

L2SR: Learning to Sample and Reconstruct for Accelerated MRI via Reinforcement Learning

Pu Yang¹ and Bin Dong²

¹ the School of Mathematical Sciences, Peking University, Beijing, P.R.China

² Beijing International Center for Mathematical Research, Peking University, Beijing, P.R.China

E-mail: dongbin@math.pku.edu.cn

Submitted 29 October 2023

Abstract. Magnetic Resonance Imaging (MRI) is a widely used medical imaging technique, but its long acquisition time can be a limiting factor in clinical settings. To address this issue, researchers have been exploring ways to reduce the acquisition time while maintaining the reconstruction quality. Previous works have focused on finding either sparse samplers with a fixed reconstructor or finding reconstructors with a fixed sampler. However, these approaches do not fully utilize the potential of joint learning of samplers and reconstructors. In this paper, we propose an alternating training framework for jointly learning a good pair of samplers and reconstructors via deep reinforcement learning (RL). In particular, we consider the process of MRI sampling as a sampling trajectory controlled by a sampler, and introduce a novel sparse-reward Partially Observed Markov Decision Process (POMDP) to formulate the MRI sampling trajectory. Compared to the dense-reward POMDP used in existing works, the proposed sparse-reward POMDP is more computationally efficient and has a provable advantage. Moreover, the proposed framework, called L2SR (Learning to Sample and Reconstruct), overcomes the training mismatch problem that arises in previous methods that use dense-reward POMDP. By alternately updating samplers and reconstructors, L2SR learns a pair of samplers and reconstructors that achieve state-of-the-art reconstruction performances on the fastMRI dataset. Codes are available at <https://github.com/yangpuPKU/L2SR-Learning-to-Sample-and-Reconstruct>.

Keywords: Accelerated MRI, Deep RL, Sparse-reward POMDP, Alternating Training

1. Introduction

Magnetic resonance imaging (MRI) is a valuable medical imaging tool that does not involve ionizing radiation. However, it suffers from slow data acquisition, which limits its widespread clinical use.

To accelerate MRI, Compressed sensing (CS) [1] techniques have been introduced to speed up the MRI scanning process by reducing the number of measurements. Traditional CS-MRI methods [2, 3] first subsample the measurements in the Fourier domain with heuristic sampling policies and then reconstruct MR images by solving an ill-posed inverse problem. These optimization-based methods are interpretable

and with theoretical guarantees. Although they have notably reduced acquisition time compared to commercial MRI, they still have room for improvement.

Recent advances in Deep Learning (DL) [4–10] have led to significant improvements in MRI reconstruction by using DL models as reconstructors. These models use heuristic sampling policies similar to CS-MRI and are trained on specific datasets. However, the quality of the sampling policies and reconstructors can be quantified by examining their acceleration-quality trade-off. The heuristic sampling policies may have a limited acceleration-quality trade-off, as they cannot adaptively decide the sampling strategy according to the anatomical structures of patients.

To overcome this, some approaches learn an adaptive sampling policy rather than by human design. One such approach views the sequential sampling in k-space as a feed back control process, formulates the sampling trajectory as a Partially Observed Markov Decision Process (POMDP) and solves it via deep Reinforcement Learning (RL). Existing dynamic sampling methods [11–13] propose using a dense-reward POMDP to learn parametric samplers with a fixed pre-trained reconstructor via deep RL. However, these methods suffer from the training mismatch, where the sampling pattern used to pre-train the reconstructor is different from the pattern given by the RL-learned sampler. This can lead to suboptimal performance in the learned pair of samplers and reconstructors.

In this paper, we propose a novel alternating training framework, L2SR (Learning to Sample and Reconstruct), which incorporates an innovative sparse-reward POMDP formulation to facilitate the joint optimization of MRI sampling policies and reconstruction models. In contrast to previous efforts [11–13] our approach addresses the training mismatch issue, which emerges when samplers and reconstructors are optimized as isolated subproblems rather than focusing on the comprehensive joint problem. Unlike dense-reward POMDPs, our sparse-reward formulation uniquely disentangles the sampling and reconstruction stages by deliberately eliminating intermediate reconstructions during the sampling process. The final reconstruction occurs only at the end, leading to a unique joint sampler-reconstructor optimization problem. This problem can be solved by alternately optimizing the sampler (with the reconstructor kept constant) through RL, and the reconstructor (with the sampler kept constant) through back-propagation. Our experiments on fastMRI dataset [14] demonstrates L2SR’s capability to enhance MRI reconstruction quality.

The main contributions of this paper are:

- We formulate the MRI sampling and reconstruction process as a joint optimization problem of learning optimal MRI samplers and reconstructors, a novel formulation not explored in previous study. This formulation could potentially enhance the trade-off between acceleration and quality.
- We propose a novel sparse-reward POMDP formulation for MRI sampling that addresses three key issues with dense-reward POMDPs: (1) It is more computationally efficient for inference by avoiding intermediate reconstructions; (2) It avoids the distributional mismatch issue, which is proven to have an advantage in reconstruction performance; (3) It provides a viable approach to tackle the joint sampler-reconstructor optimization problem.
- We develop an alternating training framework called L2SR that leverages the sparse-reward POMDP to jointly optimize samplers and reconstructors. A key novelty is that L2SR overcomes the training mismatch issue that arises due to the decomposition of the joint optimization problem into separate sampler and

reconstructor subproblems in prior works.

- Our experiments demonstrate the state-of-the-art reconstruction performance on the fastMRI benchmark, validating the benefits of our proposed solutions for the joint optimization problem.

The remainder of this paper is organized as follows. In section 2, we discuss related works. In section 3, we provide notations of accelerated MRI and formulate optimization problems. In section 4, we delve into the issues inherent to the design of the dense-reward POMDP, followed by the introduction of the sparse-reward POMDP, along with its associated dynamic sampling training framework and alternating training framework. Experimental results are presented in section 5, and finally, we conclude our findings in section 6.

2. Related Work

In this section, we briefly review prior works on sampling and reconstruction in accelerated MRI.

2.1. Learning Reconstructors with Heuristic Sampling

Existing heuristic sampling policies for MRI reconstruction [2, 15, 16] include uniform density random sampling, variable density sampling, Poisson-disc sampling, equispaced sampling, and continuous-trajectory variable density sampling. These sampling strategies are very simple and easy to implement. Methods of learning reconstructors with a fixed heuristic sampling policy can be divided into two categories: optimization-based methods and DL-based methods.

Optimization-based methods use constrained optimization problems to obtain the reconstruction image. These problems minimize an ℓ_1 -norm regularization term for sparsity while constraining the ℓ_2 -norm error in k -space for accuracy (see Equation [3] in [2]). The regularization term is often referred to as Total Variation (TV) [17] or Wavelet [18]. These optimization problems can be solved using iterative optimization methods such as Iterative Soft Thresholding Algorithm (ISTA) [19] and Iterative Reweighted Least Squares (IRLS) [20]. While these methods are interpretable and have theoretical guarantees, they depend on human-designed image priors and are not adaptive to specific datasets.

DL-based methods use DL models as reconstructors for MRI, significantly improving reconstruction quality. Unet-based network models [4, 5] and GAN-based network models [6, 7] learn an end-to-end mapping from sampled measurements to reconstruction images. Unrolled dynamic models such as ADMM-net [8], Variational-net [9], and ISTA-net [10] first unroll an iterative algorithm to form the backbone network architecture and then replace some of its operators with neural networks. These methods are data-driven, and hence adaptive to specific datasets. However, compared to optimization-based methods, DL-based methods are less interpretable and may have generalization issues.

2.2. Learning Dynamic Sampling with Fixed Reconstructors via Deep RL

Heuristic sampling policies may not be adaptable to specific datasets, which limits their acceleration-quality trade-off. Since the sampling process is carried out over time, it can be considered a finite sequential decision-making process. In Computed

Tomography (CT) scanning, this dynamic sampling method was first proposed by [21], and a similar idea was later proposed in the MRI region. These methods [11–13] formulate the MRI sampling trajectory as a Partially Observed Markov Decision Process (POMDP) (see section 3.2) and learn parametric samplers with a fixed pre-trained reconstructor by modern deep RL algorithms, such as DDQN [22] and policy gradient [23]. RL enables the learning of a non-greedy sequential decision function without relying on hand-designed policies. These methods use the RL-learned policies to guide the scanning based on the measurements collected in the preceding stage, thus achieving personalized and data-adaptive scanning.

2.3. Jointly Learning Samplers and Reconstructors via Back-Propagation

If we fix a sampler to find an optimal reconstructor or fix a reconstructor to find an optimal sampler, we may obtain suboptimal pairs of samplers and reconstructors. Therefore, recent works aim to learn a good pair of samplers and reconstructors for a better acceleration-quality trade-off. These works propose to jointly train samplers and reconstructors end-to-end using Back-Propagation (BP).

One line of works [24–28] use a learnable column vector of probabilistic sub-sampling masks \mathbf{p} as the sampler, which is independent of the specific image, and jointly train it together with a neural network-based reconstruction model \mathcal{R} . These works learn a better sampler \mathbf{p} than human-designed heuristic sampling policies. However, these samplers are not adaptive to each specific image. The ideas of these works can be illustrated by the following diagram:

$$\boxed{\mathbf{p}} \xrightarrow[\mathbf{y}]{\text{sample}} \mathbf{y}_T \xrightarrow{\boxed{\mathcal{R}}} \mathbf{x}_T$$

where the modules in the boxes are learnable.

Another line of works [29–32] use a neural network-based model π as the sampler, which is adaptive to sampled measurements. The sampler π is jointly trained with a DL-based reconstructor \mathcal{R} end-to-end via BP. These methods learn sampling strategies that are adaptive to each image. However, training the sampler and the reconstructor by BP can be challenging. On the one hand, gradients are passed through long trajectories leading to gradient vanishing or exploding. On the other hand, GPU memory limits the size of neural networks due to long trajectories. Therefore, these works often solve weakened forms of the joint optimization problem (3.1). The ideas of these works can be illustrated by the following diagram:

$$\mathbf{y}_0 \xrightarrow{\boxed{\mathcal{R}}} \mathbf{x}_0 \xrightarrow[\mathbf{y}]{\boxed{\pi}} \mathbf{y}_1 \xrightarrow{\boxed{\mathcal{R}}} \mathbf{x}_1 \xrightarrow[\mathbf{y}]{\boxed{\pi}} \mathbf{y}_2 \cdots \mathbf{y}_T \xrightarrow{\boxed{\mathcal{R}}} \mathbf{x}_T$$

where the modules in the boxes are learnable.

In this paper, we also aim to learn a pair of samplers and reconstructors jointly. Instead of training end-to-end by BP, we will design an RL-based joint training framework.

3. Problem Formulation

In section 3.1, we specify the notation of CSMRI. In section 3.2, we formulate the optimization problem.

3.1. Notation

In the context of this study, our focus is confined to the analysis of two-dimensional (2D) accelerated MRI in conjunction with one-dimensional (1D) vertical Cartesian sampling. We define a ground truth image, denoted by the vector $\mathbf{x} \in \mathbb{R}^{N \times N}$, and the corresponding fully sampled measurements in the Fourier domain (often referred to as k-space), represented by the vector $\mathbf{y} \in \mathbb{C}^{N \times N}$, obtained via the two-dimensional Discrete Fourier Transform, \mathcal{F} , i.e., $\mathbf{y} = \mathcal{F}(\mathbf{x})$.

Further, let \mathcal{D} signify a dataset comprising ground truth images. In accelerated MRI, the acquisition of measurements in the Fourier domain is realized by masking \mathbf{y} using a binary mask matrix $\mathbf{M} \in \{0, 1\}^{N \times N}$, rendering the observation of $\mathbf{M} \odot \mathbf{y}$, where \odot designates the Hadamard product. In this study, all binary mask matrices satisfy the property that each of their columns is either all 0's or all 1's, unless specified otherwise.

The process of reconstruction involves a specific mapping $\mathcal{R} : \mathbb{C}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$, which is customarily formulated as a parametric neural network denoted by $\mathcal{R}(\cdot; \theta_{\mathcal{R}})$.

Additionally, we define a heuristic sampling policy π_t^h as a probability distributional function accepting the N -dimensional binary column vector \mathbf{a} as an input. Specifically, π_t^h sample \mathbf{a} under the constraint that $\|\mathbf{a}\|_1 = t$, i.e., π_t^h is constructed in such a way that $\pi_t^h(\mathbf{a}) = 0$ if $\|\mathbf{a}\|_1 \neq t$. Denote the mask matrix $\mathbf{M}^{\mathbf{a}} = \mathbf{1} \cdot \mathbf{a}^T$ with the property that $\|\mathbf{M}^{\mathbf{a}}\|_{\infty} = \|\mathbf{a}\|_1$.

We give a list of notation index and where it is defined in [Appendix A](#).

3.2. Optimization Problems

To get an optimal pair of samplers and reconstructors, we want to solve the following joint optimization problem based on dense-reward POMDP

$$\begin{aligned}
 & \max_{\theta_{\pi}, \theta_{\mathcal{R}}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\pi} S(\mathbf{x}_T, \mathbf{x}) \\
 & \text{s.t. } \mathbf{y}_t = \mathbf{M}_t \odot \mathcal{F}(\mathbf{x}), \quad t = 0, 1, \dots, T, \\
 & \quad \mathbf{x}_t = \mathcal{R}(\mathbf{y}_t; \theta_{\mathcal{R}}), \quad t = 0, 1, \dots, T, \\
 & \quad a_t \sim \pi(\cdot \mid \mathbf{x}_t; \theta_{\pi}), \quad t = 0, 1, \dots, T-1, \\
 & \quad \mathbf{M}_{t+1} = \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t}), \quad t = 0, 1, \dots, T-1,
 \end{aligned} \tag{3.1}$$

where T denotes the length of sampling trajectory, S is an image similarity metric, $a_t \in \mathcal{A} = \{n\}_{n=1}^N$, the sampler $\pi : \mathbb{R}^{N \times N} \rightarrow \Delta^N$ is the N -dimensional discrete conditional probability distribution function, $\Delta^N = \{\mathbf{u} \in [0, 1]^N \mid \sum_i u_i = 1\}$ denotes the N -dimensional probability simplex, \mathbf{M}_0 is an initial binary mask matrix, and $\mathbb{1}(\cdot)$ is a matrix indicator function. For simplicity, we denote the objective function $\mathbb{E}_{\pi} S(\mathbf{x}_T, \mathbf{x})$ in (3.1) as $J_T^{\text{dense}}(\mathbf{x}; \pi, \mathcal{R})$.

One of the challenges in solving this joint optimization problem directly is its complexity. One approach is to train both θ_{π} and $\theta_{\mathcal{R}}$ end-to-end by BP, as mentioned in section 2.3. However, training may fail due to the long sampling trajectory, and existing BP methods often solve weakened forms of (3.1). Another approach is to use reinforcement learning (RL). However, the existing dense-reward partially observable Markov decision process (POMDP) may not be suitable for end-to-end joint training. When training the sampler via RL, we must keep the environment unchanged, and thus the reconstructor as a part of the environment must also remain the same. This

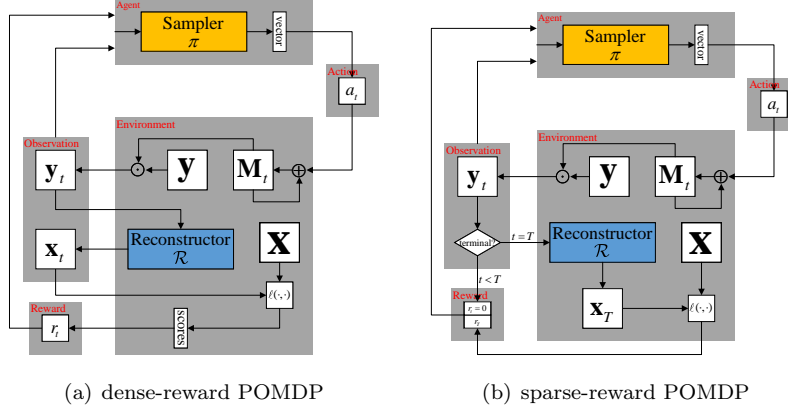


Figure 1. Diagram of the dense-reward and sparse-reward POMDP.

contradicts the end-to-end joint training. Additionally, it is difficult to design a multi-agent POMDP for the two learnable agents, the sampler π and the reconstructor \mathcal{R} .

DL-based MRI methods can obtain a better reconstructor with a fixed heuristic sampling policy π^h by solving the following suboptimization problem of (3.1):

$$\max_{\theta_{\mathcal{R}}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{a} \sim \pi^h} [\mathcal{S}(\mathcal{R}(\mathbf{M}^{\mathbf{a}} \odot \mathcal{F}(\mathbf{x}); \theta_{\mathcal{R}}), \mathbf{x})]. \quad (3.2)$$

Dynamic sampling methods can obtain better samplers by solving the following suboptimization problem of (3.1) with a fixed pre-trained reconstruction model $\mathcal{R}(\cdot)$:

$$\max_{\theta_{\pi}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{dense}}(\mathbf{x}; \pi(\cdot; \theta_{\pi}), \mathcal{R}). \quad (3.3)$$

To solve (3.3), they formulate the sequential sampling process as the following *dense-reward POMDP* (illustrated in figure 1(a)):

- Observation

$$\mathbf{y}_t = \mathbf{M}_t \odot \mathbf{y}, \quad t = 0, 1, \dots, T, \quad (3.4)$$

$$\mathbf{x}_t = \mathcal{R}(\mathbf{y}_t), \quad t = 0, 1, \dots, T. \quad (3.5)$$

- Action set \mathcal{A} and action $a_t \in \mathcal{A}$

$$a_t \sim \pi(\cdot | \mathbf{x}_t), \quad t = 0, 1, \dots, T-1. \quad (3.6)$$

- Transition

$$\mathbf{M}_{t+1} = \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t}), \quad t = 0, 1, \dots, T-1, \quad (3.7)$$

and \mathbf{y}_{t+1} and \mathbf{x}_{t+1} is calculated by (3.4) and (3.5).

- Reward

$$r_t = \mathcal{S}(\mathbf{x}_t, \mathbf{x}) - \mathcal{S}(\mathbf{x}_{t-1}, \mathbf{x}), \quad t = 1, \dots, T. \quad (3.8)$$

- Discount factor

$$\gamma \in [0, 1]. \quad (3.9)$$

These methods solve (3.3) via deep RL with the above dense-reward POMDP.

However, a recent study by [32] has revealed a problem with training when solving the suboptimization problem (3.3). Specifically, there is a *training mismatch* that occurs when the heuristic sampling policy used to pre-train the reconstructor is different from the policy learned through reinforcement learning. This mismatch leads to a suboptimal pair of samplers and reconstructors for the joint optimization problem (3.1).

4. Proposed Method

In this section, we first conduct an analysis of the dense-reward POMDP and identify some inherent issues in section 4.1. Based on these insights, we then propose a novel sparse-reward POMDP in section 4.2 to address the identified issues. Furthermore, we present two novel training frameworks: dynamic sampling training framework in section 4.3 and alternate training framework in section 4.4, both of which stem from the sparse-reward POMDP.

4.1. The Issues with Dense-reward POMDP

Apart from the training mismatch issue arising from tackling the sub-optimization problem, our comprehensive analysis of the dense-reward POMDP has revealed issues associated with the MDP design. Specifically, the dense-reward POMDP embeds reconstructions into the environment, particularly within transitions and rewards. This design introduces two main problems: high computational cost and distributional mismatch. These findings pave the way for the introduction of our sparse-reward POMDP.

4.1.1. High Computational Cost Since the dense-reward POMDP involves reconstructions in the transition, it first performs reconstruction (3.5) and then sampling (3.6) at each step in the inference process. However, this approach is very time-consuming since it requires performing many reconstructions. The dynamic sampling method for CT scanning [21] uses the original sampling signals rather than reconstructed images as the input of the sampler, which is more computationally efficient. We aim to find a better POMDP that requires only one reconstruction in the inference process.

4.1.2. Distributional Mismatch Since the dense-reward POMDP involves reconstructions in the reward calculations (3.8), it requires evaluating the similarity of the ground truth and the reconstruction image $S(\mathbf{x}_t, \mathbf{x})$ at every intermediate state. Therefore, the dense-reward POMDP requires a reliable reconstructor for all the intermediate states, which means the reconstructor should be pre-trained with a mixture of heuristic sampling policies $\pi_{\text{mix}}^h = \sum_{t=\|\mathbf{M}_0\|_\infty}^{\|\mathbf{M}_0\|_\infty+T} c_t \pi_t^h$ where $\sum c_t = 1$, instead of only the terminal sampling policy $\pi_{T+\|\mathbf{M}_0\|_\infty}^h$. However, as we will show later, pre-training with the mixture of policies may not improve the final reconstruction performance, i.e., the optimal value of the joint optimization problem (3.1). We will justify this claim in theorem 2.

4.2. Sparse-Reward POMDP

To improve inference efficiency and eliminate distributional mismatch, we propose a *sparse-reward POMDP* that reconstructs after completing the entire trajectory and receives a non-zero reward after the final reconstruction. Specifically, the proposed sparse-reward POMDP is as follows (illustrated in figure 1(b)):

- Observation

$$\mathbf{y}_t = \mathbf{M}_t \odot \mathbf{y}, \quad t = 0, 1, \dots, T. \quad (4.1)$$

- Action set \mathcal{A} and action $a_t \in \mathcal{A}$

$$a_t \sim \pi(\cdot \mid \mathbf{y}_t), \quad t = 0, 1, \dots, T-1, \quad (4.2)$$

where the sampler $\pi : \mathbb{C}^{N \times N} \rightarrow \Delta^N$ is the N-dimensional discrete conditional probability distribution function.

- Transition

$$\mathbf{M}_{t+1} = \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t}), \quad t = 0, 1, \dots, T-1, \quad (4.3)$$

and \mathbf{y}_t is calculated by (4.1)

- Reward

$$\begin{cases} r_t = 0, & t = 1, \dots, T-1 \\ r_T = S(\mathcal{R}(\mathbf{y}_T), \mathbf{x}) \end{cases}. \quad (4.4)$$

- Discount factor

$$\gamma = 1. \quad (4.5)$$

It derives a sparse-reward joint optimization problem:

$$\begin{aligned} \max_{\theta_\pi, \theta_{\mathcal{R}}} \quad & \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_\pi S(\mathcal{R}(\mathbf{y}_T), \mathbf{x}) \\ \text{s.t.} \quad & \mathbf{y}_t = \mathbf{M}_t \odot \mathcal{F}(\mathbf{x}), \quad t = 0, 1, \dots, T, \\ & a_t \sim \pi(\cdot \mid \mathbf{y}_t; \theta_\pi), \quad t = 0, 1, \dots, T-1, \\ & \mathbf{M}_{t+1} = \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t}), \quad t = 0, 1, \dots, T-1, \end{aligned} \quad (4.6)$$

and we denote the objective function $\mathbb{E}_\pi S(\mathcal{R}(\mathbf{y}_T), \mathbf{x})$ in (4.6) as $J_T^{\text{sparse}}(\mathbf{x}; \pi, \mathcal{R})$.

Solving the sparse-reward joint optimization problem yields a reconstructor and a sampler that are at least as strong as those obtained from solving the dense-reward joint optimization problem.

Theorem 1 (joint optimization problem) *The optimal values of optimization problems (3.1) and (4.6) satisfy*

$$\sup_{\substack{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N) \\ \mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{dense}}(\mathbf{x}) \leq \sup_{\substack{\pi \in C(\mathbb{C}^{N \times N}, \Delta^N) \\ \mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{sparse}}(\mathbf{x}). \quad (4.7)$$

Proof See [Appendix B.1](#).

Remark. In our optimization problems (3.1) and (4.6), the sampler and reconstructor are parameterized by θ_π and $\theta_{\mathcal{R}}$, reflecting their potential practical implementation as neural networks with learnable parameters. However, in theorem 1, we place them within the broader continuous function space, which provides analytical

convenience for theoretical explorations. This remark also applies to the following theorem 2.

Theorem 1 gives us the confidence to switch from solving the dense-reward joint optimization problem defined in equation (3.1) to solving the sparse-reward joint optimization problem defined in equation (4.6).

Next, we can demonstrate that the sparse-reward POMDP is more computationally efficient than the dense-reward POMDP. The inference process of the dense-reward POMDP can be represented as follows:

$$\mathbf{y}_0 \xrightarrow{\mathcal{R}} \mathbf{x}_0 \xrightarrow[\mathbf{y}]{\pi} \mathbf{y}_1 \xrightarrow{\mathcal{R}} \mathbf{x}_1 \xrightarrow[\mathbf{y}]{\pi} \mathbf{y}_2 \cdots \mathbf{y}_T \xrightarrow{\mathcal{R}} \mathbf{x}_T.$$

On the other hand, the proposed sparse-reward POMDP can be represented as:

$$\mathbf{y}_0 \xrightarrow[\mathbf{y}]{\pi} \mathbf{y}_1 \xrightarrow[\mathbf{y}]{\pi} \mathbf{y}_2 \cdots \mathbf{y}_T \xrightarrow{\mathcal{R}} \mathbf{x}_T.$$

It can be observed that the sparse-reward POMDP does not require reconstruction at the sampling stage. Thus, the sparse-reward POMDP is computationally more efficient.

Then, we provide a theoretical explanation of how sparse-reward POMDP avoids the distributional mismatch in dense-reward POMDP. In the sparse-reward POMDP, the reconstructor only operates on the terminal state to obtain the reward, and a terminal heuristic sampling policy π_T^h is used to pre-train the reconstructor $\mathcal{R}(\cdot)$. We can learn samplers with the proposed sparse-reward POMDP by solving the suboptimization problem of (4.6) with a fixed pre-trained reconstruction model $\mathcal{R}(\cdot)$:

$$\max_{\theta_\pi} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{sparse}}(\mathbf{x}; \pi(\cdot; \theta_\pi), \mathcal{R}). \quad (4.8)$$

The following theorem ensures that the sparse-reward suboptimization problem (4.8) produces a sampler that is no weaker than the dense-reward one (3.3), thus supporting the claim in the previous section.

Theorem 2 (distributional mismatch) *Let the terminal sampling policy $\pi_{T+\|\mathbf{M}_0\|_\infty}^h$ satisfies $\pi_{T+\|\mathbf{M}_0\|_\infty}^h(\mathbf{a}) > 0$ for all binary column vectors \mathbf{a} satisfying $\|\mathbf{a}\|_1 = T + \|\mathbf{M}_0\|_\infty$. Suppose that there exists a reconstructor $\mathcal{R}^{\text{sparse}}$ satisfying*

$$\mathcal{R}^{\text{sparse}} \in \operatorname{argmax}_{\mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{a} \sim \pi_{T+\|\mathbf{M}_0\|_\infty}^h} S(\mathcal{R}(\mathbf{M}^{\mathbf{a}} \odot \mathcal{F}(\mathbf{x})), \mathbf{x}). \quad (4.9)$$

Then, for any continuous reconstructor $\mathcal{R}^{\text{dense}}$, we have

$$\sup_{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N)} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{dense}}(\mathbf{x}; \pi, \mathcal{R}^{\text{dense}}) \leq \sup_{\pi \in C(\mathbb{C}^{N \times N}, \Delta^N)} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{sparse}}(\mathbf{x}; \pi, \mathcal{R}^{\text{sparse}}). \quad (4.10)$$

Proof See [Appendix B.2](#).

Remark 1. Theorem 2 demonstrates that the optimal value of (3.3) is no greater than that of (4.8). Therefore, compared to pre-training $\mathcal{R}^{\text{sparse}}$ with the terminal heuristic sampling policy π_T^h for the sparse-reward POMDP, pre-training

with any other heuristic sampling policies for the dense-reward POMDP yields no benefit. In particular, previous dynamic sampling methods have utilized a mixture of heuristic sampling policy $\pi_{\text{mix}}^h = \sum_{t=\|\mathbf{M}_0\|_\infty}^{\|\mathbf{M}_0\|_\infty+T} c_t \pi_t^h$ for pre-training reconstructors with the dense-reward POMDP, where $\sum c_t = 1$. Furthermore, we will demonstrate empirically that using π_{mix}^h for pre-training is generally less effective than using π_T^h , which we refer to as a distributional mismatch.

Remark 2. Assumption ‘ $\pi_{T+\|\mathbf{M}_0\|_\infty}^h(\mathbf{a}) > 0$ for all binary column vectors \mathbf{a} satisfying $\|\mathbf{a}\|_1 = T + \|\mathbf{M}_0\|_\infty$ ’ is necessary and easily attainable in experimental settings. The conclusion does not hold without this assumption, as demonstrated by a counter-example in [Appendix B.2](#).

Apart from improving inference efficiency and resolving the distributional mismatch, the proposed sparse-reward POMDP also offers an additional advantage. It decouples the sampling and reconstruction processes, which means that the sampling trajectories are no longer reliant on the reconstructor, but only on the sampler and images. This decoupling provides a viable solution for tackling the joint optimization problem of sparse-reward (4.6).

4.3. Dynamic Sampling Training Framework

In this subsection, we present a dynamic sampling training framework that utilizes a pretrained fixed reconstructor to solve the sparse-reward suboptimization problem (4.8). Our experimental results in section 5 demonstrate that this algorithm outperforms previous RL-based methods that rely on dense-reward POMDP, thus providing further evidence that eliminating distributional mismatch can improve reconstruction performance.

We begin by pre-training the reconstructor using a terminal heuristic sampling policy π_T^h . Subsequently, we employ the proposed sparse-reward POMDP through RL to train the sampler with the fixed reconstructor. Mathematically, we first solve for $\theta_{\mathcal{R}}$ by maximizing the expected reconstruction loss S of the reconstructor:

$$\theta_{\mathcal{R}} = \operatorname{argmax}_{\theta_{\mathcal{R}}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{a} \sim \pi_T^h} [S(\mathcal{R}(\mathbf{M}^{\mathbf{a}} \odot \mathcal{F}(\mathbf{x}); \theta_{\mathcal{R}}), \mathbf{x})]. \quad (4.11)$$

and then solve for θ_{π} by maximizing the expected cumulative reward J_T^{sparse} under the sparse-reward POMDP framework:

$$\theta_{\pi} = \operatorname{argmax}_{\theta_{\pi}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{sparse}}(\mathbf{x}; \pi(\cdot; \theta_{\pi}), \mathcal{R}(\cdot; \theta_{\mathcal{R}})). \quad (4.12)$$

To solve the optimization problem (4.11), we use a standard DL-based MRI reconstruction approach and train the reconstructor using the Adam optimizer [33].

To solve the optimization problem (4.12), we leverage the proposed sparse-reward POMDP through RL. Specifically, we adopt the Actor-to-Critic (A2C) algorithm [34], an efficient deep RL method to optimize the policy $\pi(\cdot; \theta_{\pi})$.

We refer to the entire training process as L2S which stands for Learning to Sample. We summarize this process in algorithm 1 and illustrated it in figure 2(a).

Remark. While we still require a heuristic policy π^h , it is only used to initialize the reconstructor at the beginning and does not rely on human knowledge or design. Even random initialization is sufficient to provide a reasonable starting point before training (see section 5). The core of our training process involves RL-based optimization of π in a data-driven way, without restricting to human-designed heuristics. This remark also applies to algorithm 2.

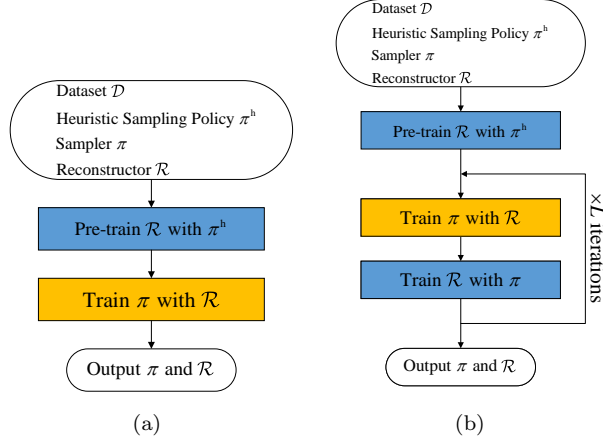


Figure 2. Overview of (a) the training framework of dynamic sampling with fixed reconstructors, and (b) the alternating training framework.

Algorithm 1 Learning to Sample (L2S)

Require: a sampler π , a reconstructor \mathcal{R} , a heuristic sampling policy π^h , an MRI dataset \mathcal{D} , and the maximum number of BP iterations n

for $i = 1$ to n **do**

 sample a batch $\{\mathbf{x}^{(b)}\}_{b=1}^B \sim \mathcal{D}$ and binary mask vectors $\mathbf{a}^{(b)} \sim \pi^h$

 compute $\mathcal{L}(\theta_{\mathcal{R}}) = -\frac{1}{B} \sum_{b=1}^B S(\mathcal{R}(\mathbf{M}^{\mathbf{a}^{(b)}} \mathcal{F} \mathbf{x}^{(b)}; \theta_{\mathcal{R}}), \mathbf{x}^{(b)})$

 optimize $\theta_{\mathcal{R}}$ with the loss \mathcal{L} by Adam

end for

build environment (sparse-reward POMDP)

train sampler π with the environment by A2C

Output sampler π and reconstructor \mathcal{R}

4.4. Alternating Training Framework

In this subsection, we present an alternating training framework to solve the sparse-reward joint optimization problem (4.6). By jointly optimizing the sampler and reconstructor, this algorithm eliminates training mismatch.

We begin by pre-training the reconstructor using a terminal heuristic sampling policy π_T^h . During training, we alternately train the sampler with the fixed reconstructor and train the reconstructor with the fixed sampler. Mathematically, we solve the following optimization problems in sequence:

$$\theta_{\mathcal{R}}^{(0)} = \operatorname{argmax}_{\theta_{\mathcal{R}}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{a} \sim \pi_T^h} [S(\mathcal{R}(\mathbf{M}^{\mathbf{a}} \odot \mathcal{F}(\mathbf{x}); \theta_{\mathcal{R}}), \mathbf{x})], \quad (4.13)$$

$$\theta_{\pi}^{(l)} = \operatorname{argmax}_{\theta_{\pi}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{sparse}}(\mathbf{x}; \pi(\cdot; \theta_{\pi}), \mathcal{R}(\cdot; \theta_{\mathcal{R}}^{(l-1)})), \quad (4.14)$$

$$\theta_{\mathcal{R}}^{(l)} = \operatorname{argmax}_{\theta_{\mathcal{R}}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} J_T^{\text{sparse}}(\mathbf{x}; \pi(\cdot; \theta_{\pi}^{(l)}), \mathcal{R}(\cdot; \theta_{\mathcal{R}})), \quad (4.15)$$

where L is the number of alternation and $l = 1, 2, \dots, L$.

Algorithm 2 Learning to Sample and Reconstruct (L2SR)

Require: a sampler π , a reconstructor \mathcal{R} , a heuristic sampling policy π^h , an MRI dataset \mathcal{D} , the number of alternations L , and the maximum number of BP iterations $\{n^{(l)}\}_{l=0}^L$

```

for  $i = 1$  to  $n^{(0)}$  do
  sample a batch  $\{\mathbf{x}^{(b)}\}_{b=1}^B \sim \mathcal{D}$  and binary mask vectors  $\mathbf{a}^{(b)} \sim \pi^h$ 
  compute  $\mathcal{L}(\theta_{\mathcal{R}}) = -\frac{1}{B} \sum_{b=1}^B S(\mathcal{R}(\mathbf{M}^{\mathbf{a}^{(b)}} \mathcal{F}\mathbf{x}^{(b)}; \theta_{\mathcal{R}}), \mathbf{x}^{(b)})$ 
  optimize  $\theta_{\mathcal{R}}$  with the loss  $\mathcal{L}$  by Adam
end for
for  $l = 1$  to  $L$  do
  build environment (sparse-reward POMDP)
  train sampler  $\pi$  with the environment by A2C
  for  $i = 1$  to  $n^{(l)}$  do
    sample a batch  $\{\mathbf{x}^{(b)}\}_{b=1}^B \sim \mathcal{D}$  and its acquisition sequences  $\mathbf{a}^{(b)} \sim \pi(\mathbf{x}^{(b)})$ 
    compute  $\mathcal{L}(\theta_{\mathcal{R}}) = -\frac{1}{B} \sum_{b=1}^B S(\mathcal{R}(\mathbf{M}^{\mathbf{a}^{(b)}} \mathcal{F}\mathbf{x}^{(b)}; \theta_{\mathcal{R}}), \mathbf{x}^{(b)})$ 
    optimize  $\theta_{\mathcal{R}}$  with the loss  $\mathcal{L}$  by Adam
  end for
end for
Output sampler  $\pi$  and reconstructor  $\mathcal{R}$ 

```

Optimization problems (4.13) and (4.14) are identical to (4.11) and (4.12) respectively. We employ the same methods to solve them.

Thanks to the proposed sparse-reward POMDP, which separates sampling and reconstruction, we can solve optimization problem (4.15) using BP. The following proposition enables training the reconstructor with a fixed sampler using gradient-based techniques such as the Adam optimizer.

Proposition 1 *The derivative of J_T^{sparse} w.r.t. $\theta_{\mathcal{R}}$ is*

$$\nabla_{\theta_{\mathcal{R}}} J_T^{\text{sparse}}(\mathbf{x}) = \mathbb{E}_{\{a_t\}_{t=0}^{T-1} \sim \pi} [\nabla_{\theta_{\mathcal{R}}} S(\mathcal{R}(\mathbf{y}_T; \theta_{\mathcal{R}}), \mathbf{x})], \quad (4.16)$$

where $\{a_t\}_{t=0}^{T-1} \sim \pi$ means a sequential acquisition according to (4.6) and $\mathbf{y}_T = \mathbf{y}_T(\mathbf{M}_0, \{a_t\}_{t=0}^{T-1}) = (\mathbf{M}_0 + \sum_{t=0}^{T-1} \mathbf{M}^{a_t}) \odot \mathbf{y}$.

Proof See [Appendix B.3](#).

This proposition states that we can compute the gradient of J_T^{sparse} w.r.t. $\theta_{\mathcal{R}}$. Specifically, we sample a sampling trajectory $\{a_t\}_{t=0}^{T-1}$ from the ground truth \mathbf{x} using the learned policy π to get \mathbf{y}_T , then reconstruct \mathbf{x}_T by applying the reconstruction function \mathcal{R} to \mathbf{y}_T , and finally compute the gradient of the similarity between the reconstructed image and the ground truth w.r.t. $\theta_{\mathcal{R}}$. Furthermore, we run multiple trajectories to compute the average of the gradient, and thus optimize (4.15) by Adam optimizer.

We refer to the entire training process as L2SR which stands for Learning to Sample and Reconstruct. We summarize this process in algorithm 2 and illustrated it in figure 2(b).

Remark. The proposed alternating training framework is not suitable for the dense-reward POMDP, since there is no equivalent derivative of J_T^{dense} w.r.t. $\theta_{\mathcal{R}}$ like the one presented in proposition 1. We provide a detailed explanation of this limitation in [Appendix B.4](#).

Table 1. Acceleration factors and initial accelerations factors of random sampling policies for pre-training reconstructors. We utilize $\mathcal{R}^{\text{dense}}$ for PG-MRI [11] and Greedy Oracle, and $\mathcal{R}^{\text{sparse}}$ for Random and L2S.

$\times 4$ acceleration		
	acceleration factor	initial acceleration factor
$\mathcal{R}^{\text{dense}}$	[4,4,4,6,6,8]	[4,6,8,6,8,8]
$\mathcal{R}^{\text{sparse}}$	4	8 (Base) or 32 (Long)
$\times 8$ acceleration		
	acceleration factor	initial acceleration factor
$\mathcal{R}^{\text{dense}}$	[8,8,8,12,12,16]	[8,12,16,12,16,16]
$\mathcal{R}^{\text{sparse}}$	8	16 (Base) or 64 (Long)
$\times 16$ acceleration		
	acceleration factor	initial acceleration factor
$\mathcal{R}^{\text{dense}}$	[16,16,16,24,24,32]	[16,16,16,24,24,32]
$\mathcal{R}^{\text{sparse}}$	16	32 (Base) or 128 (Long)

5. Experiments

In this section, we delve into the comprehensive evaluation of our proposed methods, L2S and L2SR.

5.1. Implementation

Setup: We evaluate the proposed methods under two different settings. In the ‘**Fixed Reconstructor**’ (abbreviated as ‘Fixed’) setting, the reconstructor is pre-trained using a heuristic sampling policy and is subsequently utilized with our proposed **L2S** method. In contrast, the ‘**Joint Training**’ (abbreviated as ‘Joint’) setting involves the simultaneous training of both the sampler and the reconstructor, employing our novel **L2SR** framework.

Similarity Metric: In our training process, we adopt the structural similarity index (SSIM) [35] as the image similarity metric S . Although SSIM might not be optimized specifically for medical imaging tasks, our framework has been designed with flexibility in mind. It can easily integrate alternative metrics such as Peak Signal-to-Noise Ratio (PSNR) or Wasserstein GAN (WGAN) distance [36, 37], depending on the particular application requirements. This adaptability ensures that our choice of SSIM as the representative metric for this study does not limit the framework’s applicability or functionality, and alternative metrics can be substituted without any loss of generality.

Acquisition: We define the *acceleration factor* as $N/\|\mathbf{a}\|_1 (\geq 1)$ to measure the time overhead of sampling, denoted as $\times(N/\|\mathbf{a}\|_1)$ -acceleration. Most sampling policies start by acquiring a certain number of columns at the central low-frequency region, where more information is concentrated, and then sample the rest of the k -space according to their unique strategy. We also define the *initial acceleration factor* as the ratio between N and the number of sampling columns in the first state. We explore two types of initialization: *Base-horizon*, where the initial acceleration factor is twice the acceleration factor, and *Long-horizon*, where it is eight times the acceleration factor.

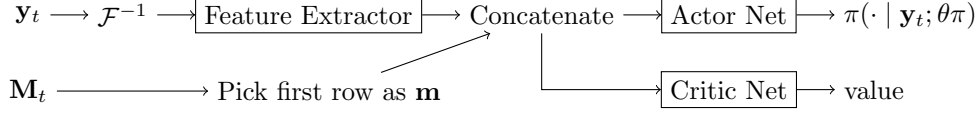


Figure 3. Transversal view of the policy model comprising a 2D Inverse Fourier Transform, a feature extractor, and actor and critic neural networks. The learnable parts are enclosed in boxes. The feature extractor is a CNN-based neural network following the architecture from [11]. The actor and critic nets are both constructed as fully connected neural networks.

Heuristic Sampling Policy for Pre-training: For the pre-training of reconstructors $\mathcal{R}^{\text{dense}}$ and $\mathcal{R}^{\text{sparse}}$, we employ a random sampling policy as the heuristic approach. The specific acceleration factors and initial acceleration factors used in this process are detailed in table 1. It is important to note that the selection of heuristic sampling policies for pre-training $\mathcal{R}^{\text{sparse}}$ aligns with the underlying assumptions detailed in theorem 2. This consistency ensures that our approach is grounded in a well-defined theoretical framework.

Dataset: We utilize the single-coil knee dataset and the multi-coil brain dataset from the commonly-used fastMRI dataset [14]. The data preprocessing follows the protocol established in [11]: We partition the dataset, reserving 20% of it for test and the rest for training. For the single-coil knee dataset, we derive the ground truth image \mathbf{x} by cropping the original to the central 128×128 region, using half the available volumes, and removing the outer slices of each volume, resulting in 6959 training slices, 1779 validation slices, and 1715 test slices. For the multi-coil brain dataset, we obtain the ground truth image \mathbf{x} by cropping to the central 256×256 region, using one-fifth of the available volumes, and assembling a dataset of 11312 training slices, 4372 validation slices, and 2832 test slices.

Policy model: The policy model’s architecture is depicted in figure 3. It takes as input the binary mask matrix \mathbf{M}_t and the observation \mathbf{y}_t , and processes them through a series of components including a 2D Inverse Fourier Transform, a feature extractor, and actor and critic networks. The final outputs are a discrete probability distribution $\pi(\cdot | \mathbf{y}_t; \theta\pi)$ over actions, generated by the actor net, and a scalar value representing the predicted value of the current state, produced by the critic net.

Reconstruction model: We employ an easy-to-implement standard U-Net [38] architecture with 16 channels, consisting of 8 blocks, as found in the fastMRI repository. The input to it is the zero-filled reconstruction image, represented by $|\mathcal{F}^{-1}(\mathbf{y}_t)|$, having the shape $(N, N, 1)$, and the output is the reconstruction image with the same shape. While we use U-Net with 16 channels for our experiments, it’s essential to note that our L2SR framework does not confine the choice of reconstruction models. The usage of the U-Net model serves as an illustrative example.

5.2. Algorithm Comparisons

In the ‘Fixed Reconstructor’ setting, we evaluate the L2S against three baseline methods: (1) Random: randomly selecting 1-d lines from a uniform distribution; (2) PG-MRI [11]: solving the dense-reward POMDP by policy gradient (which is one of the state-of-the-art dynamic sampling methods); (3) Greedy Oracle: a one-step oracle policy that has access to ground truth at test time. We pre-train the dense-reward reconstructor $\mathcal{R}^{\text{dense}}$ with a mixture heuristic sampling policy for PG-MRI and

Table 2. Main results: reconstruction results (mean and standard deviation) in terms of SSIM and PSNR values on the test dataset. For ‘Fixed (Reconstructor)’, $+\mathcal{R}$ means the heuristic sampling policy for pre-training. The best results (highest mean) for a specific acceleration factor and initial acceleration factor among the compared algorithms are shown in bold numbers. The best results for a specific acceleration factor are shown in blue numbers.

(a) Knee dataset $\times 4$ -acceleration

		Base-horizon		Long-horizon	
		SSIM	PSNR	SSIM	PSNR
Fixed	Random $+\mathcal{R}^{\text{sparse}}$	0.7222 \pm 0.0405	26.27 \pm 1.78	0.715 \pm 0.0329	25.16 \pm 1.78
	PG-MRI $+\mathcal{R}^{\text{dense}}$ [11]	0.7523 \pm 0.0375	26.54 \pm 1.74	0.7674 \pm 0.0299	25.23 \pm 1.59
	L2S $+\mathcal{R}^{\text{sparse}}$ (Ours)	0.7543 \pm 0.0372	26.93 \pm 1.79	0.7838 \pm 0.0286	26.86 \pm 1.68
	Greedy Oracle $+\mathcal{R}^{\text{dense}}$	0.7658 \pm 0.035	26.74 \pm 1.75	0.7837 \pm 0.0264	25.79 \pm 1.7
Joint	LOUPE [24]	0.7243 \pm 0.0387	26.2 \pm 1.75	0.72 \pm 0.0314	25.09 \pm 1.79
	τ -Step Seq [32]	0.7649 \pm 0.0408	27.26 \pm 1.85	0.8025 \pm 0.0315	27.69 \pm 1.81
	L2SR (Ours)	0.7681 \pm 0.0416	27.38 \pm 1.9	0.8097 \pm 0.0333	28.15 \pm 1.83

(b) Knee dataset $\times 8$ -acceleration

		Base-horizon		Long-horizon	
		SSIM	PSNR	SSIM	PSNR
Fixed	Random $+\mathcal{R}^{\text{sparse}}$	0.6039 \pm 0.0492	24.17 \pm 1.71	0.5915 \pm 0.0442	22.86 \pm 2.01
	PG-MRI $+\mathcal{R}^{\text{dense}}$ [11]	0.623 \pm 0.0504	24.21 \pm 1.69	0.633 \pm 0.0446	23.51 \pm 1.74
	L2S $+\mathcal{R}^{\text{sparse}}$ (Ours)	0.6258 \pm 0.0505	24.65 \pm 1.72	0.6441 \pm 0.0453	24.28 \pm 1.71
	Greedy Oracle $+\mathcal{R}^{\text{dense}}$	0.6376 \pm 0.0488	24.44 \pm 1.71	0.6509 \pm 0.043	23.41 \pm 1.74
Joint	LOUPE [24]	0.6002 \pm 0.0474	23.96 \pm 1.7	0.5876 \pm 0.0413	22.64 \pm 1.97
	τ -Step Seq [32]	0.6359 \pm 0.0572	25.07 \pm 1.73	0.6614 \pm 0.0508	25.26 \pm 1.76
	L2SR (Ours)	0.6332 \pm 0.053	24.98 \pm 1.74	0.6712 \pm 0.0511	25.61 \pm 1.79

(c) Brain dataset $\times 8$ -acceleration

		Base-horizon		Long-horizon	
		PSNR	SSIM	PSNR	SSIM
Fixed	Random $+\mathcal{R}^{\text{sparse}}$	0.8643 \pm 0.053	30.66 \pm 4.35	0.8405 \pm 0.0668	28.65 \pm 4.68
	PG-MRI $+\mathcal{R}^{\text{dense}}$ [11]	0.8771 \pm 0.0473	31.37 \pm 4.18	0.8766 \pm 0.0476	31.31 \pm 4.17
	L2S $+\mathcal{R}^{\text{sparse}}$ (Ours)	0.8782 \pm 0.0475	31.54 \pm 4.17	0.8686 \pm 0.0529	30.25 \pm 4.23
	Greedy Oracle $+\mathcal{R}^{\text{dense}}$	0.8822 \pm 0.0453	31.64 \pm 4.15	0.8771 \pm 0.0532	31.17 \pm 4.31
Joint	LOUPE [24]	0.8553 \pm 0.0555	30.15 \pm 4.32	0.8211 \pm 0.0691	27.34 \pm 3.99
	τ -Step Seq [32]	0.8862 \pm 0.041	31.93 \pm 3.98	0.8921 \pm 0.0397	32.07 \pm 3.88
	L2SR (Ours)	0.8899 \pm 0.043	32.57 \pm 4.09	0.8969 \pm 0.0417	32.87 \pm 4.02

(d) Brain dataset $\times 16$ -acceleration

		Base-horizon		Long-horizon	
		SSIM	PSNR	SSIM	PSNR
Fixed	Random $+\mathcal{R}^{\text{sparse}}$	80.31 \pm 0.0825	27.56 \pm 5.03	0.7837 \pm 0.0921	26.28 \pm 5.02
	PG-MRI $+\mathcal{R}^{\text{dense}}$ [11]	0.821 \pm 0.0722	28.25 \pm 4.71	0.8218 \pm 0.0721	28.36 \pm 4.68
	L2S $+\mathcal{R}^{\text{sparse}}$ (Ours)	0.8263 \pm 0.0703	28.59 \pm 4.73	0.819 \pm 0.0738	27.65 \pm 4.71
	Greedy Oracle $+\mathcal{R}^{\text{dense}}$	0.8312 \pm 0.0677	28.75 \pm 4.64	0.8251 \pm 0.0734	28.36 \pm 4.86
Joint	LOUPE [24]	0.8003 \pm 0.0821	27.55 \pm 4.977	0.7855 \pm 0.0894	26.29 \pm 4.748
	τ -Step Seq [32]	0.8314 \pm 0.0652	29.05 \pm 4.636	0.8384 \pm 0.0631	29.34 \pm 4.59
	L2SR (Ours)	0.8404 \pm 0.066	29.39 \pm 4.69	0.8456 \pm 0.0652	29.37 \pm 4.74

Greedy Oracle, and pre-train the sparse-reward reconstructor $\mathcal{R}^{\text{sparse}}$ with a terminal heuristic sampling policy for Random and L2S.

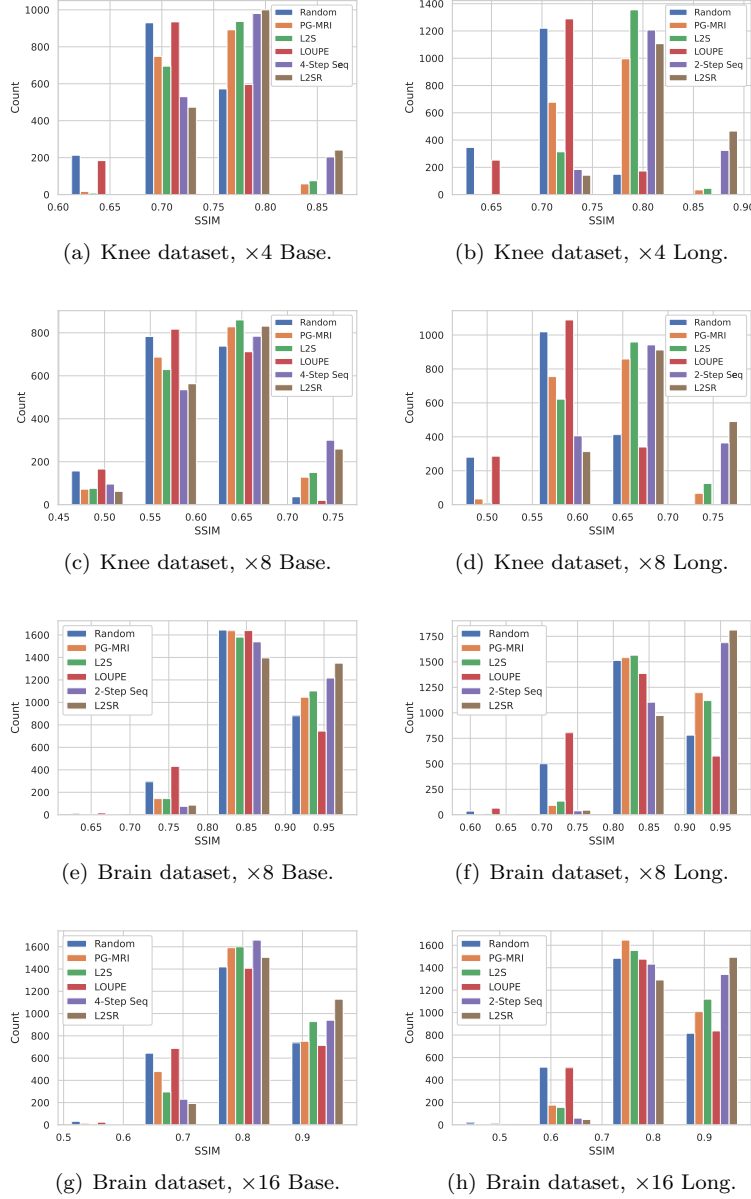


Figure 4. Histograms of SSIM values as shown in table 2. Each figure contains histograms of six methods: Random, PG-MRI, L2S, LOUPE, τ -Step Seq, L2SR.

In the ‘Joint Training’ setting, we evaluate the L2SR against two baseline methods: (1) LOUPE [24]: jointly training a learnable non-sequential sampler and a reconstructor; (2) τ -Step Seq. [32]: an advanced end-to-end sequential sampling and reconstruction method, where τ indicates the number of sampling steps, typically set to either 2 or 4 (and we report the best results obtained with either 2 or 4).

Additional details regarding all methods can be found in [Appendix C](#).

Table 3. Comparison of inference costs of different methods. We calculate the computational complexity of sampling and reconstruction required to an MRI scanning. C_π and $C_\mathcal{R}$ denote the computational complexity of taking one sample and one reconstruction individually. We also test the average inference time over knee dataset.

	Computational Complexity	Average Inference Time (s)			
		×4-accelration		×8-accelration	
		Base	Long	Base	Long
Random & LOUPE	$C_\mathcal{R}$			0.0040	
PG-MRI	$TC_\pi + (T + 1)C_\mathcal{R}$	0.0843	0.1423	0.0444	0.0727
Greedy Oracle	$T(N - \frac{T-1}{2})C_\mathcal{R}$	1.2134	2.0440	0.7071	1.1824
L2S & L2SR (Ours)	$TC_\pi + C_\mathcal{R}$	0.0201	0.0325	0.0126	0.0190
τ-Step Seq	$\tau C_\pi + (\tau + 1)C_\mathcal{R}$	0.0225($\tau = 2$), 0.0321($\tau = 4$)			

5.3. Main Results

5.3.1. Reconstruction Performance The reconstruction results of all compared accelerated MRI methods are quantified through SSIM and PSNR values, as detailed in table 2, with more granular histograms presented in figure 4. The outcomes can be understood in two key aspects. First, under the 'Fixed Reconstructor' scenario, L2S consistently outperforms other methods in most cases, signifying the importance of addressing the distributional mismatch issue in enhancing reconstruction performance. Secondly, L2SR emerges as the superior method in most cases across both settings. This underlines the effectiveness of solving the overall joint optimization problem compared to merely focusing on weakened forms or suboptimization issues.

5.3.2. Inference Complexity The computational complexities and corresponding inference times of trained models are presented in table 3. Both L2S and L2SR exhibit lower complexity and faster inference times compared to previous methods such as PG-MRI and Greedy Oracle. This empirically substantiates our claim that the proposed sparse-reward POMDP enhances efficiency by eliminating the need for intermediate reconstructions.

While the utilization of sparse rewards may raise concerns regarding increased training difficulty, the gains in inference efficiency make it a valuable tradeoff in practical applications, especially in the field of medical imaging. The focus of our experiments has been on assessing inference times, reflecting the emphasis on test efficiency in real-world deployment scenarios.

5.4. Ablations and Further Discussions

5.4.1. The Best Initial Acceleration Factor In medical imaging, the primary objective is often to attain the finest possible reconstruction for a pre-established acceleration factor. With this focus, the initial acceleration factor is treated as a tunable hyperparameter, with its optimal value determined through empirical analysis. Table 4 presents the SSIM values obtained with different initial acceleration factors. It shows that the proposed L2SR achieves the best reconstruction performance when the initial acceleration factor is established at $N/2$ for the knee dataset and $N/4$ for the brain dataset. It's worth noting that a larger initial acceleration factor grants the sampler additional degrees of freedom, leading to a superior optimal value of (4.6). However,

Table 4. Empirically searching the best acceleration factors in terms of SSIM values for L2SR. We consider different settings: $\times 4$ acceleration factor for knee dataset, $\times 8$ acceleration factor for knee dataset, $\times 8$ acceleration factor for brain dataset and $\times 16$ acceleration factor for brain dataset. The best results for each settings are shown in bold numbers.

initial acceleration factor	knee		brain	
	$\times 4$ -acceleration	$\times 8$ -acceleration	$\times 8$ -acceleration	$\times 16$ -acceleration
$N/1$	0.8055 ± 0.0327	0.6418 ± 0.0456	0.8862 ± 0.0469	0.8385 ± 0.069
$N/2$	0.8171 ± 0.0316	0.6712 ± 0.0511	0.8911 ± 0.0452	0.8456 ± 0.0652
$N/4$	0.8097 ± 0.0333	0.658 ± 0.0529	0.8969 ± 0.0417	0.8468 ± 0.0655
$N/8$	0.7999 ± 0.036	0.6332 ± 0.053	0.893 ± 0.043	0.8404 ± 0.066
$N/16$	0.7681 ± 0.0416	—	0.8899 ± 0.043	—

Table 5. Influence of discount factor to the proposed sparse-reward POMDP. We show SSIM values of L2S with respect to discount factors under $\times 4$ -acceleration on the knee test dataset.

	Base-horizon	Long-horizon
$\gamma = 0.5$	0.7129 ± 0.0372	0.7065 ± 0.0298
$\gamma = 0.9$	0.7528 ± 0.0363	0.7826 ± 0.0264
$\gamma = 1.0$	0.7543 ± 0.0372	0.7838 ± 0.0286

it also poses a challenge by making the training more complex due to the reduced initial information available.

5.4.2. Discount Factor In table 5, we present the impact of the discount factor on L2S. Our proposed L2S method achieves the best reconstruction performance when $\gamma = 1$. In contrast, previous dynamic sampling methods with dense-reward POMDP, as described in [12] and [11], have shown their optimal performance at discount factors of 0.5 and 0.9, respectively. The contrastive results make sense as the proposed sparse-reward POMDP is explicitly designed to get the reward at the end of the trajectory, thereby learning a long-sighted policy.

5.4.3. Number of Alternation The L2SR algorithm uses a fixed number of alternations as its stopping condition. The reconstruction performance in relation to the rounds of alternating training is depicted in figure 5. Typically, the algorithm converges within $L = 5$.

5.4.4. Adaptability Across Sampling Schemes A significant benefit of our framework lies in its extensibility across different sampling schemes. While the core of our experiments employs 1D vertical Cartesian sampling, the innovations central to our framework have broader applications. To demonstrate its adaptability, we have extended our method to radial sampling [2], where the selection changes from choosing a column to picking an angle in k-space. The core training process remains unchanged. The result in figure 6 confirms our method’s superiority over baselines within radial sampling schemes, which highlights the extensibility of our framework across various acquisition patterns.

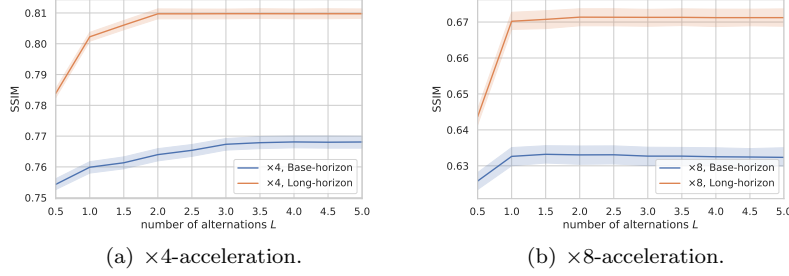


Figure 5. The choice of the number of alternation. We show SSIM values of L2SR with respect to different rounds of alternating training on the knee test dataset. Specifically, " $L = l.5$ " means the $(l + 1)$ th round of training the sampler with the learned reconstructor, i.e. solving (4.14).

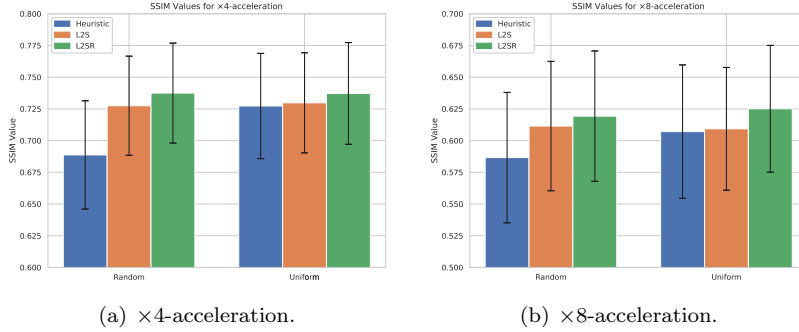


Figure 6. Comparison of SSIM values using the radial sampling pattern on the knee test dataset. The bars show the mean SSIM values, while the error bars indicate variances. We evaluate our L2S and L2SR methods against a pre-trained reconstructor baseline, comparing under two distinct initial heuristic sampling policies: Random and Uniform.

5.4.5. RL Algorithms We have experimented with PPO [39], an advanced RL algorithm, to learn a better sampling policy. However, we have found that PPO has slow convergence in practice and only slightly improves the SSIM values (around $0.001 \sim 0.002$). Therefore, we have decided not to use it in our proposed method.

5.5. Visualization

In figure 7, we present a sampling and reconstruction example of the knee dataset under $\times 4$ -acceleration, while in figure 8, we demonstrate the same for the brain dataset under $\times 8$ -acceleration. A comparison of these examples reveals that the proposed L2SR method produces images of superior visual quality. Notably, the images generated through L2SR exhibit fewer artifacts and bear a closer visual resemblance to the ground truth compared to those created by competing methods.

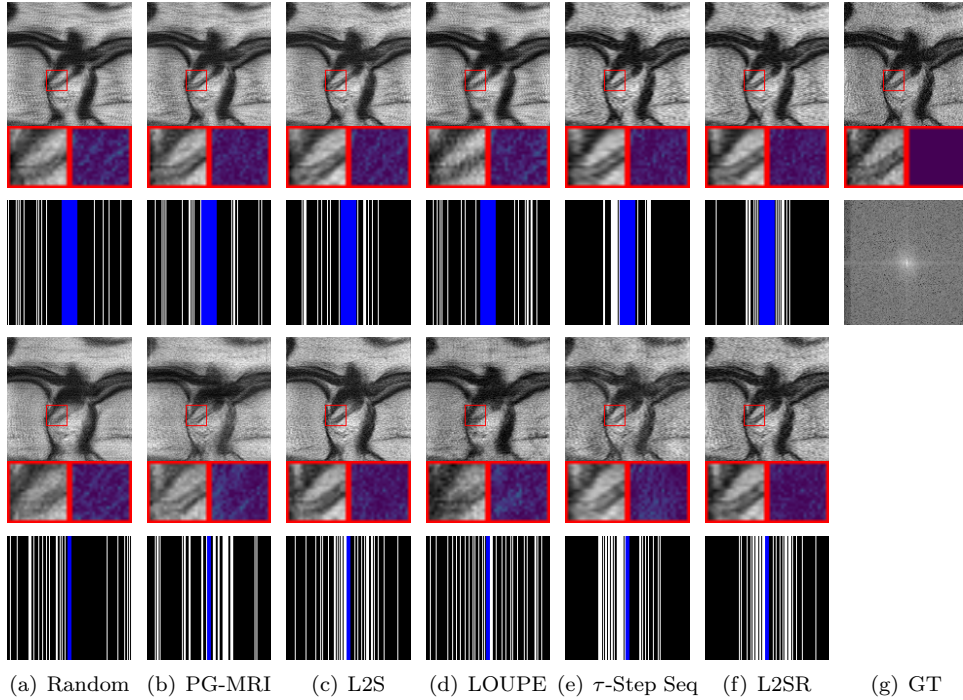


Figure 7. Visualisation of an example of the knee dataset for our methods and other competing methods under $\times 4$ -acceleration. From left to right, the columns display various accelerated MRI methods: Random, PG-MRI, L2S, LOUPE, τ -Step Seq, and L2SR, with the final column showing the ground truth (GT) and fully sampled k-space. From top to bottom, The rows represent: Base-horizon reconstruction images, Base-horizon masks, Long-horizon reconstruction images, and Long-horizon masks. An enlarged specific region is presented at the bottom left of each reconstruction image, along with its corresponding error map at the bottom right. The blue marks indicate measurements sampled from the low-frequency part of k-space in the first state.

6. Conclusion and Discussion

In this paper, we proposed a novel alternating training framework to jointly learn a personalized sampling policy and a corresponding reconstruction model for accelerated MRI. Specifically, we formulated the MRI sampling trajectory as a sparse-reward POMDP to learn a personalized sampler. Compared to existing dynamic sampling methods that utilize dense-reward POMDP, our proposed sparse-reward POMDP is more computationally efficient and avoids the distributional mismatch. Furthermore, the proposed framework, called L2SR, solve the joint optimization problem of learning a pair of samplers and reconstructors, thus eliminating the training mismatch.

Our empirical results on the fastMRI dataset demonstrate that the sparse-reward POMDP improves reconstruction quality by eliminating the distributional mismatch. Moreover, L2SR achieves the best acceleration-quality trade-off and inferences much faster than existing dynamic sampling methods. Overall, our proposed method provides a promising solution for accelerating MRI while maintaining high reconstruction quality.

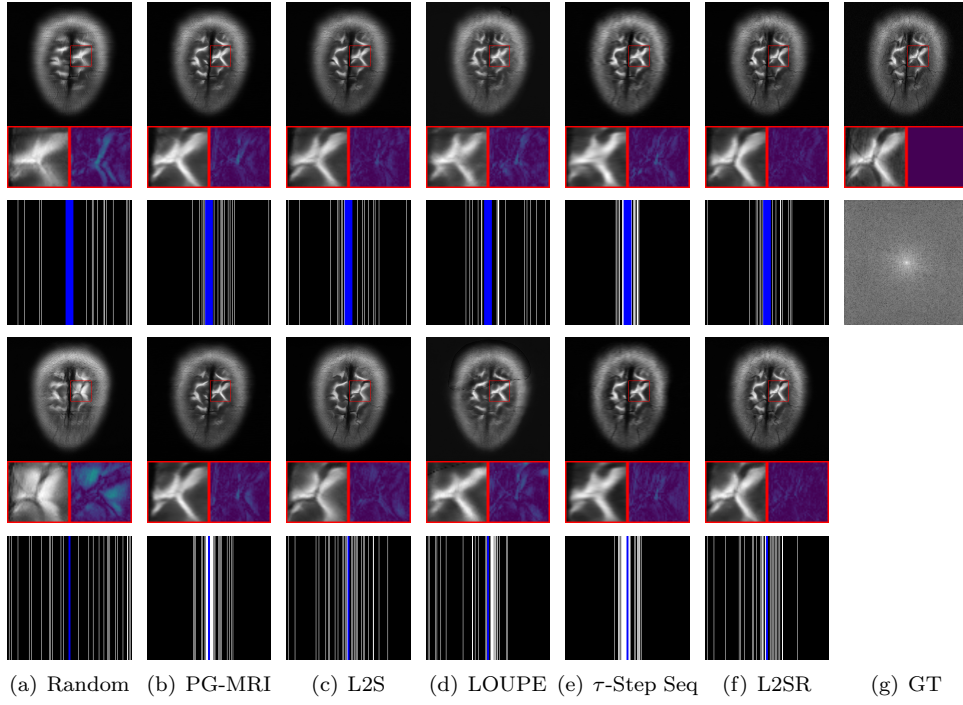


Figure 8. Visualisation of an example of the brain dataset for our methods and other competing methods under $\times 8$ -acceleration.

However, our work has two limitations. Firstly, the alternating training framework has no theoretical guarantee of convergence. Secondly, the sparse-reward POMDP requires an exploratory RL algorithm to solve (4.14). Our RL algorithm is not exploratory enough and is not applicable to longer sequences or larger action spaces. In future work, we plan to address these limitations and adapt our approach to realistic MRI scanning problems.

Data availability statement

The data that support the findings of this study are openly available at the following URL: <https://fastmri.med.nyu.edu/>.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 12090022.

Appendix A. Notation Index

We give a list of notation index and where it is defined in table A1.

Table A1. Notation index.

Noun	Notation	Reference
image width	N	section 3.1
ground truth	\mathbf{x}	section 3.1
2D Discrete Fourier Transform	\mathcal{F}	section 3.1
fully sampled measurement	\mathbf{y}	section 3.1
binary mask matrix	\mathbf{M}	section 3.1
observation	\mathbf{y}	section 3.1
reconstructor	\mathcal{R}	section 3.1
sampler	π	section 3.1
heuristic sampling policy	π_t^h	section 3.1
binary mask vector	\mathbf{a}	section 3.1
dense-reward POMDP		section 3.1
discount factor	γ	section 3.1
dense-reward joint optimization problem		(3.1)
N-dimensional probability simplex	Δ^N	(3.1)
reconstruction suboptimization problem		(3.2)
dense-reward suboptimization problem		(3.3)
sparse-reward POMDP		section 4.2
sparse-reward joint optimization problem		(4.6)
continuous function space	\mathcal{C}	theorem 1 and 2
sparse-reward suboptimization problem		(4.8)
Fixed Reconstructor		section 5.1
Joint Training		section 5.1
acceleration factor		section 5.1
initial acceleration factor		section 5.1
Base-horizon		section 5.1
Long-horizon		section 5.1

Appendix B. Supplements and Proofs for the Theoretic Analysis

Appendix B.1. Proof of theorem 1

We prove it with the idea of dynamic programming.

Proof Let $v_{\pi, \mathcal{R}}^{dense}(\mathbf{x}, \mathbf{M}_t)$ and $v_{\pi, \mathcal{R}}^{sparse}(\mathbf{x}, \mathbf{M}_t)$ be the value function of dense-reward POMDP and sparse-reward POMDP, defined as

$$v_{\pi, \mathcal{R}}^{dense}(\mathbf{x}, \mathbf{M}_t) = \mathbb{E}_{\pi} \sum_{s=t+1}^T r_s, \quad \text{where } \mathbf{y}_t = \mathbf{M}_t \odot \mathcal{F}(\mathbf{x}),$$

$$v_{\pi, \mathcal{R}}^{sparse}(\mathbf{x}, \mathbf{M}_t) = \mathbb{E}_{\pi} r_T, \quad \text{where } \mathbf{y}_t = \mathbf{M}_t \odot \mathcal{F}(\mathbf{x}).$$

We define $v_{\pi, \mathcal{R}}^{dense}(\mathcal{D}, \mathbf{M}_t) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} v_{\pi, \mathcal{R}}^{dense}(\mathbf{x}, \mathbf{M}_t)$ and $v_{\pi, \mathcal{R}}^{sparse}(\mathcal{D}, \mathbf{M}_t) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} v_{\pi, \mathcal{R}}^{sparse}(\mathbf{x}, \mathbf{M}_t)$. Indeed, $v_{\pi, \mathcal{R}}^{dense}(\mathcal{D}, \mathbf{M}_0)$ is the objective function of (3.1) and $v_{\pi, \mathcal{R}}^{sparse}(\mathcal{D}, \mathbf{M}_0)$ is the objective function of (4.6). We want to prove:

$$\sup_{\substack{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N) \\ \mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})}} v_{\pi, \mathcal{R}}^{dense}(\mathcal{D}, \mathbf{M}_0) \leq \sup_{\substack{\pi \in C(\mathbb{C}^{N \times N}, \Delta^N) \\ \mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})}} v_{\pi, \mathcal{R}}^{sparse}(\mathcal{D}, \mathbf{M}_0).$$

For $\forall \epsilon > 0$, there exists a pair of $\pi_\epsilon \in C(\mathbb{R}^{N \times N}, \Delta^N)$ and $\mathcal{R}_\epsilon \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})$ satisfying

$$\sup_{\substack{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N) \\ \mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})}} v_{\pi, \mathcal{R}}^{dense}(\mathcal{D}, \mathbf{M}_0) - v_{\pi_\epsilon, \mathcal{R}_\epsilon}^{dense}(\mathcal{D}, \mathbf{M}_0) < \epsilon.$$

We have the recursion formula for both $v_{\pi, \mathcal{R}}^{dense}(\mathbf{x}, \mathbf{M}_0)$ and $v_{\pi, \mathcal{R}}^{sparse}(\mathbf{x}, \mathbf{M}_0)$:

$$\begin{aligned} v_{\pi, \mathcal{R}}^{dense}(\mathbf{x}, M_t) &= \sum_{a_t=1}^N \pi(a_t \mid \mathbf{x}_t) \cdot v_{\pi, \mathcal{R}}^{dense}(\mathbf{x}, \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t})), \\ v_{\pi, \mathcal{R}}^{sparse}(\mathbf{x}, M_t) &= \sum_{a_t=1}^N \pi(a_t \mid \mathbf{y}_t) \cdot v_{\pi, \mathcal{R}}^{sparse}(\mathbf{x}, \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t})). \end{aligned}$$

Let $\pi_\epsilon^* \in C(\mathbb{C}^{N \times N}, \Delta^N)$ defined as

$$\pi_\epsilon^*(\cdot \mid \mathbf{y}) = \pi_\epsilon(\cdot \mid \mathcal{R}_\epsilon(\mathbf{y})), \quad \forall \mathbf{y} \in \mathbb{C}^{N \times N}.$$

Since the similarity metric function S is bounded by $[0, 1]$, $v_{\pi, \mathcal{R}} \in [0, 1]$. Recursively, we have

$$\begin{aligned} v_{\pi_\epsilon, \mathcal{R}_\epsilon}^{dense}(\mathcal{D}, \mathbf{M}_T) &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} S(\mathcal{R}_\epsilon(\mathbf{M}_T \odot \mathcal{F}(\mathbf{x})), \mathbf{x}) \\ &= v_{\pi_\epsilon^*, \mathcal{R}_\epsilon}^{sparse}(\mathcal{D}, \mathbf{M}_T), \end{aligned}$$

$$\begin{aligned} &v_{\pi_\epsilon, \mathcal{R}_\epsilon}^{dense}(\mathcal{D}, \mathbf{M}_{T-1}) \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{a_{T-1}=1}^n \pi_\epsilon(a_{T-1} \mid \mathbf{x}_{T-1}) \cdot v_{\pi_\epsilon, \mathcal{R}_\epsilon}^{dense}(\mathbf{x}, \mathbb{1}(\mathbf{M}_{T-1} + \mathbf{M}^{a_{T-1}})) \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{a_{T-1}=1}^n \pi_\epsilon^*(a_{T-1} \mid \mathbf{y}_{T-1}) \cdot v_{\pi_\epsilon, \mathcal{R}_\epsilon}^{dense}(\mathbf{x}, \mathbb{1}(\mathbf{M}_{T-1} + \mathbf{M}^{a_{T-1}})) \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{a_{T-1}=1}^n \pi_\epsilon^*(a_{T-1} \mid \mathbf{y}_{T-1}) \cdot v_{\pi_\epsilon^*, \mathcal{R}_\epsilon}^{sparse}(\mathbf{x}, \mathbb{1}(\mathbf{M}_{T-1} + \mathbf{M}^{a_{T-1}})) \\ &= v_{\pi_\epsilon^*, \mathcal{R}_\epsilon}^{sparse}(\mathcal{D}, \mathbf{M}_{T-1}), \end{aligned}$$

...

Finally, we have

$$v_{\pi_\epsilon}^{dense}(\mathcal{D}, \mathbf{M}_0) = v_{\pi_\epsilon^*}^{sparse}(\mathcal{D}, \mathbf{M}_0),$$

which means that for $\forall \epsilon$, there exists a pair of $\pi_\epsilon^* \in C(\mathbb{C}^{N \times N}, \Delta^N)$ and $\mathcal{R}_\epsilon \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})$ satisfying

$$\sup_{\substack{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N) \\ \mathcal{R} \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})}} v_{\pi, \mathcal{R}}^{dense}(\mathcal{D}, \mathbf{M}_0) < v_{\pi_\epsilon^*, \mathcal{R}_\epsilon}^{sparse}(\mathcal{D}, \mathbf{M}_0) + \epsilon.$$

Because of the arbitrariness of ϵ , the proof is completed!

Appendix B.2. Proof of theorem 2

Proof Without loss of generality, suppose that $\mathbf{M}_0 = \mathbf{0}$.

We denote

$$J(\mathbf{a}, \mathcal{R}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} S(\mathcal{R}(\mathbf{M}^{\mathbf{a}} \odot \mathcal{F}(\mathbf{x})), \mathbf{x}).$$

We first prove that for $\forall \|\mathbf{a}\|_1 = T$, we have

$$J(\mathbf{a}, \mathcal{R}^{dense}) \leq J(\mathbf{a}, \mathcal{R}^{sparse}),$$

by contradiction.

If it does not hold, there $\exists \|\mathbf{a}^*\|_1 = T$ that

$$J(\mathbf{a}^*, \mathcal{R}^{dense}) > J(\mathbf{a}^*, \mathcal{R}^{sparse}).$$

Then, we prove that it contradicts to the the maximum property of \mathcal{R}^{sparse} .

Notice that the input of \mathcal{R} is $\tilde{\mathbf{y}} = \mathbf{M}^{\mathbf{a}} \odot \mathbf{y}$, so there exists a well-defined function $\mathcal{A} : \mathbb{C}^{N \times N} \rightarrow \{0, 1\}^N, \tilde{\mathbf{y}} \rightarrow \mathbf{a}$. Specifically, values of \mathbf{a} at the indicators of zero columns of $\tilde{\mathbf{y}}$ equal to 0, and others equal to 1. We define $\mathcal{Y} : \mathbb{C}^{N \times N} \rightarrow \mathbb{R}^+, \tilde{\mathbf{y}} \rightarrow \min_{\mathcal{A}(\tilde{\mathbf{y}})(a_t)=1} \{\|\mathbf{M}^{a_t} \odot \tilde{\mathbf{y}}\|_2\}$ as the minimal norm of the non-zero column of $\tilde{\mathbf{y}}$.

We construct a new reconstructor on two disjoint closed subsets of $\mathbb{C}^{N \times N}$:

$$\mathcal{R}_\epsilon^{new}(\tilde{\mathbf{y}}) = \begin{cases} \mathcal{R}^{sparse}, & \|\mathcal{A}(\tilde{\mathbf{y}})\|_1 \leq T \text{ and } \mathcal{A}(\tilde{\mathbf{y}}) \neq \mathbf{a}^* \\ \mathcal{R}^{dense}, & \mathcal{A}(\tilde{\mathbf{y}}) = \mathbf{a}^* \text{ and } \mathcal{Y}(\tilde{\mathbf{y}}) \geq \epsilon \end{cases},$$

where $\epsilon > 0$ is pending. (We give a supplementary description for the convenience of understanding: the constrain of the second term $\mathcal{Y}(\tilde{\mathbf{y}}) \geq \epsilon$ is just for continuity.) Then, we further extend the domain of $\mathcal{R}_\epsilon^{new}$ to $\mathbb{C}^{N \times N}$ by Tietze extension theorem, denoted as $\mathcal{R}_\epsilon^{New} : \mathbb{C}^{N \times N} \rightarrow \mathbb{R}_\epsilon^{N \times N} \in C$.

Finally, we verify that $\mathcal{R}_\epsilon^{New}$ is better than \mathcal{R}^{sparse} , that is

$$\mathbb{E}_{\mathbf{a} \sim \pi_T^h} J(\mathbf{a}, \mathcal{R}^{sparse}) < \mathbb{E}_{\mathbf{a} \sim \pi_T^h} J(\mathbf{a}, \mathcal{R}_\epsilon^{New}),$$

which contradicts to the maximum property.

Since $\pi_T^h(\mathbf{a}^*) > 0$, we only need to verify that

$$J(\mathbf{a}^*, \mathcal{R}^{sparse}) < J(\mathbf{a}^*, \mathcal{R}_\epsilon^{New}).$$

Indeed, we choose a sufficiently small ϵ satisfying

$$\begin{aligned} & \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{1}_{\{\mathcal{Y}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}) \geq \epsilon\}} \cdot [S(\mathcal{R}^{dense}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) - S(\mathcal{R}^{sparse}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x)] \\ & > \frac{2}{3}(J(\mathbf{a}^*, \mathcal{R}^{dense}) - J(\mathbf{a}^*, \mathcal{R}^{sparse})), \end{aligned}$$

and

$$\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{1}_{\{\mathcal{Y}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}) < \epsilon\}} < \frac{1}{3}(J(\mathbf{a}^*, \mathcal{R}^{dense}) - J(\mathbf{a}^*, \mathcal{R}^{sparse})).$$

Then, we have

$$\begin{aligned}
& J(\mathbf{a}^*, \mathcal{R}_\epsilon^{\text{New}}) - J(\mathbf{a}^*, \mathcal{R}^{\text{sparse}}) \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[S(\mathcal{R}_\epsilon^{\text{New}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) - S(\mathcal{R}^{\text{sparse}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) \right] \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{1}_{\{\mathcal{Y}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}) \geq \epsilon\}} \cdot \left[S(\mathcal{R}^{\text{dense}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) - S(\mathcal{R}^{\text{sparse}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) \right] \\
&\quad + \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{1}_{\{\mathcal{Y}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}) < \epsilon\}} \cdot \left[S(\mathcal{R}_\epsilon^{\text{New}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) - S(\mathcal{R}^{\text{sparse}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) \right] \\
&\geq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{1}_{\{\mathcal{Y}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}) \geq \epsilon\}} \left[S(\mathcal{R}^{\text{dense}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) - S(\mathcal{R}^{\text{sparse}}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}), x) \right] \\
&\quad - \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{1}_{\{\mathcal{Y}(\mathbf{M}^{\mathbf{a}^*} \mathcal{F} \mathbf{x}) < \epsilon\}} \\
&> \left(\frac{2}{3} - \frac{1}{3} \right) (J(\mathbf{a}^*, \mathcal{R}^{\text{dense}}) - J(\mathbf{a}^*, \mathcal{R}^{\text{sparse}})) \\
&= \frac{1}{3} (J(\mathbf{a}^*, \mathcal{R}^{\text{dense}}) - J(\mathbf{a}^*, \mathcal{R}^{\text{sparse}})) \\
&> 0.
\end{aligned}$$

Therefore, it is contradicts to the maximum property of $\mathcal{R}^{\text{sparse}}$. Therefore, for $\forall \|\mathbf{a}\|_1 = T$,

$$J(\mathbf{a}, \mathcal{R}^{\text{dense}}) \leq J(\mathbf{a}, \mathcal{R}^{\text{sparse}}).$$

The rest proof is similar to the proof of theorem 1. Let $v_\pi^{\text{dense}}(\mathbf{x}, \mathbf{M}_t) = v_{\pi, \mathcal{R}^{\text{dense}}}^{\text{dense}}(\mathbf{x}, \mathbf{M}_t)$ and $v_\pi^{\text{sparse}}(\mathbf{x}, \mathbf{M}_t) = v_{\pi, \mathcal{R}^{\text{sparse}}}^{\text{sparse}}(\mathbf{x}, \mathbf{M}_t)$ as defined in theorem 1. We define $v_\pi^{\text{dense}}(\mathcal{D}, \mathbf{M}_t) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} v_\pi^{\text{dense}}(\mathbf{x}, \mathbf{M}_t)$ and $v_\pi^{\text{sparse}}(\mathcal{D}, \mathbf{M}_t) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} v_\pi^{\text{sparse}}(\mathbf{x}, \mathbf{M}_t)$. Indeed, $v_\pi^{\text{dense}}(\mathcal{D}, \mathbf{M}_0)$ is the objective function of (3.3) and $v_\pi^{\text{sparse}}(\mathcal{D}, \mathbf{M}_0)$ is the objective function of (4.8). We want to prove:

$$\sup_{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N)} v_\pi^{\text{dense}}(\mathcal{D}, \mathbf{M}_0) \leq \sup_{\pi \in C(\mathbb{C}^{N \times N}, \Delta^N)} v_\pi^{\text{sparse}}(\mathcal{D}, \mathbf{M}_0).$$

For $\forall \epsilon > 0$, there exists a sampler π_ϵ satisfying

$$\sup_{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N)} v_\pi^{\text{dense}}(\mathcal{D}, \mathbf{M}_0) - v_{\pi_\epsilon}^{\text{dense}}(\mathcal{D}, \mathbf{M}_0) < \epsilon.$$

Let $\pi_\epsilon^* \in C(\mathbb{C}^{N \times N}, \Delta^N)$ defined as

$$\pi_\epsilon^*(\cdot | \mathbf{y}) = \pi_\epsilon(\cdot | \mathcal{R}_\epsilon(\mathbf{y})), \quad \forall \mathbf{y} \in \mathbb{C}^{N \times N}.$$

We have the recursion formula for both $v_\pi^{\text{dense}}(\mathbf{x}, \mathbf{M}_0)$ and $v_\pi^{\text{sparse}}(\mathbf{x}, \mathbf{M}_0)$:

$$\begin{aligned}
v_\pi^{\text{dense}}(\mathbf{x}, M_t) &= \sum_{a_t=1}^N \pi(a_t | \mathbf{x}_t) \cdot v_\pi(\mathbf{x}, \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t})), \\
v_\pi^{\text{sparse}}(\mathbf{x}, M_t) &= \sum_{a_t=1}^N \pi(a_t | \mathbf{y}_t) \cdot v_{\pi, \mathcal{R}}(\mathbf{x}, \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t})).
\end{aligned}$$

Since the similarity metric function S is bounded by $[0, 1]$, $v_\pi \in [0, 1]$. Recursively, we have

$$\begin{aligned}
v_{\pi_\epsilon}^{\text{dense}}(\mathcal{D}, \mathbf{M}_T) &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} S(\mathcal{R}^{\text{dense}}(\mathbf{M}_T \odot \mathcal{F}(\mathbf{x})), \mathbf{x}) \\
&\leq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [S(\mathcal{R}^{\text{sparse}}(\mathbf{M}_T \odot \mathcal{F}(\mathbf{x})), \mathbf{x})] \\
&= v_{\pi_\epsilon^*}^{\text{sparse}}(\mathcal{D}, \mathbf{M}_T).
\end{aligned}$$

(The inequality is because $J(\mathbf{a}, \mathcal{R}^{\text{dense}}) \leq J(\mathbf{a}, \mathcal{R}^{\text{sparse}})$ where we set $\mathbf{M}_T = \mathbf{1} \cdot \mathbf{a}^T$)

$$\begin{aligned}
& v_{\pi_\epsilon}^{\text{dense}}(\mathcal{D}, \mathbf{M}_{T-1}) \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{a_{T-1}=1}^n \pi_\epsilon(a_t \mid \mathbf{x}_{T-1}) \cdot v_{\pi_\epsilon}^{\text{dense}}(\mathbf{x}, \mathbb{1}(\mathbf{M}_{T-1} + \mathbf{M}^{a_{T-1}})) \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{a_{T-1}=1}^n \pi_\epsilon^*(a_t \mid \mathbf{y}_{T-1}) \cdot v_{\pi_\epsilon}^{\text{dense}}(\mathbf{x}, \mathbb{1}(\mathbf{M}_{T-1} + \mathbf{M}^{a_{T-1}})) \\
&\leq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{a_t=1}^n \pi_\epsilon^*(a_t \mid \mathbf{y}_{T-1}) \cdot [v_{\pi_\epsilon^*}^{\text{sparse}}(\mathbf{x}, \mathbb{1}(\mathbf{M}_{T-1} + \mathbf{M}^{a_{T-1}}))] \\
&= v_{\pi_\epsilon^*}^{\text{sparse}}(\mathcal{D}, \mathbf{M}_{T-1}),
\end{aligned}$$

...

Finally, we have

$$v_{\pi_\epsilon}^{\text{dense}}(\mathcal{D}, \mathbf{M}_0) \leq v_{\pi_\epsilon^*}^{\text{sparse}}(\mathcal{D}, \mathbf{M}_0),$$

which means that for $\forall \epsilon$, there exists a pair of $\pi_\epsilon^* \in C(\mathbb{C}^{N \times N}, \Delta^N)$ and $\mathcal{R}_\epsilon \in C(\mathbb{C}^{N \times N}, \mathbb{R}^{N \times N})$ satisfying

$$\sup_{\pi \in C(\mathbb{R}^{N \times N}, \Delta^N)} v_{\pi, \mathcal{R}}^{\text{dense}}(\mathcal{D}, \mathbf{M}_0) < v_{\pi_\epsilon^*}^{\text{sparse}}(\mathcal{D}, \mathbf{M}_0) + \epsilon.$$

Because of the arbitrariness of ϵ , the statement holds!

□

Remark. We briefly show a counter-example of theorem 2 without the first assumption that ' $\pi_{T+\|\mathbf{M}_0\|_\infty}^h(\mathbf{a}) > 0$ for all binary column vectors \mathbf{a} satisfying $\|\mathbf{a}\|_1 = T + \|\mathbf{M}_0\|_\infty$ '. Similarly, we let $\mathbf{M}_0 = \mathbf{0}$. Suppose that there exists \mathbf{a}^* that $\|\mathbf{a}^*\|_1 = T$ which satisfies $\pi_T(\mathbf{a}^*) = 0$. Let dataset $\mathcal{D} = \{\mathbf{x}\}$, where $\mathbf{M}^{\mathbf{a}^*} \odot \mathcal{F}(\mathbf{x}) \neq \mathbf{0}$ and $\mathbf{M}^{1-\mathbf{a}^*} \odot \mathcal{F}(\mathbf{x}) > \mathbf{0}$. Then, let $\mathcal{R}^{\text{sparse}}$ be the solution of the optimization problem (4.9). Moreover, let $\mathcal{R}^{\text{dense}}(\mathbf{M}^{\mathbf{a}^*} \odot \mathcal{F}(\mathbf{x})) = \mathbf{x}$ and $\mathcal{R}^{\text{sparse}}(\mathbf{M}^{\mathbf{a}^*} \odot \mathcal{F}(\mathbf{x})) = \mathbf{0}$. The construction in the above proof guarantees the continuity of the two reconstructors. Therefore, that $\pi^{\text{dense}} = \mathbf{a}^*$ leads to the fact that $\mathcal{R}^{\text{dense}}$ can do reconstructs perfectly, while $\mathcal{R}^{\text{sparse}}$ can not.

Appendix B.3. Proof of proposition 1

Our proof mainly utilizes the fact that taking the derivative and taking the expectation can be interchanged.

Proof Here we consider the sparse-reward POMDP.

$$\begin{aligned}
\nabla_{\theta_{\mathcal{R}}} J_T^{\text{sparse}}(\mathbf{x}) &= \nabla_{\theta_{\mathcal{R}}} \mathbb{E}_{\{a_t\}_{t=0}^{T-1} \sim \pi} [\mathcal{S}(\mathcal{R}(\mathbf{y}_T; \theta_{\mathcal{R}}), \mathbf{x})] \\
&= \mathbb{E}_{\{a_t\}_{t=0}^{T-1} \sim \pi} [\nabla_{\theta_{\mathcal{R}}} \mathcal{S}(\mathcal{R}(\mathbf{y}_T; \theta_{\mathcal{R}}), \mathbf{x})].
\end{aligned}$$

The second equality arises from interchanging the differentiation and expectation order. This process is justified as: firstly, the differentiation concerning $\theta_{\mathcal{R}}$ is π -independent; secondly, π represents a discrete distribution, making the expectation equivalent to a finite sum. These conditions enable the application of the Dominated Convergence Theorem, sanctioning our operation switch, hence confirming the second equality.

□

Appendix B.4. Alternate Training for Dense-reward POMDP Fails

We now demonstrate that the proposed alternate training framework is not suitable for the dense-reward POMDP. We start by calculating the derivative of J^{dense} w.r.t the $\theta_{\mathcal{R}}$.

Proposition 2 *The derivative of J^{dense} w.r.t. $\theta_{\mathcal{R}}$ is*

$$\begin{aligned} \nabla_{\theta_{\mathcal{R}}} J_T^{\text{dense}}(\mathbf{x}) &= \sum_{t=0}^{T-1} \mathbb{E}_{\{a_s\}_{s=0}^{t-1} \sim \pi} \left[\sum_{a_t} \nabla_{\theta_{\mathcal{R}}} \pi(a_t \mid \mathcal{R}(\mathbf{y}_t; \theta_{\mathcal{R}})) \cdot q_t \right] \\ &\quad + \mathbb{E}_{\{a_t\}_{t=0}^{T-1} \sim \pi} [\nabla_{\theta_{\mathcal{R}}} S(\mathcal{R}(\mathbf{y}_T; \theta_{\mathcal{R}}), \mathbf{x})], \end{aligned} \quad (\text{B.1})$$

where $\{a_s\}_{s=0}^{t-1} \sim \pi$ means a sequential acquisition according to (3.1), $\mathbf{y}_t = \mathbf{y}_t(\mathbf{M}_0, \{a_s\}_{s=0}^{t-1}) = (\mathbf{M}_0 + \sum_{s=0}^{t-1} \mathbf{M}^{a_s}) \odot \mathbf{y}$, and $q_t = q(\mathbf{x}, \mathbf{M}_t, a_t)$ is the Q -function of the dense-reward POMDP for $t = 1, \dots, T$.

Proof Here we consider the dense-reward POMDP. Let $v(\mathbf{x}, \mathbf{M}_t)$ be the value function and $q(\mathbf{x}_t, a_t)$ be the Q function at timestep t , defined as

$$v(\mathbf{x}, \mathbf{M}_t) = \mathbb{E}_{\{a_s\}_{s=t}^{T-1} \sim \pi} \sum_{s=t+1}^T r_s, \quad \text{where } \mathbf{y}_t = \mathbf{M}_t \odot \mathcal{F}(\mathbf{x}),$$

$$q(\mathbf{x}, \mathbf{M}_t, a_t) = r_{t+1} + v(\mathbf{x}, \mathbf{M}_{t+1}), \quad \text{where } \mathbf{M}_{t+1} = \mathbb{1}(\mathbf{M}_t + \mathbf{M}^{a_t}).$$

Indeed,

$$J_T^{\text{dense}}(\mathbf{x}) = v(\mathbf{x}, \mathbf{M}_0) + S(\mathcal{R}(\mathbf{y}_0), \mathbf{x}).$$

For $t = 0, 1, \dots, T-1$, we have

$$\begin{aligned} &\nabla_{\theta_{\mathcal{R}}} v(\mathbf{x}, \mathbf{M}_t) \\ &= \nabla_{\theta_{\mathcal{R}}} \left[\sum_{a_t} \pi(a_t \mid \mathbf{x}_t) \cdot q(\mathbf{x}, \mathbf{M}_t, a_t) \right] \\ &= \sum_{a_t} [\nabla_{\theta_{\mathcal{R}}} \pi \cdot q_t + \pi \cdot \nabla_{\theta_{\mathcal{R}}} q_t] \\ &= \sum_{a_t} [\nabla_{\theta_{\mathcal{R}}} \pi \cdot q_t + \pi \cdot \nabla_{\theta_{\mathcal{R}}} (r_{t+1} + v(\mathbf{x}, \mathbf{M}_{t+1}))] \\ &= \sum_{a_t} [\nabla_{\theta_{\mathcal{R}}} \pi \cdot q_t + \pi \cdot \nabla_{\theta_{\mathcal{R}}} (S(\mathcal{R}(\mathbf{y}_{t+1}), \mathbf{x}) + v(\mathbf{x}, \mathbf{M}_{t+1}))] - \nabla_{\theta_{\mathcal{R}}} S(\mathcal{R}(\mathbf{y}_t), \mathbf{x}), \end{aligned}$$

and

$$\nabla_{\theta_{\mathcal{R}}} v(\mathbf{x}, \mathbf{M}_T) = 0.$$

Finally, we have

$$\begin{aligned}
& \nabla_{\theta_{\mathcal{R}}} J_T^{\text{dense}}(\mathbf{x}) \\
&= \sum_{t=0}^{T-1} \sum_{a_0, \dots, a_{t-1}} \Pr_{\pi}(\mathbf{y}_0 \xrightarrow{\{a_s\}_{s=0}^{t-1}} \mathbf{y}_t) \cdot \left[\sum_{a_t} \nabla_{\theta_{\mathcal{R}}} \pi(a_t \mid \mathcal{R}(\mathbf{y}_t)) \cdot q_t \right] \\
&\quad + \sum_{a_0, \dots, a_{T-1}} \Pr_{\pi}(\mathbf{y}_0 \xrightarrow{\{a_t\}_{t=0}^{T-1}} \mathbf{y}_T) \cdot \nabla_{\theta_{\mathcal{R}}} S(\mathbf{x}_T, \mathbf{x}) \\
&= \sum_{t=0}^{T-1} \mathbb{E}_{\{a_s\}_{s=0}^{t-1} \sim \pi} \left[\sum_{a_t} \nabla_{\theta_{\mathcal{R}}} \pi(a_t \mid \mathcal{R}(\mathbf{y}_t)) \cdot q_t \right] + \mathbb{E}_{\{a_t\}_{t=0}^{T-1} \sim \pi} [\nabla_{\theta_{\mathcal{R}}} S(\mathcal{R}(\mathbf{y}_T), \mathbf{x})].
\end{aligned}$$

□

Although proposition 2 gives the derivative of dense-reward objective function w.r.t parameters of the reconstructor, we empirically find that training reconstructor with the derivative by gradient-based methods fails. The reason may be that the sequential acquisition $\{a_t\}_{t=0}^{t-1} \sim \pi$ depends on not only π but also \mathcal{R} . Changing the reconstructor results to an unpredictable change of the sampling strategy, leading to the failure. From another perspective, the reconstructor can also be viewed as an RL agent, so gradient-based methods hardly work.

Appendix C. Implementation Details

In this section, We will show the implementation details of all methods. In section Appendix C.1, we show the basic settings of all algorithms. In section Appendix C.2, we show the implementation details of L2S and L2SR. In section Appendix C.3, we show the implementation details of competing methods.

Appendix C.1. Basic Settings

Both our methods and competing methods, are performed with the same basic settings.

SSIM Values. SSIM hyperparameters are kept to their original values in [35]. The dynamic range is chosen to be the maximum pixel value in the corresponding ground truth volume.

Loss Function. Since the proposed sparse-reward POMDP uses SSIM values as the similarity metric, for consistency, all algorithms use negative SSIM values as the loss.

Appendix C.2. Our Methods

Pre-training the reconstructor is the same as the competing method ‘Random’, as detailed in the following subsection. We implement the A2C algorithm based on the stable-baselines3 [40] which is a popular framework for reliable implementations of RL algorithms. The learning rate is set to 0.0003, the update timestep is set to T , and the other hyperparameters are to the default values. The reconstructor is trained by Adam optimizer for 10 epochs in the alternating training stage. The learning rate for both θ_{π} and $\theta_{\mathcal{R}}$ decays at a rate of 3 at each alternate training. We refer you to our official code for more details.

Appendix C.3. Competing Methods

Random. We train the reconstructor with random policy by Adam [33] optimizer for 50 epochs. Other hyperparameters inherit the fastMRI repository. We always use early stopping to get the best model on the validation dataset.

PG-MRI. We use the best performing $\gamma = 0.9$ Non-Greedy method in [11]. We train the sampler with its default hyperparameters. The fixed reconstructor is pre-trained by "Random" heuristic sampling policy. Notice that the reconstructor pre-trained is different from [11], since the loss is negative SSIM instead of L_1 -norm and the training dataset is half of the volumes instead of full. In addition, we use early stopping based on the validation set rather than training for the full 50 epochs for better reconstruction performances.

Greedy Oracle. The one-step greedy oracle policy has access to the ground truth for reference. We choose the action that increases similarity the most at test time with a fixed reconstructor the same as above.

LOUPE. This method jointly trains the parametric probability mask \mathbf{p} and the reconstructor end-to-end via BP without pre-training. We use the implementation by [32] and train with its default hyperparameters.

τ -Step Seq. This method jointly trains the sampler and the reconstructor end-to-end via BP without pre-training. We use the default hyperparameters in [32].

References

- [1] Candès E J, Romberg J and Tao T 2006 *IEEE Transactions on information theory* **52** 489–509
- [2] Lustig M, Donoho D and Pauly J M 2007 *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* **58** 1182–1195
- [3] Lustig M, Donoho D L, Santos J M and Pauly J M 2008 *IEEE signal processing magazine* **25** 72–82
- [4] Lee D, Yoo J, Tak S and Ye J C 2018 *IEEE Transactions on Biomedical Engineering* **65** 1985–1995
- [5] Hyun C M, Kim H P, Lee S M, Lee S and Seo J K 2018 *Physics in Medicine & Biology* **63** 135007
- [6] Yang G, Yu S, Dong H, Slabaugh G, Dragotti P L, Ye X, Liu F, Arridge S, Keegan J, Guo Y and Firmin D 2018 *IEEE Transactions on Medical Imaging* **37** 1310–1321
- [7] Quan T M, Nguyen-Duc T and Jeong W K 2018 *IEEE Transactions on Medical Imaging* **37** 1488–1497
- [8] yang y, Sun J, Li H and Xu Z 2016 Deep admm-net for compressive sensing mri *Advances in Neural Information Processing Systems* vol 29 ed Lee D, Sugiyama M, Luxburg U, Guyon I and Garnett R (Curran Associates, Inc.)
- [9] Hammernik K, Klatzer T, Kobler E, Recht M P, Sodickson D K, Pock T and Knoll F 2018 *Magnetic resonance in medicine* **79** 3055–3071
- [10] Zhang J and Ghanem B 2018 Ista-net: Interpretable optimization-inspired deep network for image compressive sensing *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
- [11] Bakker T, van Hoof H and Welling M 2020 Experimental design for mri by greedy policy search *Advances in Neural Information Processing Systems* vol 33 ed Larochelle H, Ranzato M, Hadsell R, Balcan M and Lin H (Curran Associates, Inc.) pp 18954–18966
- [12] Pineda L, Basu S, Romero A, Calandra R and Drozdal M 2020 Active mr k-space sampling with reinforcement learning *International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer)* pp 23–33
- [13] Liu Y, Pang Y, Jin R and Wang Z 2022 *ArXiv* **abs/2203.05756**
- [14] Zbontar J, Knoll F, Sriram A, Murrell T, Huang Z, Muckley M J, Defazio A, Stern R, Johnson P, Bruno M *et al.* 2018 *arXiv preprint arXiv:1811.08839*
- [15] Vasanawala S, Murphy M, Alley M T, Lai P, Keutzer K, Pauly J M and Lustig M 2011 Practical parallel imaging compressed sensing mri: Summary of two years of experience in accelerating

- body mri of pediatric patients *2011 ieee international symposium on biomedical imaging: From nano to macro* (IEEE) pp 1039–1043
- [16] Chauffert N, Ciuciu P, Kahn J and Weiss P 2014 *SIAM Journal on Imaging Sciences* **7** 1962–1992
 - [17] Rudin L I, Osher S and Fatemi E 1992 *Physica D: nonlinear phenomena* **60** 259–268
 - [18] Chan R H, Chan T F, Shen L and Shen Z 2003 *SIAM Journal on Scientific Computing* **24** 1408–1432
 - [19] Daubechies I, Defrise M and De Mol C 2004 *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **57** 1413–1457
 - [20] Daubechies I, DeVore R, Fornasier M and Güntürk C S 2010 *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **63** 1–38
 - [21] Shen Z, Wang Y, Wu D, Yang X and Dong B 2022 *Inverse Problems and Imaging* **16** 179–195
 - [22] Van Hasselt H, Guez A and Silver D 2016 Deep reinforcement learning with double q-learning *Proceedings of the AAAI conference on artificial intelligence* vol 30
 - [23] Baxter J and Bartlett P L 2001 *Journal of Artificial Intelligence Research* **15** 319–350
 - [24] Bahadir C D, Dalca A V and Sabuncu M R 2019 Learning-based optimization of the under-sampling pattern in mri *International Conference on Information Processing in Medical Imaging* (Springer) pp 780–792
 - [25] Weiss T, Vedula S, Senouf O, Michailovich O, Zibulevsky M and Bronstein A 2020 Joint learning of cartesian under sampling andre construction for accelerated mri *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* pp 8653–8657
 - [26] Zhang J, Zhang H, Wang A, Zhang Q, Sabuncu M, Spincemaille P, Nguyen T D and Wang Y 2020 Extending loupe for k-space under-sampling pattern optimization in multi-coil mri *International Workshop on Machine Learning for Medical Image Reconstruction* (Springer) pp 91–101
 - [27] Sun H, Dalca A V and Bouman K L 2020 Learning a probabilistic strategy for computational imaging sensor selection *2020 IEEE International Conference on Computational Photography (ICCP)* pp 1–12
 - [28] Wang G, Luo T, Nielsen J F, Noll D C and Fessler J A 2021 *arXiv preprint arXiv:2101.11369*
 - [29] Jin K H, Unser M and Yi K M 2019 *arXiv preprint arXiv:1901.04547*
 - [30] Zhang Z, Romero A, Muckley M J, Vincent P, Yang L and Drozdal M 2019 Reducing uncertainty in undersampled mri reconstruction with active acquisition *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp 2049–2058
 - [31] Van Gorp H, Huijben I, Veeling B S, Pezzotti N and Van Sloun R J 2021 Active deep probabilistic subsampling *International Conference on Machine Learning* (PMLR) pp 10509–10518
 - [32] Yin T, Wu Z, Sun H, Dalca A V, Yue Y and Bouman K L 2021 *arXiv preprint arXiv:2105.06460*
 - [33] Kingma D P and Ba J 2014 *arXiv preprint arXiv:1412.6980*
 - [34] Mnih V, Badia A P, Mirza M, Graves A, Lillicrap T, Harley T, Silver D and Kavukcuoglu K 2016 Asynchronous methods for deep reinforcement learning *International conference on machine learning* (PMLR) pp 1928–1937
 - [35] Wang Z, Bovik A, Sheikh H and Simoncelli E 2004 *IEEE Transactions on Image Processing* **13** 600–612
 - [36] Arjovsky M, Chintala S and Bottou L 2017 Wasserstein generative adversarial networks *International conference on machine learning* (PMLR) pp 214–223
 - [37] Huang Z, Heng W and Zhou S 2019 Learning to paint with model-based deep reinforcement learning *Proceedings of the IEEE/CVF international conference on computer vision* pp 8709–8718
 - [38] Ronneberger O, Fischer P and Brox T 2015 U-net: Convolutional networks for biomedical image segmentation *International Conference on Medical image computing and computer-assisted intervention* (Springer) pp 234–241
 - [39] Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O 2017 *arXiv preprint arXiv:1707.06347*
 - [40] Raffin A, Hill A, Gleave A, Kanervisto A, Ernestus M and Dormann N 2021 *Journal of Machine Learning Research* **22** 1–8 URL <http://jmlr.org/papers/v22/20-1364.html>