdot \mid \widehat{\boldsymbol{x}}_k, \theta_k)$, and accrue r_k.

The learning objective is to maximize the expected return. For a fixed initial state $\widehat{\boldsymbol{x}}_1$,

$$J(\boldsymbol{w}_a) = V^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}_1). \tag{4}$$

For policy-gradient estimation we use the action–value and advantage functions. The action–value links actions to value via

$$V^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}) = \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)\, Q^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}, \theta), \tag{5}$$

where the action–value function is formally defined as

$$Q^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}_1, \theta) = \mathbb{E}_{\boldsymbol{\tau}^{\theta}\sim\pi^{\theta}_{\mathrm{chain}}}\left[\sum_{k=1}^{M} r_k \mid \widehat{\boldsymbol{x}}_1,\, \theta_1 = \theta\right], \tag{6}$$

and $\pi^{\theta}_{\mathrm{chain}}$ denotes the trajectory distribution conditioned on taking action θ at the first time step.

The advantage measures how much better an action is than the on-policy average:

$$A^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}, \theta) = Q^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}, \theta) - V^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}}), \tag{7}$$

which we use as a variance-reducing baseline in the policy gradient.

Building on this framework, we consider two approaches to optimal stopping. The key difference is whether stopping is modeled explicitly via a terminal policy $\pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$. Introducing π_{ter} augments the trajectory distribution in Equation (3) with an additional (Bernoulli) decision at each step, which in turn alters the policy–gradient contributions for the angle policy. In the terminal–policy variant, we also compute gradients with respect to $\boldsymbol{w}_t$.

3. **Method.**

This section presents two approaches to optimal stopping. The first approach treats termination as an additional action in the action space. The second approach defines a separate termination policy, which is optimized jointly with the angle selection policy.

3.1. **Naive optimal stopping.** Optimal stopping for sOED can be implemented by introducing an additional terminal action within the action space. In our previous formulation [34], the action space consisted of the 180 discrete angles $0^\circ, \dots, 179^\circ$. In the present work, we augment this space with an explicit termination action, yielding 181 discrete actions in total. The policy $\pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_{\mathrm{a}})$ produces logits $z(\widehat{\boldsymbol{x}}) \in \mathbb{R}^{181}$, inducing a categorical distribution $h = \mathrm{softmax}(z)$ over these actions. The selected action is the index $\theta \in \{1, \dots, 181\}$, where $\theta = 1, \dots, 180$ correspond to the angles $0^\circ, \dots, 179^\circ$, and $\theta = 181$ denotes termination. The *reward function* for 'termination' and 'continuation' is defined by $R(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}}, \theta)$, which accounts for the decision to either continue or terminate, as follows:

$$R(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}}, \theta) = \begin{cases} -b, & \text{if } \theta = \theta_{max}, \\ \mathrm{PSNR}(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}}), & \text{if otherwise,} \end{cases}$$

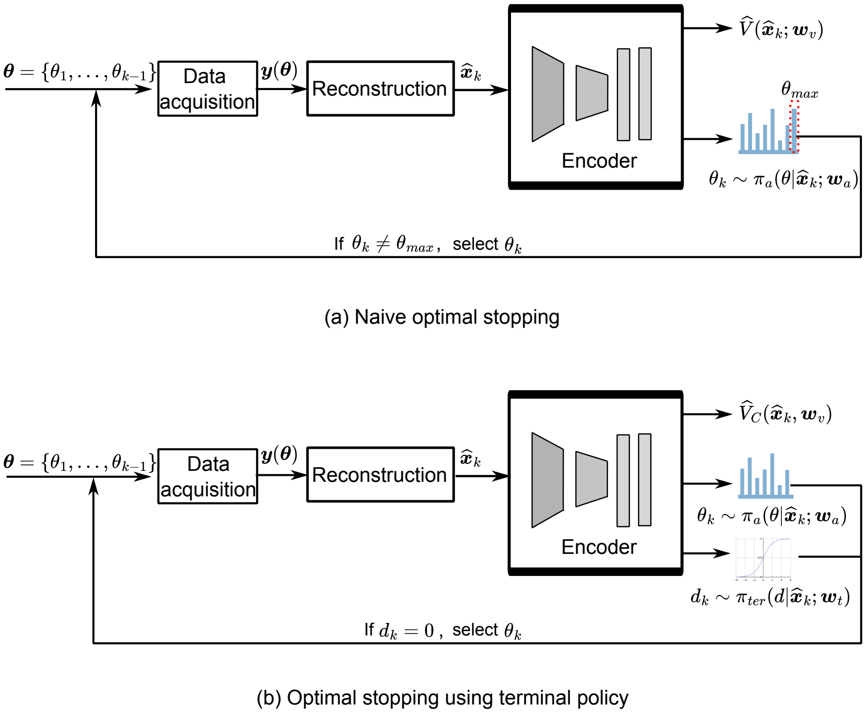
where $\mathrm{PSNR}(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}})$ is a function that serves as the immediate reward, evaluating the quality of the experiment at the stopping point $\widehat{\boldsymbol{x}}$, and $-b$ is a negative scaling factor representing the experimental cost incurred at each step. With PSNR (dB) as the utility, the per-step cost b is also in dB and serves as a minimum marginal-gain threshold (e.g., $b = 0.5$ means we continue only if PSNR increases by at least 0.5 dB); in applications b can be calibrated from time or dose per view. This mechanism works because the agent aims to maximize the cumulative reward: as long as selecting an additional angle leads to a PSNR gain that exceeds the experimental cost for continuing, the agent will proceed. It will choose to stop only when further actions are expected to result in no meaningful improvement in PSNR relative to the experimental cost. When termination is disabled (or $b = 0$), this approach reduces to the angle-only setting of [34].

As described in our previous work [34] and shown in Figure (3, a), after selecting $k-1$ angles $\boldsymbol{\theta} = \{\theta_1, ..., \theta_{k-1}\}$, the *observations* $\boldsymbol{y}(\boldsymbol{\theta})$ are obtained from the data acquisition. At step k, the reconstructed image $\widehat{\boldsymbol{x}}_k$ serves as the *belief state* and is fed to a shared encoder $E(\cdot)$; the initial state is $\widehat{\boldsymbol{x}}_1 = \mathbf{0}$. The encoder produces features $h_k = E(\widehat{\boldsymbol{x}}_k)$ that are consumed by Actor–Critic heads; no decoder is used, as images are reconstructed by the iterative solver. One branch is designed to estimate the state-value function $V^{\pi_{\mathrm{a}}}(\widehat{\boldsymbol{x}})$, and we use $\widehat{V}(\widehat{\boldsymbol{x}}; \boldsymbol{w}_v)$ for this approximation, where $\boldsymbol{w}_v$ denotes the neural network parameters. The other branch outputs the distribution over the *action space*, which consists of all possible angles and the terminal probability $\theta_{\max}$ as the final action. The inclusion of a mixture of actions for angles and termination represents the primary difference from the standard method described in our previous work [34]. Consequently, the Temporal Difference (TD) error [30] for the terminal action is modified to account for the reward at the termination state $\mathrm{PSNR}(\widehat{\boldsymbol{x}}_{k+1}, \bar{\boldsymbol{x}})$, as shown in line 10 of Algorithm (1). Algorithm (1) outlines the modified algorithm based on the previous work [34].

3.2. **Optimal stopping using terminal policy.**

3.2.1. *Objective function.* Different from the naive way of implementing optimal stopping, we consider an independent parameterized terminal policy [2], which maps the state $\widehat{\boldsymbol{x}}$ to a stochastic stopping decision $d \sim \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$. The continuation corresponds to $d = 0$, while termination corresponds to $d = 1$. Apart from the stopping decision, the reward function remains the same as in naive optimal stopping.

In the context of optimal stopping for sOED, the framework integrates both the terminal policy for scan termination and the action policy for angle selection,


$\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{k-1}\}$
Data acquisition
$\boldsymbol{y}(\boldsymbol{\theta})$
Reconstruction
$\widehat{\boldsymbol{x}}_k$
Encoder
$\widehat{V}(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$
θ_{max}
$\theta_k \sim \pi_a(\theta|\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$
If $\theta_k \neq \theta_{max}$, select θ_k
$\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_{k-1}\}$
Data acquisition
$\boldsymbol{y}(\boldsymbol{\theta})$
Reconstruction
$\widehat{\boldsymbol{x}}_k$
Encoder
$\widehat{V}_C(\widehat{\boldsymbol{x}}_k, \boldsymbol{w}_v)$
$\theta_k \sim \pi_a(\theta|\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$
$d_k \sim \pi_{ter}(d|\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t)$
If $d_k = 0$, select θ_k

(a) Naive optimal stopping

(b) Optimal stopping using terminal policy

FIGURE 3. This figure illustrates the workflows: (a) Naive stopping: after selecting angles $\{\theta_1, \ldots, \theta_{k-1}\}$ and acquiring projections $\boldsymbol{y}(\boldsymbol{\theta})$, the reconstruction $\widehat{\boldsymbol{x}}_k$ becomes the belief state. A shared encoder then outputs (i) the state-value estimate $\widehat{V}(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$ and (ii) a distribution over all angles plus a terminal action $\theta_{\max}$. (b) Terminal-policy stopping: the encoder additionally outputs a probability distribution over termination versus continuation. Another distinction is that the top branch estimates the continuation value function using $\widehat{V}_C(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$. In both workflows, a termination signal halts the process. A convolutional encoder produces features consumed by Actor–Critic heads; no decoder is used, as images are reconstructed by the iterative solver.

instead of combining them within the same action space as in the naive approach. First, we reformulate the probability chain from Equation (3) to incorporate the termination decision. The updated probability chain reflects the decision-making process of the terminal policy, which evaluates at each step whether the trajectory should terminate or continue.

Similar to Equation (3), the probability chain $\pi_{\text{chain,C}}(\boldsymbol{\tau}_C; \boldsymbol{w}_a, \boldsymbol{w}_t)$ with the inclusion of the termination distribution is expressed as:

$$\pi_{\text{chain,C}}(\boldsymbol{\tau}_C; \boldsymbol{w}_a, \boldsymbol{w}_t) = \prod_{k=1}^{T} \pi_{\text{sec}}(d_k, \widehat{\boldsymbol{x}}_{k+1}, \theta_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t), \tag{8}$$

Algorithm 1

```
Initialize the action policy parameters $\boldsymbol{w}_a$, and the value function parameters $\boldsymbol{w}_v$ randomly. Define the maximal experimental steps $M$. Set step sizes $\alpha^{\boldsymbol{w}_a} > 0$ and $\alpha^{\boldsymbol{w}_v} > 0$.
for each episode do:
    Get a phantom sample $\bar{\boldsymbol{x}}$ then a zero matrix serves as the initial state $\widehat{\boldsymbol{x}}_1$ and $k = 0$.
    while:
        Select the angle based on the softmax policy, which maps the inputs
        to a probability distribution that sums to 1: $\theta_k \sim \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$
        Get new measurements $\mathbf{y}_k$ from the data acquisition
        Reconstruct new image $\widehat{\boldsymbol{x}}_{k+1}$
        Get reward for continuation $-b$
        Estimate the state-values $\widehat{V}(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$ and $\widehat{V}(\widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_v)$
        Compute TD error:
        if $\theta_k = \theta_{\max}$:
            Get reward for termination $\mathrm{PSNR}(\widehat{\boldsymbol{x}}_{k+1}, \bar{\boldsymbol{x}})$
            $\delta_k = -b + \mathrm{PSNR}(\widehat{\boldsymbol{x}}_{k+1}, \bar{\boldsymbol{x}}) - \widehat{V}(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$
        else:
            $\delta_k = -b + \widehat{V}(\widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_v) - \widehat{V}(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$
        Update the action policy function parameters $\boldsymbol{w}_a$:
        $\boldsymbol{w}_a \leftarrow \boldsymbol{w}_a + \alpha^{\boldsymbol{w}_a} \nabla_{\boldsymbol{w}_a} \log \pi_{\mathrm{a}}(\theta_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)\delta_k$
        Update value function parameters $\boldsymbol{w}_v$:
        $\boldsymbol{w}_v \leftarrow \boldsymbol{w}_v + \alpha^{\boldsymbol{w}_v} \nabla_{\boldsymbol{w}_v} \widehat{V}(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)\delta_k$
        Increase the step number $k \mathrel{+}= 1$
        if $\theta_k = \theta_{\max}$ or $k = M$:
            break
end for
```

where the function π_{sec} is defined as:

$$\begin{aligned}&\pi_{\mathrm{sec}}(d, \widehat{\boldsymbol{x}}^{'}, \theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a, \boldsymbol{w}_t)\\ &= \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t) + \Big(1 - \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)\Big)\pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)\pi_t(\widehat{\boldsymbol{x}}^{'} \mid \widehat{\boldsymbol{x}}, \theta).\end{aligned} \tag{9}$$

The stopping step T is determined by the terminal policy, which is less or equal to the maximal step M and $d_M \equiv 1$. *Convention.* The terminal policy is Bernoulli over $d \in \{0, 1\}$, with $d = 1$ denoting termination and $d = 0$ continuation. Whenever d appears inside $\pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$ without an explicit sum over d, it refers to the event $d = 1$. Consequently, $1 - \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$ denotes the probability of continuation.

A trajectory up to the stopping point is generated from the new $\pi_{\mathrm{chain,C}}$ as:

$$\{\widehat{\boldsymbol{x}}_1, (d_1, \theta_1, \widehat{\boldsymbol{x}}_2, -b), (d_2, \theta_2, \widehat{\boldsymbol{x}}_3, -b), \ldots, (d_T, \theta_T, \widehat{\boldsymbol{x}}_{T+1}, p_T)\}.$$

This trajectory represents the sequence of states ($\widehat{\boldsymbol{x}}$), termination indicators (d), experimental cost ($-b$), and quality evaluations $\mathrm{PSNR}(\widehat{\boldsymbol{x}}_T, \bar{\boldsymbol{x}})$ (p_T) at termination. The process is designed to terminate at step T as soon as $d_T = 1$, ensuring that the trajectory is finite and explicitly concludes. For all prior steps ($i < T$), $d_i = 0$, indicating that the process continues during those steps.

The objective function, representing the value function at the fixed initial state $\widehat{\boldsymbol{x}}_1$, aims to learn both the action and terminal policies that maximize the expected cumulative rewards:

$$\begin{aligned} J(\boldsymbol{w}_a, \boldsymbol{w}_t) &= V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\ &= \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\,\mathrm{PSNR}(\widehat{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}) \\ &\quad + \Big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\Big)\, V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1). \end{aligned} \tag{10}$$

Here, the continuation state-value function is defined as follows [12, 24]:

$$V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) = -b + \mathbb{E}_{\boldsymbol{\tau}_C^{(2)} \sim \pi_{\mathrm{chain,C}}^{(2)}}\left[V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_2)\right], \tag{11}$$

where $\boldsymbol{\tau}_C^{(2)}$ represents the trajectory starting from $\widehat{\boldsymbol{x}}_2$, following the probability chain described in Equation (8).

Similarly, the continuation action-value function is defined only when the trajectory continues, as one of its inputs is the action: $Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta)$.

The optimal terminal policy $\pi^*_{\mathrm{ter}}(d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t)$ at step k is defined as [6, 24, 12]:

$$\pi^*_{\mathrm{ter}}(d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t) = \begin{cases} \mathbb{I}\Big(\mathrm{PSNR}(\widehat{\boldsymbol{x}}_k, \bar{\boldsymbol{x}}) \geq V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_k)\Big), & \text{if } k < M, \\ 1, & \text{if } k = M, \end{cases} \tag{12}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and M is the maximum number of steps in the experiment.

Figure (3, b) illustrates the workflow of optimal stopping using the terminal policy. Compared to the naive optimal stopping, this approach includes an additional branch for the terminal policy, which uses the sample from Sigmoid function to determine continuation or termination. Another distinction is that the top branch estimates the continuation value function $V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_k)$ using $\widehat{V}_C(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$.

3.2.2. *Policy gradient.* To jointly solve the optimal stopping and action selection problems for sOED, we propose a novel policy gradient method. The policy gradients are calculated using Equation (10).

For the gradient with respect to the action policy $\boldsymbol{w}_a$, the detailed unrolling recursive derivation is provided in Appendix (B). By sampling from N trajectories, the gradient is expressed as:

$$\begin{aligned} \nabla_{\boldsymbol{w}_a} J(\boldsymbol{w}_a, \boldsymbol{w}_t) &= \nabla_{\boldsymbol{w}_a} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\ &\propto \sum_{n=1}^{N}\left(\sum_{k=1}^{T} \nabla_{\boldsymbol{w}_a} \log \pi_{\mathrm{a}}(\theta_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a) Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_k, \theta)\right), \end{aligned} \tag{13}$$

To improve the stability and efficiency of action policy gradient computation, a baseline is introduced by replacing the continuation action-state value function in Equation (13) with an advantage function [30]. The continuation advantage function is defined by incorporating the output of the terminal policy; further details are provided in the Appendix (A). The continuation advantage function is approximated as:

$$\begin{aligned} A_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) \approx -b &+ \Big(1 - \pi_{\mathrm{ter}}(d' \mid \widehat{\boldsymbol{x}}'; \boldsymbol{w}_t)\Big) V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}') \\ &+ \pi_{\mathrm{ter}}(d' \mid \widehat{\boldsymbol{x}}'; \boldsymbol{w}_t)\mathrm{PSNR}(\widehat{\boldsymbol{x}}', \bar{\boldsymbol{x}}) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}). \end{aligned} \tag{14}$$

We perform stochastic gradient ascent to update the parameters of the angle policy, $\boldsymbol{w}_a$, using a step size of $\alpha^{\boldsymbol{w}_a}$.

$$\boldsymbol{w}_a \leftarrow \boldsymbol{w}_a + \alpha^{\boldsymbol{w}_a} \nabla_{\boldsymbol{w}_a} \log \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a) A_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) \tag{15}$$

For the gradient with respect to the terminal policy $\boldsymbol{w}_t$, the detailed unrolling recursive derivation is provided in the Appendix (C). By sampling from N trajectories, the gradient is expressed as:

$$
\begin{aligned}
\nabla_{\boldsymbol{w}_t} J(\boldsymbol{w}_a, \boldsymbol{w}_t) &= \nabla_{\boldsymbol{w}_t} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) \\
&\propto \sum_{n=1}^{N} \left(\sum_{k=1}^{T} \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t) \Big(\mathrm{PSNR}(\widehat{\boldsymbol{x}}_k, \bar{\boldsymbol{x}}) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_k) \Big) \right),
\end{aligned}
\tag{16}
$$

We also perform stochastic gradient ascent to update the parameters of the terminal policy, $\boldsymbol{w}_t$, using a step size of $\alpha^{\boldsymbol{w}_t}$.

$$
\boldsymbol{w}_t \leftarrow \boldsymbol{w}_t + \alpha^{\boldsymbol{w}_t} \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t) \Big(\mathrm{PSNR}(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}}) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}})) \Big)
\tag{17}
$$

The complete algorithm is presented in Algorithm (2).

Algorithm 2

```
Initialize the action policy parameters $\boldsymbol{w}_a$, the terminal policy parameters $\boldsymbol{w}_t$,
and the value function parameters $\boldsymbol{w}_v$ randomly. Define the maximal experimental steps $M$. Set step sizes $\alpha^{\boldsymbol{w}_a} > 0$, $\alpha^{\boldsymbol{w}_t} > 0$, $\alpha^{\boldsymbol{w}_v} > 0$.
for each episode do:
    Get a phantom sample $\bar{\boldsymbol{x}}$ then a zero matrix serves as the initial state $\widehat{\boldsymbol{x}}_1$
    and $k = 0$.
    while $d_k = 0$ and $k < M$:
        Select the angle based on the softmax policy, which maps the inputs
        to a probability distribution that sums to 1: $\theta_k \sim \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)$
        Get new measurements $\mathbf{y}_k$ from the data acquisition
        Reconstruct new image $\widehat{\boldsymbol{x}}_{k+1}$
        Get reward for continuation $-b$, and rewards for termination
        $\mathrm{PSNR}(\widehat{\boldsymbol{x}}_k, \bar{\boldsymbol{x}})$ and $\mathrm{PSNR}(\widehat{\boldsymbol{x}}_{k+1}, \bar{\boldsymbol{x}})$
        Determine the terminal action based on the Sigmoid policy, which
        maps the inputs to terminal probability: $d_k \sim \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t)$
        Estimate the state-values $\widehat{V}_C(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$ and $\widehat{V}_C(\widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_v)$
        Compute TD error: $\delta_k = -b + \Big(1 - \pi_{\mathrm{ter}}(d_{k+1} \mid \widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_t)\Big) \widehat{V}_C(\widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_v)$
        $+ \pi_{\mathrm{ter}}(d_{k+1} \mid \widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_t) \mathrm{PSNR}(\widehat{\boldsymbol{x}}_{k+1}, \bar{\boldsymbol{x}}) - \widehat{V}_C(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v)$
        Update policy function parameters $\boldsymbol{w}_a$:
        $\boldsymbol{w}_a \leftarrow \boldsymbol{w}_a + \alpha^{\boldsymbol{w}_a} \nabla_{\boldsymbol{w}_a} \log \pi_{\mathrm{a}}(\theta_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a) \delta_k$
        Update policy function parameters $\boldsymbol{w}_t$:
        $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_t + \alpha^{\boldsymbol{w}_t} \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t) \Big( \mathrm{PSNR}(\widehat{\boldsymbol{x}}_k, \bar{\boldsymbol{x}}) - \widehat{V}_C(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v) \Big)$
        Update value function parameters $\boldsymbol{w}_v$:
        $\boldsymbol{w}_v \leftarrow \boldsymbol{w}_v + \alpha^{\boldsymbol{w}_v} \nabla_{\boldsymbol{w}_v} \widehat{V}_C(\widehat{\boldsymbol{x}}_k; \boldsymbol{w}_v) \delta_k$
        Increase the step number $k$ += 1
end for
```

Termination is modeled either as (Algorithm 1) an additional action in a single 181-way categorical policy, or (Algorithm 2) a dedicated binary termination head alongside a 180-way angle head, both sharing the same encoder. In both cases, policy parameters are optimized jointly without alternating/block-coordinate updates. Unlike exhaustive search, which evaluates many remaining candidates at each step, the learned policy selects one angle per-step. Once trained offline, the model can be used directly online; per-step latency is effectively dominated by the reconstruction update.

4. Results.

The proposed optimal stopping method for sOED was validated through a series of X-ray CT experiments. The model was first trained using synthetic data and then evaluated on experimental X-ray CT data to test its performance and generalizability.

4.1. Dataset.

4.1.1. *Synthetic dataset.* We created a synthetic dataset that includes features with strongly nonuniform orientations, corresponding to highly informative imaging angles. The dataset includes three shapes—parallelograms, triangles, and pentagons —each varying in scale, rotation, and position. These variations result in differing requirements for the number of angles needed to achieve accurate reconstructions. Figure (4) provides example images from this dataset, while the parameters used for its generation are outlined in the Appendix (E).

4.1.2. *Experimental dataset.* All data were collected at the FleX-ray laboratory of Centrum Wiskunde en Informatica (CWI) in Amsterdam, the Netherlands [36]. A scanning approach was used to acquire projections with dimensions of 956 × 10 pixels. The source-to-object and detector-to-object distances were both set to 225 mm. An exposure time of 80 ms per-projection was applied, and the source spectrum was shaped using filters consisting of 0.1 mm zinc, 0.2 mm copper, and 0.5 mm aluminum. A total of 3601 projections were acquired. Figure (5) illustrates the scanning setup.

The dataset comprises two laser-cut objects—triangular and pentagonal —fabricated from 6 mm-thick transparent acrylate. Each shape is represented by 12 samples of varying sizes. The right-angle edges of the triangular samples range from 2.8 cm to 4.0 cm, while the edges of the pentagonal samples range from 2.5 cm to 3.0 cm. Each scanning session involved different placements of the objects, resulting in variations in rotation and translation. To create a dataset with two noise levels, two emission currents were used: 600 μA and 100 μA. The lower emission current (100 μA) produced data with higher noise levels. Finally, 12 groups of projections were acquired for each shape and each noise level, resulting in a total of 48 groups of projections.

For the purpose of this study, the number of projections was reduced to 361 by subsampling every 10th projection. To approximate the cone-beam geometry as a fan-beam geometry, only the middle row of the detector was used. The column size of the projections was then reduced to 239 by selecting every fourth pixel on the detector. Furthermore, the fan-beam data was rebinned into a parallel-beam data with 180 projections to simplify angle selection. This transformation made it more precise to identify the informative angles, as they are tangential to the edges in parallel-beam geometry.

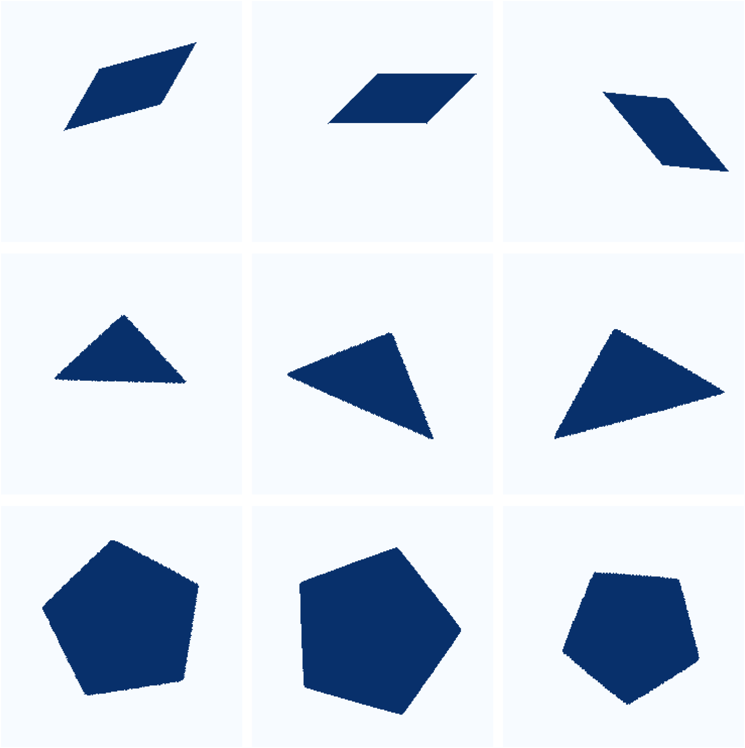


FIGURE 4. The figure shows nine samples from the synthetic dataset, including parallelograms, triangles, and pentagons. Each shape type is represented by three samples.

4.2. **Implementation.** For training on the synthetic data, the Astra Toolbox [31, 32] was used to generate the simulated projections. Reconstruction $\widehat{\boldsymbol{x}}(\boldsymbol{\theta})$ is computed with SIRT under a nonnegativity constraint (150 iterations), a stable choice for sparse-view CT. The reconstructor is held fixed throughout to isolate the effect of the design policy—angle selection and stopping—which is the focus of this study.

For the algorithm implementation, the architectures of the encoder and the Actor–Critic neural networks are detailed in the Appendix (F). During training, weights of 1.0 and 0.5 were assigned to the actor loss and critic loss in Algorithm (1) and Algorithm (2,) respectively. To encourage exploration during training, an entropy loss with a weight of 0.01 was included. Compared to Algorithm (1), Algorithm (2) included an additional loss term for the terminal policy with a weight of 1.0. These parameter settings were empirically chosen to strike an optimal balance between policy optimization, accurate value estimation, and robust exploration during training. The network weights were optimized using the Adam optimizer [20] with a learning rate of 10^{-4} and a weight decay of 10^{-5}. To prevent selecting the same angle multiple times, previously chosen angles are masked according to the procedure described in [17].

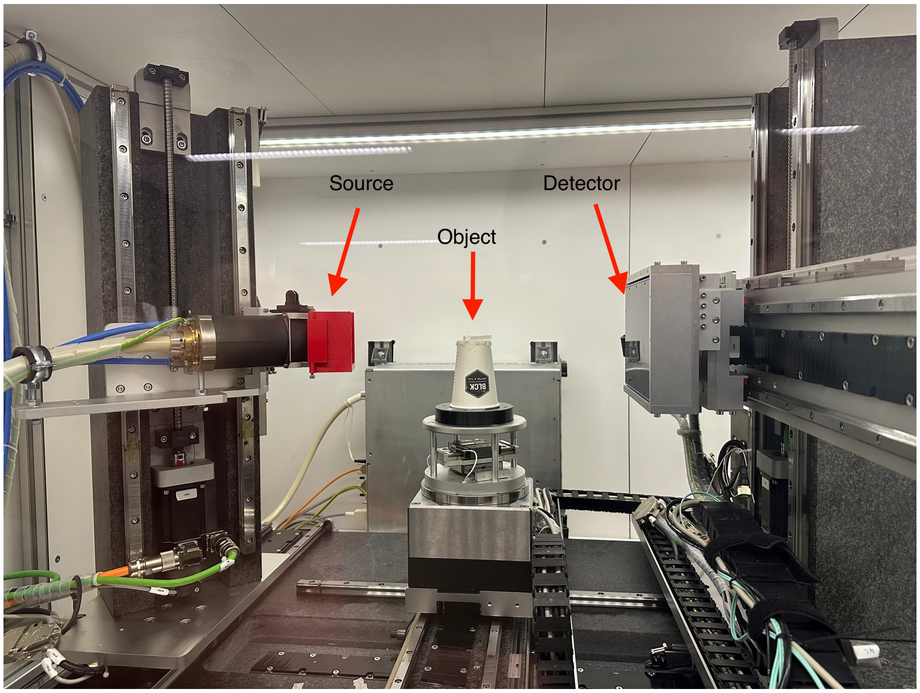


FIGURE 5. Experimental scanning setup at the FleX-ray laboratory. The X-ray source (left) and flat-panel detector (right) remain fixed, while the object is positioned on a paper cup atop the rotation stage centered between them. During acquisition, the stage rotates to capture projections from multiple angles.

4.3. **Training on synthetic data with Gaussian noise.** In this experiment, we investigated whether the two algorithms could adaptively determine the optimal number of angles for CT imaging by adjusting the experimental costs before termination. The projections incorporated 5% Gaussian noise. For Algorithms (1) and (2), the experimental costs ranged from -0.4 to -0.9 in intervals of 0.1. At higher experimental costs, priority is placed on guaranteeing image quality, whereas at lower costs the emphasis shifts toward reducing the number of projections. A separate model was trained for each experimental cost setting using the synthetic dataset shown in Figure (4). To prevent excessive nontermination during training, the maximum number of angles, M, in Algorithm (1) and Algorithm (2), was limited to 20.

During training, each episode involved sampling a data point from the dataset shown in Figure (4). The policy was trained at each angle selection step until termination, thereby completing the episode. A total of 80,000 episodes (i.e., 80,000 sampled data) were considered. Figure (6), and Figure (7)) report training dynamics on the training distribution (5% Gaussian noise) and are used to analyze learning behavior. Quantitative evaluation on held-out data appears in Section 5.4 (unseen rotations and noise levels) and Section 5.5 (experimental X-ray CT).

However, the naive stopping mechanism in Algorithm (1) (stop-as-action) did not function reliably. Across per-step cost settings $b \in [0.4, 0.9]$ (displayed in the plots as $-b$), the policy typically exhausted the maximum number of angles. As shown in Figure (6), the setting $b = 0.5$ (label "-0.5") produced shape-dependent stopping with different angle counts, whereas $b = 0.6$ (label "-0.6") failed to yield a valid stopping policy.

These outcomes indicate that, in this formulation, the policy-gradient signal to the "stop" class is too weak/sparse, leading to instability, lack of robust convergence, and a tendency to ignore the terminal action.

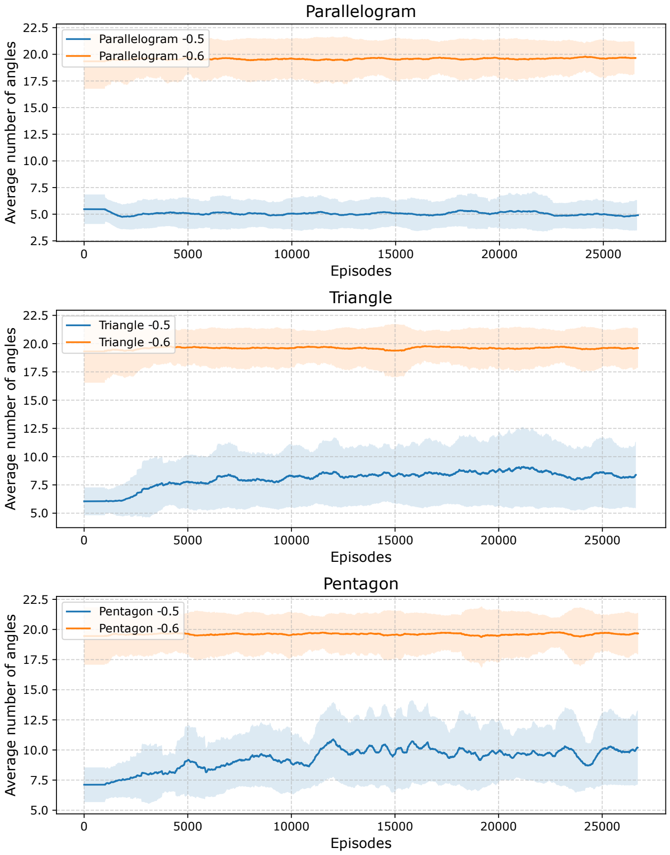


FIGURE 6. Comparison of the number of angles selected by the naive policy and by experimental cost settings of -0.5 and -0.6. The curves represent the mean number of angles during training, averaged over every 1000 episodes, and the shaded regions indicate the variance. Results are grouped by shape: parallelogram, triangle, and pentagon.

In contrast, the terminal policy remained robust under different reward settings. Figure (7) shows that, during training, the number of angles increased, and more complex shapes required more angles. Lower costs (experimental cost is -0.5) generally led to a higher number of angles.

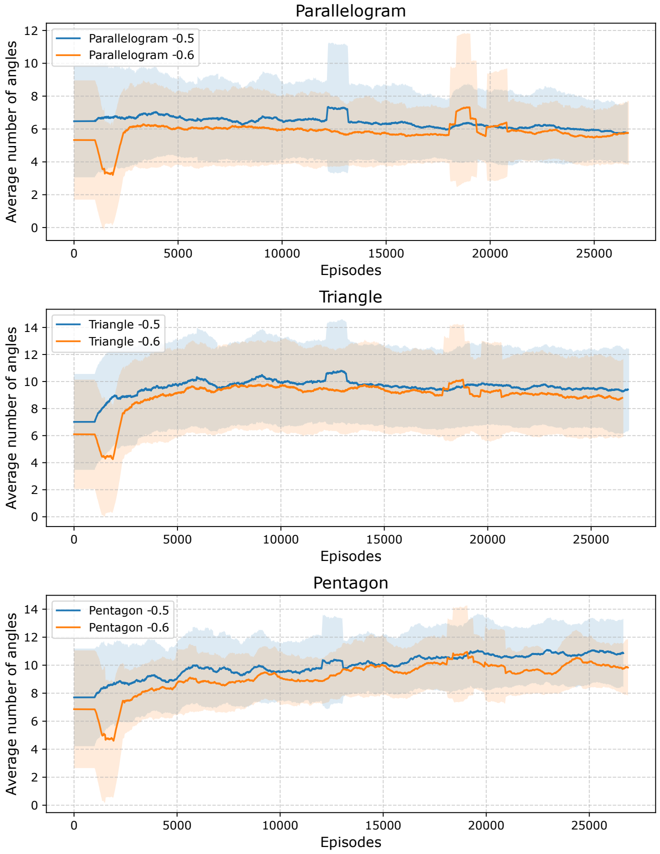


FIGURE 7. Comparison of the number of angles selected by the terminal policy and by experimental cost settings of −0.5 and −0.6. The curves represent the mean number of angles during training, averaged over every 1000 episodes, and the shaded regions indicate the variance. Results are grouped by shape: parallelogram, triangle, and pentagon.

4.4. **Validation on unseen rotations and noise levels.** To assess the generalizability of Algorithm (2) for optimal stopping, we applied the policy trained in the previous experiment (which involved 5% Gaussian noise) to synthetic data featuring unseen rotations—rotations not included in the training set shown in Figure (4)—as well as two additional noise levels: 3% and 7% Gaussian noise. The total number of unseen phantoms was 1,800, with 600 for each shape.

A standard baseline approach, the *Golden Ratio (GR) Policy* [21, 10], is considered for comparison with the RL policy. In the GR policy, an angular increment based on an irrational number is used, which leads to a nonrepeating sequence of angles that fill the angular space most evenly over time. We note that simple online stopping heuristics—e.g., stopping when the relative data residual or the relative reconstruction change falls below a threshold for L consecutive steps—require dataset-specific tuning and can either stop too early or over-acquire. Instead, we

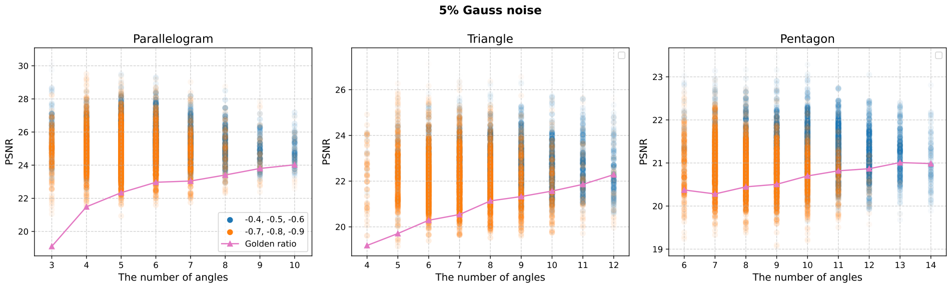


Figure 8. Simulation results are categorized into parallelogram, triangle, and pentagon shapes to illustrate how the number of angles influences PSNR under various rewards and various noise levels. Data points are color-coded by experimental costs—blue $(-0.4, -0.5, -0.6)$ and orange $(-0.7, -0.8, -0.9)$. Increased transparency indicates data points that are further from the mean. Triangular markers represent the mean PSNR values obtained from the golden ratio policy at each corresponding angle.

adopt a learned terminal policy that internalizes the quality–cost trade-off via the per-step cost b.

Figure (8) shows the validation results for experimental costs ranging from -0.4 to -0.9 (in increments of 0.1) under Gaussian noise level of 5%. Results for Gaussian noise levels of 3% and 7% are presented in Appendix (D). Each data point represents the relationship between the number of angles and the corresponding PSNR values. For clarity, results are grouped by shape (parallelogram, triangle, and pentagon). The GR policy used the same synthetic data with unseen rotations and the same number of angles chosen by the terminal policy. For each number of angles used by the GR policy, the mean PSNR value was calculated.

Overall, the RL approach with the terminal policy achieved higher PSNR values compared to the GR policy. At lower numbers of angles, the GR policy consistently produced lower mean PSNR values across all shapes, with the difference between the two methods diminishing as the number of angles increased. As shape complexity increased from parallelogram to pentagon, the performance gap between the two policies narrowed. For instance, at lower numbers of angles for the parallelogram, the RL approach yielded data points above the mean value of the GR policy, while for the more complex pentagon shape, the RL policy's advantage was less pronounced. Figure (8) uses distinct colors to represent different ranges of experimental costs, with generally lower experimental costs leading to fewer angles. As experimental cost increases, the number of angles decreases across all shapes. In Figure (8), the orange points (indicating higher costs) are concentrated at lower numbers of angles, while the blue points (indicating lower costs) are found at higher numbers of angles; however, the distinction is more evident for complex shapes, which require more angles.

Table (1) summarizes the number of angles selected for the three shapes during validation under different noise levels, with the experimental cost fixed at -0.5. In general, lower noise levels lead to fewer angles being chosen. As shown in the table, each shape requires the fewest angles at 3% Gaussian noise, whereas 7%

Table 1. Comparison of the number of angles and PSNR values for triangle and pentagon shapes under different noise levels.

Shape	Noise	Number of angles	RL (PSNR)	GR (PSNR)
Parallelogram	3%	6.02 ± 2.26	25.96 ± 1.29	22.65 ± 2.17
Parallelogram	5%	6.96 ± 2.81	25.55 ± 1.10	22.92 ± 2.06
Parallelogram	7%	9.41 ± 3.98	24.99 ± 0.95	23.21 ± 1.91
Triangle	3%	8.54 ± 1.97	23.42 ± 1.10	21.58 ± 1.71
Triangle	5%	9.43 ± 2.21	22.85 ± 1.09	21.44 ± 1.54
Triangle	7%	11.39 ± 3.09	22.05 ± 1.12	21.17 ± 1.41
Pentagon	3%	10.43 ± 1.98	22.66 ± 0.67	22.21 ± 0.51
Pentagon	5%	10.99 ± 2.01	21.47 ± 0.56	20.99 ± 0.51
Pentagon	7%	12.56 ± 2.25	20.01 ± 0.58	19.56 ± 0.52

Gaussian noise leads to the largest number of angles. Furthermore, the number of angles increases with the complexity of the geometry: the parallelogram requires the fewest angles, while the pentagon requires the most. The RL policy outperforms the GR policy under every tested condition.

4.5. **Test on experimental X-ray CT data with two noise levels.** We explored the gap between simulation and the real world by applying the model trained on the synthetic dataset to the experimental X-ray CT data described in Section 5.1.2. Building on the potential demonstrated in the simulation, we further investigated whether the trained model could adapt to changes in the real scanning environment, particularly variations in noise levels. Two noise levels were considered, based on different emission current settings (600 μA and 100 μA). The model was trained using 5% Gaussian noise on the projections, while the noise in the experimental X-ray CT typically consists of a mixture of Poisson and Gaussian noise [1]. It is important to demonstrate whether training under simplified conditions (Gaussian noise model) can produce reasonable results when applied to the more complex conditions of the real world (experimental noise).

Figure (9) shows results obtained from experimental X-ray CT data using three trained models under varying emission currents and reward settings. To evaluate reconstruction quality, we used the reconstruction obtained with all 180 angles as the ground truth. The GR policy used the same number of angles chosen by the terminal policy. For each number of angles used by the GR policy, the mean PSNR value was calculated. Several consistent trends emerged across the 12 data groups, aligning with the conclusions from the validation. First, pentagon data points generally involved a larger number of selected angles than triangles. Second, noisier data from the lower emission current (100 μA) triggered more angles than data collected at the higher emission current (600 μA). Third, although the number of samples was limited, it was still evident that lower experimental costs (e.g., -0.4, -0.5 and -0.6) led to more angles, while higher costs (e.g., -0.7, -0.8 and -0.9) required fewer angles. Finally, the RL policy clearly outperformed the GR policy when fewer angles were selected, though the difference between them diminished as the number of angles increased.

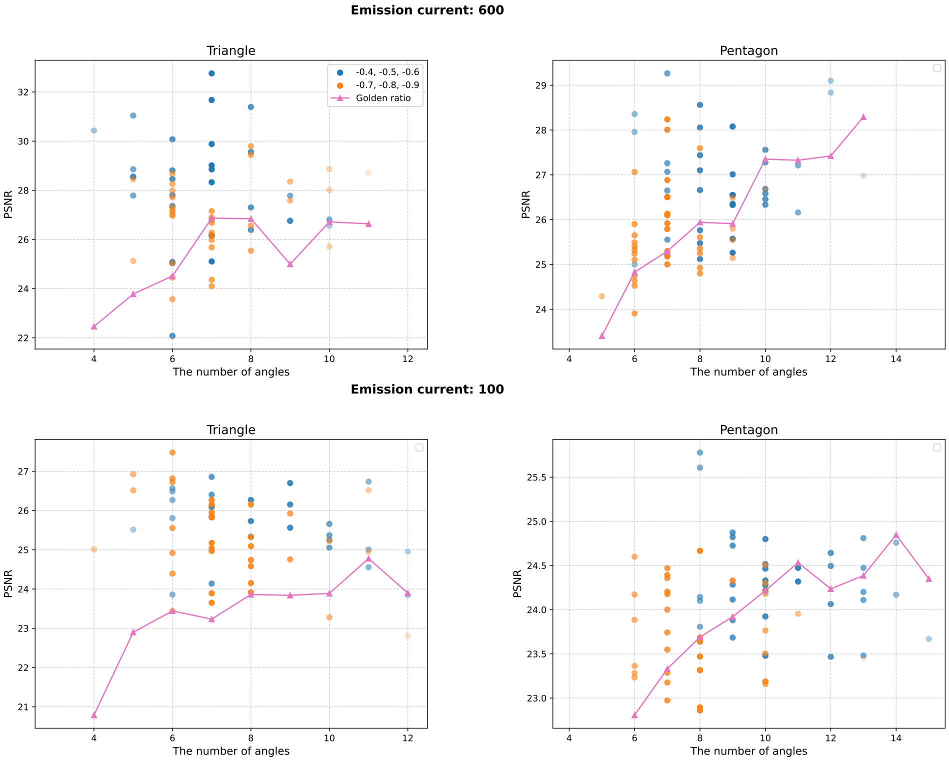


FIGURE 9. Results on experimental X-ray CT data for the triangle and pentagon shapes are shown, illustrating how the number of angles affects PSNR under various experimental costs and noise levels. Data points are color-coded by their experimental costs: blue $(-0.4, -0.5, -0.6)$ and orange $(-0.7, -0.8, -0.9)$. Triangular markers represent the mean PSNR values obtained from the GR policy at each corresponding number of angles.

Table (2) compares the performance of the RL policy and GR policy under an emission current of 600 μA and 100 μA under a reward setting of -0.5. The RL policy consistently outperformed the GR policy for triangles, while it was only comparable to the baselines for pentagons. This outcome, likely influenced by the gap between synthetic and experimental X-ray CT data, is consistent with the training results, where the RL policy did not demonstrate a particularly distinct advantage for pentagon shapes.

Additionally, Figure (10) illustrates sample reconstructions for the two policies, with the number of angles determined by the RL policy. Each colored spoke indicates the *beam direction* of a parallel-beam view at angle θ (degrees), measured counter-clockwise from the x-axis; note that θ and $\theta + 180°$ are equivalent. The results show that the angles selected by the RL policy tend to cluster around the edges. Furthermore, as the noise level increases (from an emission current of 600 μA to 100 μA), the number of selected angles increases, and their distribution becomes broader.

Table 2. Comparison of the number of selected angles and PSNR values for triangle and pentagon shapes under two different noise levels: The low noise level corresponds to an emission current of 600 μA, and the high noise level to 100 μA.

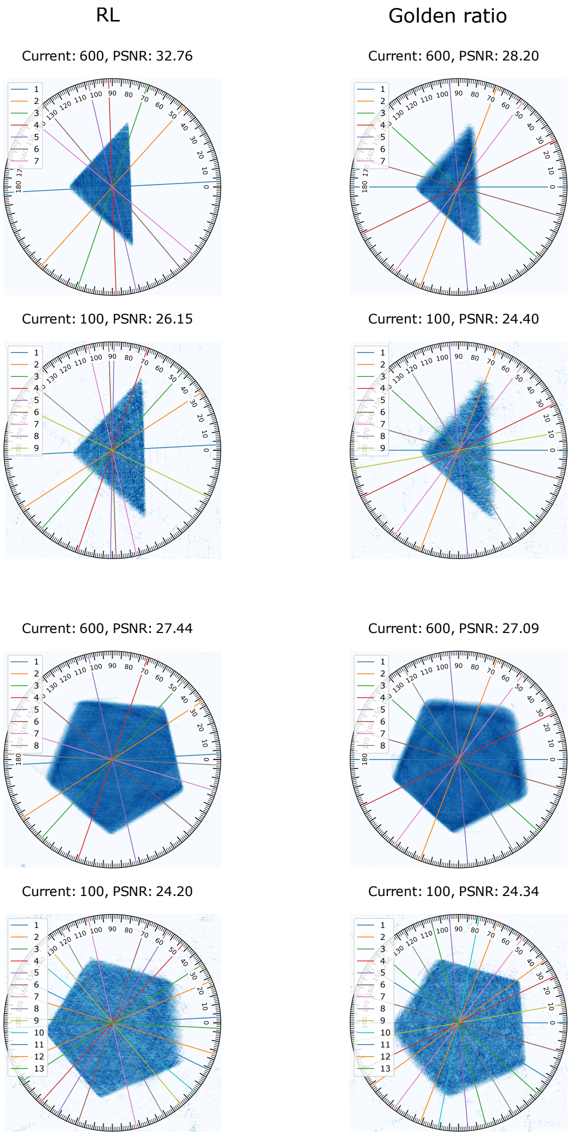
Shape	Noise	Number of angles	RL (PSNR)	GR (PSNR)
Triangle	Low	7.92 ± 3.04	27.38 ± 2.09	25.58 ± 2.10
Triangle	High	11.08 ± 2.87	25.95 ± 0.72	24.47 ± 1.32
Pentagon	Low	9.17 ± 1.86	26.63 ± 1.13	26.84 ± 0.87
Pentagon	High	10.33 ± 2.05	24.50 ± 0.63	24.40 ± 0.30

Consequently, the RL policy trained on synthetic data with a simple Gaussian noise model demonstrated reasonable performance on experimental X-ray CT data, effectively handling both optimal stopping and the selection of informative angles.

5. **Discussion.** These results demonstrate that our algorithm effectively balances experimental costs and reconstruction quality by incorporating optimal stopping and adaptively selecting informative angles through deep reinforcement learning. Moreover, the model trained on a synthetic dataset with a simple Gaussian noise model showed satisfactory performance on experimental X-ray CT data, which typically includes more complex noise, such as a mixture of Gaussian and Poisson noise. The algorithm also demonstrated its ability to adapt the number of angles (optimal stopping) to varying noise levels in real-world scenarios while maintaining a consistent level of experimental quality.

Despite these encouraging findings, several areas for improvement remain. First, the gap between simulation and real-world performance could be minimized by incorporating more realistic simulators that account for precise noise models. Second, the training process could be improved by adopting a multi-stage approach, where the model is initially trained on simulated data and subsequently fine-tuned using experimental X-ray CT data. Third, extending the dataset to three-dimensional scenarios would enable the exploration of larger action and state spaces, further enhancing the model's capability. Additionally, task-specific zooming techniques could be integrated for applications such as defect detection. Fourth, incorporating learning-based reconstruction algorithms could facilitate the extraction of informative features directly from the reconstruction process, providing a more robust basis for selecting informative angles. Fifth, while we report PSNR as the primary, model-aligned metric in this work, complementary structural and task-oriented measures (e.g., SSIM, edge preservation, and detectability) are also informative; related results with SSIM and detectability are reported in [35], and a comprehensive multi-metric evaluation is a natural direction for future work. Finally, the learned policy is trained on the shape families used in this work and generalizes within that distribution (unseen rotations/scales); we do not claim out-of-distribution generalization to arbitrary non-convex or non-homogeneous objects. In industrial CT, CAD/blueprint priors make class-specific training natural; extending to broader multi-material classes is left for future work.

6. **Conclusion.** In this paper, we proposed an approach to simultaneously optimize adaptive informative angle selection and optimal stopping by introducing a


RL

Golden ratio

Current: 600, PSNR: 32.76

Current: 600, PSNR: 28.20

Current: 100, PSNR: 26.15

Current: 100, PSNR: 24.40

Current: 600, PSNR: 27.44

Current: 600, PSNR: 27.09

Current: 100, PSNR: 24.20

Current: 100, PSNR: 24.34

FIGURE 10. Selected projection directions for a representative scan. Each colored ray denotes the projection *normal* (central ray) of a parallel-beam view at angle θ (degrees), measured counterclockwise from the x-axis and defined modulo 180°. Color encodes acquisition order (step $1 \to M$; see colorbar).

terminal policy and jointly computing the policy gradient for both the angle selection and terminal policies. Additionally, we investigated the gap between simulation and real-world scenarios. Our findings demonstrate the feasibility of achieving optimal stopping for sOED based on experimental costs. Furthermore, the trained model from simulation showed promising potential for application to experimental X-ray CT data, highlighting its value for industrial CT applications. This approach paves the way for fully adaptive scanning processes, optimizing both the selection of informative angles and the number of angles required.

7. **Data and code availability.** Experimental datasets are available on Zenodo: https://zenodo.org/records/14893740 The code for this work is available on GitHub: https://github.com/tianyuan1wang/Optimal_Stopping_RL_CT

Appendix A. Continuation advantage function. The following equations show the relationships between the continuation state-value function and the continuation action-value function (cf. Eq. (2) and Eq. (5)). Here, $\pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)$ is the angle policy, $\pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$ is the terminal policy with decision $d \in \{\mathrm{stop}, \mathrm{continue}\}$, and $\pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)$ denotes the (unknown) *state transition* induced by acquiring angle θ and applying one reconstruction update. *Convention.* Whenever d (resp. d') appears inside $\pi_{\mathrm{ter}}(d \mid \cdot)$ (resp. $\pi_{\mathrm{ter}}(d' \mid \cdot)$) without an explicit sum over the decision variable, it denotes the termination event; consequently $1 - \pi_{\mathrm{ter}}(\cdot)$ denotes continuation.

$$\begin{aligned} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) = {} & \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)\mathrm{PSNR}(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}}) \\ & + \Big(1 - \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)\Big) V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}), \end{aligned} \tag{18}$$

$$V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) = \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta), \tag{19}$$

$$Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) = -b + \sum_{\widehat{\boldsymbol{x}}'} \pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)\, V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}'). \tag{20}$$

Continuation advantage. It evaluates whether the selected action is better than the on-policy average at the current state:

$$\begin{aligned} A_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) &= Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) - \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) \quad \text{(by Eq. (19))} \\ &= -b + \sum_{\widehat{\boldsymbol{x}}'} \pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)\, V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}') - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) \quad \text{(by Eq. (20))} \\ &= -b + \sum_{\widehat{\boldsymbol{x}}'} \pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)\, \Big(\big(1 - \pi_{\mathrm{ter}}(d' \mid \widehat{\boldsymbol{x}}'; \boldsymbol{w}_t)\big)\, V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}') \\ &\quad + \pi_{\mathrm{ter}}(d' \mid \widehat{\boldsymbol{x}}'; \boldsymbol{w}_t)\, \mathrm{PSNR}(\widehat{\boldsymbol{x}}', \bar{\boldsymbol{x}}) \Big) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}). \quad \text{(by Eq. (18))} \end{aligned} \tag{21}$$

In practice, since $\pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)$ is unknown, we sample $\widehat{\boldsymbol{x}}' \sim \pi_t(\cdot \mid \widehat{\boldsymbol{x}}, \theta)$ (one Monte Carlo sample) and use

$$\begin{aligned} A_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta) \approx {} & -b + \big(1 - \pi_{\mathrm{ter}}(d' \mid \widehat{\boldsymbol{x}}'; \boldsymbol{w}_t)\big)\, V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}') \\ & + \pi_{\mathrm{ter}}(d' \mid \widehat{\boldsymbol{x}}'; \boldsymbol{w}_t)\, \mathrm{PSNR}(\widehat{\boldsymbol{x}}', \bar{\boldsymbol{x}}) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}). \end{aligned}$$

Appendix B. **Policy gradient on $\boldsymbol{w}_a$.** *Notation.* $\pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)$ is the angle policy; $\pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$ is the terminal policy with $d \in \{\text{stop}, \text{continue}\}$; $\pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)$ denotes the (unknown) *state transition* induced by acquiring θ and applying one reconstruction update.

$$
\begin{aligned}
&\nabla_{\boldsymbol{w}_a} J(\boldsymbol{w}_a, \boldsymbol{w}_t) \\
&= \nabla_{\boldsymbol{w}_a} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\
&= \nabla_{\boldsymbol{w}_a} \Big(\pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\, \mathrm{PSNR}(\widehat{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}) \\
&\quad + \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big)\, V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \Big) \quad \text{(by Eq. (18))} \\
&= \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big)\, \nabla_{\boldsymbol{w}_a} V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) (\text{PSNR and } \pi_{\mathrm{ter}} \text{ do not depend on } \boldsymbol{w}_a) \\
&= \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big) \Big(\sum_{\theta} \nabla_{\boldsymbol{w}_a} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_a)\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1, \theta) \\
&\quad + \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_a)\, \nabla_{\boldsymbol{w}_a} Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1, \theta) \Big) \quad \text{(by Eq. (19))} \\
&= \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big) \Big(\sum_{\theta} \nabla_{\boldsymbol{w}_a} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_a)\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1, \theta) \\
&\quad + \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_a) \sum_{\widehat{\boldsymbol{x}}} \pi_t(\widehat{\boldsymbol{x}} \mid \widehat{\boldsymbol{x}}_1, \theta)\, \nabla_{\boldsymbol{w}_a} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) \Big) \quad \text{(by Eq. (20))}.
\end{aligned}
\tag{22}
$$

After unrolling the recursion,

$$
\begin{aligned}
&\nabla_{\boldsymbol{w}_a} J(\boldsymbol{w}_a, \boldsymbol{w}_t) \\
&= \nabla_{\boldsymbol{w}_a} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\
&= \sum_{k=1}^{T} \sum_{\theta, \widehat{\boldsymbol{x}}} \pi_1^{(k)}(\theta, \widehat{\boldsymbol{x}}, d_{k+1} \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t) \sum_{\theta'} \nabla_{\boldsymbol{w}_a} \pi_{\mathrm{a}}(\theta' \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta') \\
&= \sum_{k=1}^{T} \sum_{\theta, \widehat{\boldsymbol{x}}} \pi_1^{(k)}(\theta, \widehat{\boldsymbol{x}}, d_{k+1} \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t) \sum_{\theta'} \pi_{\mathrm{a}}(\theta' \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)\, \frac{\nabla_{\boldsymbol{w}_a} \pi_{\mathrm{a}}(\theta' \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)}{\pi_{\mathrm{a}}(\theta' \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)}\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta') \\
&= \sum_{k=1}^{T} \sum_{\theta, \widehat{\boldsymbol{x}}} \pi_1^{(k)}(\theta, \widehat{\boldsymbol{x}}, d_{k+1} \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t) \sum_{\theta'} \pi_{\mathrm{a}}(\theta' \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)\, \nabla_{\boldsymbol{w}_a} \log \pi_{\mathrm{a}}(\theta' \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a)\, Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}, \theta'),
\end{aligned}
\tag{23}
$$

where

$$
\begin{aligned}
&\pi_1^{(k)}(\theta, \widehat{\boldsymbol{x}}, d_{k+1} \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t) \\
&\quad = \begin{cases} \displaystyle\sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a) \sum_{\widehat{\boldsymbol{x}}} \pi_t(\widehat{\boldsymbol{x}} \mid \widehat{\boldsymbol{x}}_k, \theta)\, \big(1 - \pi_{\mathrm{ter}}(d_{k+1} \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)\big), & \text{if } k \neq 0, \\ \big(1 - \pi_{\mathrm{ter}}(d_{k+1} \mid \widehat{\boldsymbol{x}}_{k+1}; \boldsymbol{w}_t)\big), & \text{if } k = 0. \end{cases}
\end{aligned}
$$

Appendix C. **Policy gradient on $\boldsymbol{w}_t$.** *Notation.* $\pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_a)$ is the angle policy; $\pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t)$ is the terminal policy with $d \in \{\text{stop}, \text{continue}\}$; $\pi_t(\widehat{\boldsymbol{x}}' \mid \widehat{\boldsymbol{x}}, \theta)$ denotes the (unknown) *state transition* induced by acquiring θ and applying one reconstruction update.

$$
\begin{aligned}
&\nabla_{\boldsymbol{w}_t} J(\boldsymbol{w}_a, \boldsymbol{w}_t) \\
&= \nabla_{\boldsymbol{w}_t} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\
&= \nabla_{\boldsymbol{w}_t} \Big(\pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\, \mathrm{PSNR}(\widehat{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}) \\
&\quad + \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big)\, V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \Big) \quad \text{(by Eq. (18))} \\
&= \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\, \mathrm{PSNR}(\widehat{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}) - \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\, V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\
&\quad + \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big)\, \nabla_{\boldsymbol{w}_t} V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \\
&= \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t) \Big(\mathrm{PSNR}(\widehat{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \Big) \\
&\quad + \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big) \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_a)\, \nabla_{\boldsymbol{w}_t} Q_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1, \theta) \quad \text{(by Eq. (19))} \\
&= \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t) \Big(\mathrm{PSNR}(\widehat{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}) - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}_1) \Big) \\
&\quad + \big(1 - \pi_{\mathrm{ter}}(d_1 \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_t)\big) \sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_1; \boldsymbol{w}_a) \sum_{\widehat{\boldsymbol{x}}} \pi_t(\widehat{\boldsymbol{x}} \mid \widehat{\boldsymbol{x}}_1, \theta)\, \nabla_{\boldsymbol{w}_t} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) \\
&\quad \text{(by Eq. (20))}.
\end{aligned}
\tag{24}
$$

After unrolling the recursion,

$$
\begin{aligned}
&\nabla_{\boldsymbol{w}_t} J(\boldsymbol{w}_a, \boldsymbol{w}_t) \\
&= \nabla_{\boldsymbol{w}_t} V^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) \\
&= \sum_{k=1}^{T} \sum_{\theta, \widehat{\boldsymbol{x}}} \pi_2^{(k)}(\theta, \widehat{\boldsymbol{x}}, d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t)\, \nabla_{\boldsymbol{w}_t} \pi_{\mathrm{ter}}(d \mid \widehat{\boldsymbol{x}}; \boldsymbol{w}_t) \Big(\mathrm{PSNR}(\widehat{\boldsymbol{x}}, \bar{\boldsymbol{x}}) \\
&\quad - V_C^{\pi_{\mathrm{a}},\pi_{\mathrm{ter}}}(\widehat{\boldsymbol{x}}) \Big),
\end{aligned}
\tag{25}
$$

where

$$
\begin{aligned}
&\pi_2^{(k)}(\theta, \widehat{\boldsymbol{x}}, d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a, \boldsymbol{w}_t) \\
&= \begin{cases} \big(1 - \pi_{\mathrm{ter}}(d_k \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_t)\big) \displaystyle\sum_{\theta} \pi_{\mathrm{a}}(\theta \mid \widehat{\boldsymbol{x}}_k; \boldsymbol{w}_a) \sum_{\widehat{\boldsymbol{x}}} \pi_t(\widehat{\boldsymbol{x}} \mid \widehat{\boldsymbol{x}}_k, \theta), & \text{if } k \neq 0, \\ 1, & \text{if } k = 0. \end{cases}
\end{aligned}
$$

Appendix D. **Additional results.** We report supplementary experiments to illustrate robustness across seeds and noise levels (3%, 5%, 7%). Architectures and hyperparameters match Section 5.2.

Table 3. Parameters used in the datasets. The scale is represented by the radius of the circumscribed circles, and the position is described by the center coordinates (x, y) of these circles.

Shapes	**Scales**	**Shifts**
Parallelogram	Radius: $42 \sim 51$	Center (x, y): $(110{,}130) \sim (110{,}130)$
Triangles	Radius: $56 \sim 89$	Center (x, y): $(110{,}130) \sim (110{,}130)$
Pentagons	Radius: $56 \sim 89$	Center (x, y): $(110{,}130) \sim (110{,}130)$

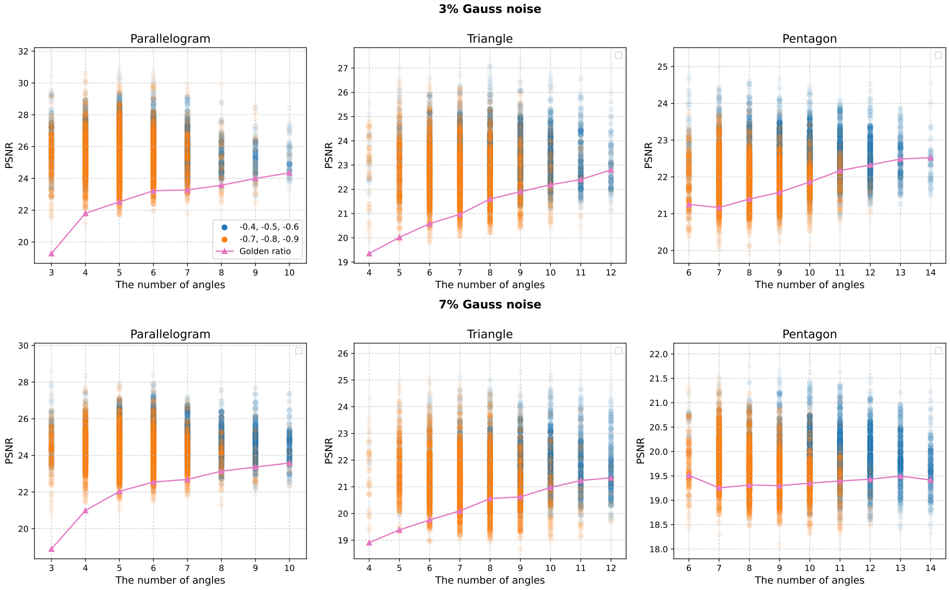


Figure 11. Simulation results are categorized into parallelogram, triangle, and pentagon shapes to illustrate how the number of angles influences PSNR under various rewards and various noise levels. Data points are color-coded by experimental costs—blue $(-0.4, -0.5, -0.6)$ and orange $(-0.7, -0.8, -0.9)$. Increased transparency indicates data points that are further from the mean. Triangular markers represent the mean PSNR values obtained from the golden ratio policy at each corresponding angle.

The trends mirror those in Figure (8). Figure (11) encodes ranges of experimental cost b by color: lower costs (blue) encourage acquiring more views, whereas higher costs (orange) penalize additional views and thus lead to fewer selected angles. Across all shapes, the number of acquired angles tends to decrease as b increases, with the separation between cost regimes more pronounced for geometrically complex phantoms that require more views at a given cost level. Moreover, higher noise levels generally drive the policy to select more angles.

Appendix E. Parameters used in datasets.

Appendix F. Architecture of the Actor–Critic network. The Actor–Critic model consists of a shared convolutional feature extractor followed by three separate heads for policy, value estimation, and terminal state prediction. The complete architecture is detailed in Table (4).

TABLE 4. Detailed architecture of the Actor–Critic neural network. A shared convolutional encoder extracts features, which are processed by separate heads for policy (actor), value estimation (critic), and terminal prediction.

Layer	**Type**	**Parameters**	**In Ch.**	**Out Ch.**	**Output Size**
1	Conv2d	kernel=3, stride=2, padding=1	1	12	120 × 120
2	GroupNorm	num_groups=4	12	12	same
3	LeakyReLU	negative_slope=0.2	12	12	same
4	MaxPool2d	kernel=2	12	12	60 × 60
5	Conv2d	kernel=3, padding=1	12	24	same
6	GroupNorm	num_groups=4	24	24	same
7	LeakyReLU	negative_slope=0.2	24	24	same
8	MaxPool2d	kernel=2	24	24	30 × 30
9	Conv2d	kernel=3, padding=1	24	48	same
10	GroupNorm	num_groups=4	48	48	same
11	LeakyReLU	negative_slope=0.2	48	48	same
12	MaxPool2d	kernel=4	48	48	7 × 7
13	Flatten	-	-	-	1 × 2352
Actor Head					
14	Linear	in=1 × 2352, out=1 × 180	-	-	1 × 180
15	Softmax	dim=-1	-	-	1 × 180
Critic Head					
16	Linear	in=1 × 2352, out=1 × 2352	-	-	1 × 2352
17	ReLU	-	-	-	same
18	Linear	in=1 × 2352, out=1 × 1	-	-	1 × 1
Terminal Head					
19	Linear	in=1 × 2352, out=1 × 1	-	-	1 × 1
20	Sigmoid	-	-	-	1 × 1

REFERENCES

[1] V. Andriiashen, R. van Liere, T. van Leeuwen, and K. J. Batenburg, X-ray Image Generation as a Method of Performance Prediction for Real-Time Inspection: A Case Study, *Journal of Nondestructive Evaluation*, **43** (2024), 79.

[2] P.-L. Bacon, J. Harb and D. Precup, The option-critic architecture, *Proceedings of the AAAI Conference on Artificial Intelligence*, **31** (2017), 1726-1734.

[3] K. J. Batenburg, W. Fortes, L. Hajdu and R. Tijdeman, Bounds on the difference between reconstructions in binary tomography, *In: International Conference on Discrete Geometry for Computer Imagery*, **6607** (2011), 369-380.

[4] K. J. Batenburg, W. J. Palenstijn, P. Balázs and J. Sijbers, Dynamic angle selection in binary tomography, *Computer Vision and Image Understanding*, **117** (2013), 306-318.

[5] R. Barbano, J. Leuschner, J. Antorán, B. Jin and J. M. Hernández-Lobato, Bayesian experimental design for computed tomography with the linearised deep image prior, preprint, `arXiv:2207.05714`, 2022.

[6] S. Becker, P. Cheridito and A. Jentzen, Deep optimal stopping, *Journal of Machine Learning Research*, **20** (2019), 1-25.

[7] T. Blau, E. V. Bonilla, I. Chades and A. Dezfouli, Optimizing sequential experimental design with deep reinforcement learning, In: *International Conference on Machine Learning (ICML)*, (2022), 2107-2128.

[8] M. Burger, A. Hauptmann, T. Helin, N. Hyvönen and J.-P. Puska, Sequentially optimized projections in X-ray imaging, *Inverse Problems*, **37** (2021), 075006.

[9] K. Chaloner and I. Verdinelli, Bayesian experimental design: A review, *Statistical Science*, **10** (1995), 273-304.

[10] T. M. Craig, A. A. Kadu, K. J. Batenburg and S. Bals, Real-time tilt undersampling optimization during electron tomography of beam sensitive samples using golden ratio scanning and recast3D, *Nanoscale*, **15** (2023), 5391-5402.

[11] A. Dabravolski, K. J. Batenburg and J. Sijbers, Dynamic angle selection in X-ray computed tomography, N*uclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, **324** (2014), 17-24.

[12] N. Damera Venkata and C. Bhattacharyya, Deep recurrent optimal stopping, *Advances in Neural Information Processing Systems*, (2024), 36.

[13] N. Elata, T. Michaeli, and M. Elad, Adaptive compressed sensing with diffusion-based posterior sampling, In: *European Conference on Computer Vision* (ECCV), (2025), 290-308, 2025.

[14] A. Foster, D. R. Ivanova, I. Malik and T. Rainforth, Deep adaptive design: Amortizing sequential Bayesian experimental design, In: *International Conference on Machine Learning (ICML)*, (2021), 3384-3395.

[15] P. C. Hansen, J. Jørgensen and W. R. B. Lionheart, *Computed Tomography: Algorithms, Insight, and Just Enough Theory*, SIAM, 2021.

[16] T. Helin, N. Hyvönen and J.-P. Puska, Edge-promoting adaptive Bayesian experimental design for X-ray imaging, *SIAM Journal on Scientific Computing*, **44** (2022), B506-B530.

[17] S. Huang and S. Ontañón, A closer look at invalid action masking in policy gradient algorithms, preprint, `arXiv:2006.14171`, 2020.

[18] X. Huan, J. Jagalur and Y. Marzouk, Optimal experimental design: Formulations and computations, *Acta Numerica*, **33** (2024), 715-840.

[19] I. G. Kazantsev, Information content of projections, *Inverse Problems*, **7** (1991), 887-898.

[20] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, preprint, `arXiv:1412.6980`, 2014.

[21] T. Kohler, A projection access scheme for iterative reconstruction based on the golden section, *IEEE Nuclear Science Symposium Conference Record*, **6** (2004), 3961-3965.

[22] D. V. Lindley, *Bayesian Statistics: A Review*, SIAM, 1972.

[23] J. L. Mueller and S. Siltanen, Linear and nonlinear inverse problems with practical applications, Society for Industrial and Applied Mathematics, 2012.

[24] G. Peskir and A. Shiryaev, *Optimal Stopping and Free-Boundary Problems*, Springer, 2006.

[25] T. Rainforth, A. Foster, D. R. Ivanova and F. Bickford Smith, Modern Bayesian experimental design, *Statistical Science*, **39** (2024), 100-114.

[26] L. Ruthotto, J. Chung and M. Chung, Optimal experimental design for inverse problems with state constraints, *SIAM Journal on Scientific Computing*, **40** (2018), B1080-B1100.

[27] E. G. Ryan, C. C. Drovandi, J. M. McGree, and A. N. Pettitt, Fully Bayesian optimal experimental design: A review, *International Statistical Review*, **84** (2016), 128-154.

[28] W. Shen and X. Huan, Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning, *Computer Methods in Applied Mechanics and Engineering*, **416** (2023), 116304.

[29] Z. Shen, Y. Wang, D. Wu, X. Yang and B. Dong, Learning to scan: A deep reinforcement learning approach for personalized scanning in CT imaging, *Inverse Problems and Imaging*, **16** (2022), 179-195.

[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

[31] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg and J. Sijbers, Fast and flexible X-ray tomography using the ASTRA toolbox, *Optics Express*, **24** (2016), 25129-25147.

[32] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg and J. Sijbers, The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography, *Ultramicroscopy*, **157** (2015), 35-47.

[33] L. Varga, P. Balázs and A. Nagy, Projection selection dependency in binary tomography, *Acta Cybernetica*, **20** (2011), 167-187.

[34] T. Wang, F. Lucka and T. van Leeuwen, Sequential experimental design for X-ray CT Using Deep Reinforcement Learning, *IEEE Transactions on Computational Imaging*, **10** (2024), 953-968.

[35] T. Wang, V. Florian, R. Schielein, C. Kretzer, S. Kasperl, F. Lucka, and T. van Leeuwen, Task-adaptive angle selection for computed tomography-based defect detection, *Journal of Imaging*, **10** (2024), 208.

[36] T. Wang, F. Lucka, and T. van Leeuwen, X-ray Computed Tomography Case Study: Triangle and Pentagon Datasets with Various Sizes, Scales, and Noise Levels, Zenodo, Feb. 2025.

Received May 2025; revised September 2025.