

Article

Not peer-reviewed version

Alternating Method of Successive Approximations

[Yunzhu Zhang](#) *

Posted Date: 29 April 2025

doi: 10.20944/preprints202504.2437.v1

Keywords: Bilevel optimization; Meta-learning; Optimal control



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Alternating Method of Successive Approximations

Yunzhu Zhang

Independent Researcher; vivienbian624@outlook.com

Abstract: In this work, we propose a principled deep learning framework for solving inverse problems by casting them as optimal control problems. Building upon variational models, we formulate the reconstruction task as the minimization of an energy functional that combines a data fidelity term with a learnable regularization parameterized by deep neural networks. To solve the resulting nonconvex and nonsmooth optimization problem, we employ a gradient flow approach, leading to a continuous-time dynamical system. Learning the network parameters is further structured as an optimal control problem, where the parameters act as controls to minimize a terminal cost and an integrated running cost. We adopt the Method of Successive Approximations (MSA), a theoretically grounded algorithm inspired by the Pontryagin Maximum Principle, to iteratively solve the control problem. Each iteration alternates between solving a forward state equation and a backward adjoint equation, followed by updating the parameters via Hamiltonian maximization. We show that when gradient ascent is used for the Hamiltonian step, the MSA framework recovers classical back-propagation. Moreover, we discuss the computational challenges associated with MSA, particularly the linear memory growth with respect to temporal discretization, and outline potential strategies for memory-efficient implementation. Numerical results in sparse-view CT and accelerated MRI reconstruction demonstrate the effectiveness and robustness of the proposed method, offering a theoretically interpretable and practically scalable alternative to traditional deep learning-based reconstruction techniques.

Keywords: bilevel optimization; meta-learning; optimal control

1. Solving Inverse Problems using Deep Learning

Deep learning (DL) techniques have witnessed remarkable progress in recent years, especially in the field of image processing, where they have consistently demonstrated significant improvements over traditional methods. Beyond image processing, DL and advanced optimization strategies have also found impactful applications in a range of complex domains such as wireless communication systems, quantum networking, and sensor network optimization, enabling solutions to previously intractable problems [1,2].

In the area of medical imaging, particularly for computed tomography (CT) and magnetic resonance imaging (MRI) reconstruction, one of the most successful DL-based strategies is known as algorithm unrolling or unfolding. This approach draws inspiration from classical iterative optimization algorithms, such as proximal gradient descent commonly used in variational methods. Instead of relying on handcrafted regularization terms designed through expert knowledge, unrolling methods integrate deep neural networks into the iterative framework. These networks are trained to learn optimal feature representations either from images or directly from sinogram data (the raw projection measurements), thereby enhancing the reconstruction process by capturing complex structures and noise patterns that traditional regularizers might miss.

More recently, dual-domain methods have gained attention, offering a further step forward by simultaneously utilizing complementary information from both the image domain and the sinogram (projection) domain. By bridging these two domains, these methods aim to improve reconstruction accuracy, resolution, and artifact suppression beyond what single-domain approaches can achieve.

Despite these exciting advances, DL-based reconstruction methods are not without limitations. One major challenge is their relatively weak theoretical foundation. Unlike classical optimization

techniques that come with rigorous convergence and stability proofs, many DL models offer limited guarantees regarding their behavior during inference, particularly under distribution shifts or in clinical settings where robustness is critical. Moreover, these models are often memory-intensive, requiring substantial computational resources for training and inference, which can be a barrier to widespread adoption. Another persistent issue is the risk of overfitting, as the models may superficially imitate optimization procedures without truly capturing the underlying physics of the reconstruction problem.

Additionally, convolutional neural networks (CNNs), a cornerstone of deep learning, have demonstrated strong performance in several key tasks within medical imaging. They have been effectively employed for reconstruction from sparse-view and low-dose datasets, projection domain synthesis to enhance missing information, post-processing of preliminary reconstructions to refine image quality, and in the integration of learned priors into traditional iterative algorithms. In many cases, these DL-enhanced approaches have outperformed conventional analytical reconstruction methods, offering improvements in image quality, noise reduction, and diagnostic reliability.

In recent years, a new and increasingly influential class of deep learning (DL)-based methods, known as learnable optimization algorithms (LOAs), has been developed for image reconstruction tasks. Unlike earlier DL approaches, LOAs are designed with mathematical justifications and formal convergence guarantees, addressing some of the critical theoretical concerns traditionally associated with DL-based inverse problem solving. These methods have demonstrated significant advancements, particularly in areas such as MRI reconstruction [3].

LOAs are fundamentally rooted in the variational framework commonly used in classical inverse problem formulations [4]. In these methods, the regularization term—historically crafted manually based on prior knowledge—is instead parameterized using deep neural networks with learnable parameters. This transition allows the regularizer to capture much more complex and realistic image features, although it introduces challenges because the resulting objective function is often nonconvex and nonsmooth. Despite this complexity, LOAs aim to develop efficient, data-driven optimization schemes that ensure convergence towards a solution, typically by embedding algorithmic structures such as proximal gradient descent, half-quadratic splitting, or iterative shrinkage within the network design.

A distinctive characteristic of LOAs is that the network architecture closely mirrors the steps of an optimization algorithm: each layer corresponds to a logical iteration, and the entire deep network is highly structured according to the underlying optimization principles. Crucially, while the deep network's parameters (such as proximal operators or penalty weights) are learned from data, the theoretical convergence properties—such as stability, robustness, and rates of convergence—are rigorously preserved, differentiating LOAs from more heuristic deep learning methods.

This innovative framework has been successfully employed to tackle practical challenges in medical imaging. For example, in sparse-view CT reconstruction, where the number of projection angles is limited to reduce radiation exposure, LOAs can adaptively learn regularization functions that suppress artifacts while preserving fine anatomical details. Similarly, in MRI reconstruction, LOAs have enabled faster imaging protocols by reconstructing high-quality images from significantly under-sampled k-space data.

Overall, learnable optimization algorithms represent a promising direction for combining the flexibility and expressive power of deep learning with the rigorous guarantees of traditional optimization theory, opening new opportunities for reliable and efficient image reconstruction across a wide range of applications.

Building upon prior work that established an optimal control framework for solving inverse problems with deep learnable regularizers [4], the current study extends this direction by integrating bilevel optimization with meta-learning strategies to further enhance adaptability and generalization across diverse tasks. By combining the dynamics of optimal control with meta-learned regularizers, our approach enables task-conditioned optimization that dynamically adjusts to different reconstruction

scenarios. This synthesis offers a powerful strategy for improving both the theoretical rigor and practical performance of learning-based inverse problem solvers.

2. Optimal Control Viewpoint in Deep Learning

In this work, we propose an optimal control framework for solving inverse problems by minimizing a variational energy model with learned parameters. Specifically, we consider the following objective functional:

$$\mathcal{E}(\mathbf{u}; \theta) = \mathcal{D}(\mathbf{u}, \mathbf{b}) + \lambda \mathcal{R}(\mathbf{u}; \theta), \quad (1)$$

where:

$\mathcal{D}(\mathbf{u}, \mathbf{b})$ is the data fidelity term, enforcing consistency between the reconstructed signal \mathbf{u} and the observed measurements \mathbf{b} ;

$\mathcal{R}(\mathbf{u}; \theta)$ is a regularization functional incorporating prior knowledge about \mathbf{u} , parameterized by a deep neural network (DNN) with learnable parameters θ ;

$\lambda > 0$ balances the trade-off between data fidelity and regularization.

For linear inverse problems, the data fidelity is typically quadratic:

$$\mathcal{D}(\mathbf{u}, \mathbf{b}) = \frac{1}{2} \|A\mathbf{u} - \mathbf{b}\|_2^2, \quad (2)$$

where A is a known measurement operator (e.g., a forward model for CT or MRI systems).

To minimize the variational energy $\mathcal{E}(\mathbf{u}; \theta)$, we utilize the gradient flow dynamics, leading to the following ordinary differential equation (ODE) [5]:

$$\frac{d\mathbf{u}(t)}{dt} = f(\mathbf{u}(t), \theta) := -\nabla_{\mathbf{u}} \mathcal{E}(\mathbf{u}(t); \theta), \quad t \in [0, T], \quad (3)$$

where $T > 0$ is a predefined terminal time. Here, $f(\mathbf{u}(t), \theta)$ represents the negative gradient of the energy functional at time t .

The goal is to learn the optimal parameters θ by casting the learning process as an optimal control problem (denoted as (P)), formulated as:

$$\min_{\theta \in \Theta} J(\theta) := \Phi(\mathbf{u}(T)) + \int_0^T r(\mathbf{u}(t), \theta) dt, \quad (4)$$

subject to the dynamical system constraint:

$$\begin{cases} \dot{\mathbf{u}}(t) = f(\mathbf{u}(t), \theta), & t \in [0, T], \\ \mathbf{u}(0) = \mathbf{u}^0, \end{cases} \quad (5)$$

where:

$\Phi(\mathbf{u}(T))$ is the terminal cost, typically measuring the mismatch between the reconstructed final state and the ground truth;

$r(\mathbf{u}(t), \theta)$ is a running cost (often acting as a regularization or penalization term during the evolution);

Θ denotes the admissible set for control parameters;

\mathbf{u}^0 represents the given initial condition.

The optimality conditions associated with this problem can be characterized by introducing a Lagrangian and deriving the corresponding adjoint state equations [6]. Define the Lagrangian \mathcal{L} :

$$\mathcal{L}(\mathbf{u}, \mathbf{p}, \theta) = \Phi(\mathbf{u}(T)) + \int_0^T (r(\mathbf{u}(t), \theta) + \langle \mathbf{p}(t), \dot{\mathbf{u}}(t) - f(\mathbf{u}(t), \theta) \rangle) dt, \quad (6)$$

where $\mathbf{p}(t)$ is the adjoint state.

Taking variations with respect to \mathbf{u} and θ and setting them to zero leads to the following optimality system:

Minimizing the variational model (1) with the gradient flow method [7], we obtain the following ODE

$$\dot{\mathbf{u}}(t) = f(\mathbf{u}(t), \theta) := -\nabla_{\mathbf{u}} \mathcal{E}(\mathbf{u}(t); \theta), \quad t \in [0, T]. \quad (7)$$

We can cast the process of learning the undetermined parameters as an optimal control problem (P) with control parameters θ :

$$\begin{aligned} \min_{\theta \in \Theta} \quad & J(\theta) := \Phi(\mathbf{u}(T)) + \int_0^T r(\mathbf{u}(t), \theta) dt, \quad (\text{P}) \\ \text{s.t.} \quad & \begin{cases} \dot{\mathbf{u}}(t) = f(\mathbf{u}(t), \theta), & 0 \leq t \leq T, \\ \mathbf{u}(0) = \mathbf{u}^0, \end{cases} \quad (\text{ODE}) \end{aligned}$$

where $\Phi(\mathbf{u}(T))$ is the terminal cost and $r(\mathbf{u}(t), \theta)$ is the running cost which usually plays role as the regularizer. Besides, Θ is the admissible set for the control parameters θ and \mathbf{u}^0 is the initial. These expressions enable the use of gradient-based optimization algorithms (e.g., stochastic gradient descent, Adam, or specialized adjoint-based methods) to update the network parameters θ efficiently while preserving the dynamics and structure of the underlying variational model.

Thus, our proposed framework unifies data-driven learning with the rigorous structure of variational optimization and control theory, providing a principled and theoretically grounded approach to solve challenging inverse problems such as sparse-view CT and accelerated MRI reconstruction.

3. Pontryagin's Maximum Principle

First we give the definition of the Hamiltonian function $H : \mathbb{R}^n \times \mathbb{R}^n \times \Theta \rightarrow \mathbb{R}$ by

$$H(\mathbf{u}; \mathbf{p}; \theta) = \mathbf{p}^\top f(\mathbf{u}, \theta) - r(\mathbf{u}, \theta), \quad (8)$$

based on which we state the Pontryagin's Maximum Principle (PMP) below

Theorem 1 (Pontryagin's Maximum Principle, Informal Statement). *If θ^* optimizes the optimal control problem (P), and $\mathbf{u}^*(t)$ is its corresponding state trajectory. Then, there exists an absolutely continuous co-state process $\mathbf{p}^*(t), t \in [0, T]$ such that*

$$\begin{cases} \dot{\mathbf{u}}^*(t) = \nabla_{\mathbf{p}} H(\mathbf{u}^*(t); \mathbf{p}^*(t); \theta^*), \quad \mathbf{u}^*(0) = \mathbf{u}_0, & (\text{ODE}) \\ \dot{\mathbf{p}}^*(t) = -\nabla_{\mathbf{u}} H(\mathbf{u}^*(t); \mathbf{p}^*(t); \theta^*), \quad \mathbf{p}^*(T) = -\nabla_{\mathbf{u}} \Phi(\mathbf{u}^*(T))^\top, & (\text{ADJ}) \\ H(\mathbf{u}^*(t); \mathbf{p}^*(t); \theta^*) = \max_{\theta} H(\mathbf{u}^*(t); \mathbf{p}^*(t); \theta), & (\text{MAX}) \end{cases}$$

are satisfied.

4. Algorithms of Successive Approximations

4.1. Basic Method of Successive Approximations

In light by the PMP, the basic Method of Successive Approximations (MSA) proposed in [8] are summarized in Algorithm 1 with an initial guess θ^0 of the control parameter [4,9].

Algorithm 1 The Basic MSA [8] to Solve Inverse Problem

- 1: **Initialize** θ^0 .
 - 2: **for** $k = 1, 2, \dots, K$ (training iterations) **do**
 - 3: Solve $\dot{\mathbf{u}}^k(t) = f(\mathbf{u}^k(t), \theta^{k-1}), \quad \mathbf{u}^k(0) = \mathbf{u}_0$;
 - 4: Solve $\dot{\mathbf{p}}^k(t) = -\nabla_{\mathbf{u}} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \theta^{k-1}), \quad \mathbf{p}^k(T) = -\nabla \Phi(\mathbf{u}^k(T))^\top$;
 - 5: Set $\theta^k = \arg \max_{\theta} \int_0^T H(\mathbf{u}^k(t), \mathbf{p}^k(t), \theta) dt$;
 - 6: **end for**
 - 7: **output:** θ^K .
-

In practice, rather than exactly solving the maximization problem in Line 5, it is often more efficient to perform a gradient ascent step. Specifically, the control parameter θ is updated according to:

$$\theta^k = \theta^{k-1} + \eta_k \nabla_{\theta} \int_0^T H(\mathbf{u}^k(t), \mathbf{p}^k(t), \theta^{k-1}) dt, \quad (9)$$

where $\eta_k > 0$ denotes the learning rate (step size) at iteration k [4].

As proved in [8], with a gradient ascent (9) to maximize the Hamiltonian, the Algorithm 1 is equivalent to gradient descent with back-propagation (BP). The same as BP, one bottleneck of the basic MSA is the linear memory cost $\mathcal{O}(T)$ to cache all the intermediate states $\{(\mathbf{u}^k(t), \mathbf{p}^k(t)) : t \in [0, T]\}$. Despite its conceptual elegance, one major bottleneck of the basic MSA (similar to standard BP) is the linear memory complexity $\mathcal{O}(T)$ due to the need to store all intermediate states $(\mathbf{u}^k(t), \mathbf{p}^k(t)) : t \in [0, T]$ for accurate gradient computation. This memory burden can become significant for long time horizons or fine temporal discretizations, leading to scalability issues [4].

For practical implementation, the forward-backward system is typically discretized. Let us denote a discretization with N time steps, $t_n = 0^N$, where $t_n = n\Delta t$ and $\Delta t = \frac{T}{N}$. Then, the forward propagation is:

$$\mathbf{u}^{k,n+1} = \mathbf{u}_n^k + \Delta t, f(\mathbf{u}_n^k, \theta^{k-1}), \quad \mathbf{u}_0^k = \mathbf{u}_0, \quad (10)$$

and the backward propagation becomes:

$$\mathbf{p}_n^k = \mathbf{p}_{n+1}^k + \Delta t, \nabla_{\mathbf{u}} H(\mathbf{u}_n^k, \mathbf{p}_n^k, \theta^{k-1}), \quad \mathbf{p}_N^k = -\nabla_{\mathbf{u}} \Phi(\mathbf{u}_N^k)^{\top}. \quad (11)$$

The discrete Hamiltonian maximization step is then:

$$\theta^k = \theta^{k-1} + \eta_k \sum_{n=0}^{N-1} \nabla_{\theta} H(\mathbf{u}_n^k, \mathbf{p}_n^k, \theta^{k-1}) \Delta t. \quad (12)$$

These discrete updates highlight the need for efficient memory management strategies [4], such as checkpointing or reversible dynamics, to alleviate the $\mathcal{O}(T)$ memory overhead.

4.2. MSA with Augmented Reverse-State

As an alternative perspective, (9) can also be viewed as solving the following ODE backward in time

$$\theta^k = \bar{\theta}^k(0), \text{ where } \frac{d\bar{\theta}^k(t)}{dt} = \eta_k \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(t)), \text{ with } \bar{\theta}^k(T) = \theta^{k-1}. \quad (13)$$

Suppose we integrate (13) backward w.r.t. time t , we get

$$\bar{\theta}^k(t) = \bar{\theta}^k(T) + \eta_k \int_T^t \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(T)) dt \quad (14a)$$

$$= \theta^{k-1} + \eta_k \int_T^t \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \theta^{k-1}) dt \quad (\text{Because we set } \bar{\theta}^k(T) = \theta^{k-1}) \quad (14b)$$

Then

$$\theta^k = \bar{\theta}^k(0) = \theta^{k-1} + \eta_k \int_T^0 \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \theta^{k-1}) dt \quad (15)$$

which is exactly the same as the gradient ascent of the Hamiltonian defined in (9).

From the definition of H , we know that $\nabla_{\theta} H(\mathbf{u}, \mathbf{p}, \theta) = \mathbf{p}^{\top} \nabla_{\theta} f(\mathbf{u}, \theta)$.

Instead of solving for co-state \mathbf{p} and control θ separately, here we can solve for the augmented reverse-state $[\mathbf{p}, \bar{\theta}]$ backward in time [4].

$$\theta^k = \bar{\theta}^k(0), \text{ where } \frac{d}{dt} \begin{bmatrix} \mathbf{p}^k(t) \\ \bar{\theta}^k(t) \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{u}} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(T)) \\ \eta_k \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(T)) \end{bmatrix}, \quad \begin{bmatrix} \mathbf{p}^k(T) \\ \bar{\theta}^k(T) \end{bmatrix} = \begin{bmatrix} -\nabla \Phi(\mathbf{u}^k(T))^{\top} \\ \theta^{k-1} \end{bmatrix}, \quad (16)$$

which can be easily verified that its numerical result is identical to Algorithm 1 with Line 5 solved by (9). But the merit is that solving (16) only demands constant memory $\mathcal{O}(1)$ to cache the augmented reverse-state $[\mathbf{p}, \bar{\theta}]$. However, in (16) we still need linear memory $\mathcal{O}(T)$ to store all intermediate forward state $\{\mathbf{u}^k(t) : t \in [0, T]\}$. One way to tackle this issue is to augment it with the reverse state and solve for $\mathbf{u}^k(t)$ backward as well with backward initial $\mathbf{u}^k(T)$ computed from the forward pass [10], as shown in (17) [4].

$$\theta^k = \bar{\theta}^k(0), \text{ where } \frac{d}{dt} \begin{bmatrix} \mathbf{u}^k(t) \\ \mathbf{p}^k(t) \\ \bar{\theta}^k(t) \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{p}} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(t)) \\ -\nabla_{\mathbf{u}} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(t)) \\ \eta_k \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(t)) \end{bmatrix}, \begin{bmatrix} \mathbf{u}^k(T) \\ \mathbf{p}^k(T) \\ \bar{\theta}^k(T) \end{bmatrix} = \begin{bmatrix} \mathbf{u}^k(T) \\ -\nabla \Phi(\mathbf{u}^k(T))^{\top} \\ \theta^{k-1} \end{bmatrix}. \quad (17)$$

The whole process of (17) has constant memory cost $\mathcal{O}(1)$ which frees the space of $\{\mathbf{u}^k(t) : t \in [0, T]\}$. But as a sacrifice we do trade the time for space as solving for (17) requires the re-computation of $\mathbf{u}^k(t)$.

If we replace the Line 4-5 in Algorithm 1 by (16) or (17), we can get an more memory-efficient algorithm with identical numerical result [4].

4.3. MSA with Backward Control Flow

As a careful observation in (13)-(17) on the right-hand side of the ODEs, all partial derivatives of H are evaluated on the initial state $\bar{\theta}^k(T)$. In this work, we further free this constraint and give more freedom to the dynamical system [4]. So instead of computing the partial derivative of H on the initial $\bar{\theta}^k(T)$, we compute it on the intermediate state $\bar{\theta}^k(t)$. As discussed in [11–13], the basic idea of the successive approximation methods is to find the optimal parameters from the guess by successive projections onto the manifold defined by the ODEs. So intuitively a better guess will contribute to better convergence performance and even better result. Our modification here gives a better guess to the optimal parameter θ^* for the Hamiltonian H at the intermediate time t , since $\bar{\theta}^k(t)$ is optimized backward in time so $\bar{\theta}^k(t)$ is usually a better estimation point than $\bar{\theta}^k(T)$. Along with (16), we summarize our proposed algorithm in Algorithm 2. The forward pass is to compute the trajectory of $\mathbf{u}^k(t)$, the backward pass is to compute the gradient flow for the co-state $\mathbf{p}^k(t)$ and control $\bar{\theta}^k(t)$. We would like to point out that here we did not extend (17) to the algorithm, because with changing to $\bar{\theta}^k(t)$, the trajectory of $\mathbf{u}^k(t)$ might flow to somewhere else, we cannot guarantee $\mathbf{u}^k(0) = \mathbf{u}_0$ anymore. And this might cause the algorithm unstable. We will leave this problem to the future work [4].

Algorithm 2 The MSA with Reverse Augmented State and Control Flow to Solve Inverse Problem

- 1: **Initialize** θ^0 .
 - 2: **for** $k = 1, 2, \dots, K$ (training iterations) **do**
 - 3: Solve $\dot{\mathbf{u}}^k(t) = f(\mathbf{u}^k(t), \theta^{k-1})$, $\mathbf{u}^k(0) = \mathbf{u}_0$;
 - 4: Solve $\frac{d}{dt} \begin{bmatrix} \mathbf{p}^k(t) \\ \bar{\theta}^k(t) \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{u}} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(t)) \\ \eta_k \nabla_{\theta} H(\mathbf{u}^k(t), \mathbf{p}^k(t), \bar{\theta}^k(t)) \end{bmatrix}$, $\begin{bmatrix} \mathbf{p}^k(T) \\ \bar{\theta}^k(T) \end{bmatrix} = \begin{bmatrix} -\nabla \Phi(\mathbf{u}^k(T))^{\top} \\ \theta^{k-1} \end{bmatrix}$;
 - 5: Set $\theta^k = \bar{\theta}^k(0)$.
 - 6: **end for**
 - 7: **output:** θ^K .
-

4.4. Time Discretization

We discretize $[0, T]$ to be $0, 1, 2, \dots, T$ and employ the explicit Euler method for the forward ODE and Verlet method [14] for the backward augmented ODE, then we have the discretized version of Algorithm 2, which is summarized in Algorithm 3. This algorithm was inspired by the learnable optimization algorithms for solving inverse problem [15,16], in the context of MRI reconstruction.

Algorithm 3 The Discretized Version of Algorithm 2 to Solve Inverse Problem [4]

```

1: Initialize  $\theta^0$ .
2: for  $k = 1, 2, \dots, K$  (training iterations) do
3:   Set  $\mathbf{u}_0^k = \mathbf{u}_0$ ;
4:   for  $t = 0, 1, 2, \dots, T-1$  do
5:      $\mathbf{u}_{t+1}^k = \mathbf{u}_t^k + \tau_k f(\mathbf{u}_t^k, \theta^{k-1})$ ;
6:   end for
7:   Set  $\mathbf{p}_T^k = -\nabla \Phi(\mathbf{u}_T^k)^\top$  and  $\bar{\theta}_T^k = \theta^{k-1}$ ;
8:   for  $t = T-1, T-2, \dots, 0$  do
9:      $\bar{\theta}_t^k = \bar{\theta}_{t+1}^k + \eta_k \nabla_{\theta} H(\mathbf{u}_t^k, \mathbf{p}_{t+1}^k, \bar{\theta}_{t+1}^k)$ ;
10:     $\mathbf{p}_t^k = \mathbf{p}_{t+1}^k - \nabla_{\mathbf{u}} H(\mathbf{u}_t^k, \mathbf{p}_{t+1}^k, \bar{\theta}_t^k)$ ;
11:  end for
12:  Set  $\theta^{k+1} = \bar{\theta}_0^{k+1}$ ;
13: end for
14: output:  $\theta^K$ .
```

5. Design of the Regularizer in the Variational Model

In this study, we design the regularizer \mathcal{R} to take the form $\mathcal{R}(\mathbf{u}) = \psi(\mathcal{G}(\mathbf{u})) \in \mathbb{R}^+$, where $\mathcal{G} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a feature extraction operator achieved by deep neural network [4]. To enforce the sparsity, the function ψ is defined as $\psi(\mathbf{u}) = \sum_{i=1}^n \log \cosh(\mathbf{u}_i)$, where $\log \cosh(x)$ is a twice differentiable function approximately equal to $|x| - \log(2)$ for large x and to $x^2/2$ for small x . Throughout this work, we parameterize the feature extraction operator \mathcal{G} as a vanilla l -layer convolutional neural network without bias separated by componentwise activation function as follows [4]:

$$\mathcal{G}(\mathbf{u}) = \mathbf{w}_l * \sigma \dots \sigma(\mathbf{w}_3 * \sigma(\mathbf{w}_2 * \sigma(\mathbf{w}_1 * \mathbf{u}))), \quad (18)$$

where $\{\mathbf{w}_k\}_{k=1}^l$ denote the convolution weights and $*$ denotes the convolution operation. Specifically, we parameterize the first convolution \mathbf{w}_0 to be d kernels of size 3×3 and the last one \mathbf{w}_l with 1 kernel of size $3 \times 3 \times d$. Besides, all hidden layers $\{\mathbf{w}_k\}_{k=2}^{l-1}$ correspond to convolutions with d kernels of size $3 \times 3 \times d$. Here, σ represents a componentwise activation function which is twice differentiable. In this work we adopt the twice differentiable function sigmoid-weighted linear unit (SiLU) [17] as the activation.

6. Discussion

Recent advances in machine learning have significantly contributed to solving complex inverse problems and improving quantitative imaging techniques. In the context of MRI reconstruction, Bian et al. [18] introduced a self-supervised learning framework with model reinforcement, demonstrating enhanced accuracy and robustness for rapid T_1 mapping. Building upon these foundations, Bian et al. [19] proposed a diffusion modeling approach with domain-conditioned prior guidance, further accelerating both MRI and qMRI reconstructions while improving fidelity to the underlying physical models. From a theoretical perspective, Bian [4] also developed an optimal control-based bilevel optimization framework for inverse problems, providing a principled way to design deep learnable regularizers and control training dynamics.

Beyond medical imaging, in robotics and aerospace engineering, Gao et al. [20] proposed an autonomous multi-robot system for spacecraft servicing, showcasing the role of intelligent control and coordination in complex operational settings. Similarly, Yang et al. [21] explored the application of machine learning to assess digitalization capabilities in business finance, highlighting the versatility of data-driven methodologies across domains. Extending this direction, Gao et al. [22] addressed the adaptive detumbling of non-rigid satellites, integrating learning-based strategies to handle uncertainties in satellite dynamics. These advances collectively underscore the importance of combining domain

knowledge, optimization theory, and machine learning techniques to address diverse challenges across imaging, finance, and autonomous systems.

7. Conclusions

This work is inspired by the work in [4] and recent advancements in multi-task MRI reconstruction using meta-learning approaches [23], we aim to design a scalable and principled learning framework that not only solves individual inverse problems efficiently but also adapts to varying data distributions and task-specific challenges.

In this work, we proposed a principled framework for solving inverse problems by casting deep learning-based reconstruction as an optimal control problem. Building on the variational model formulation, we introduced a bilevel optimization structure that integrates gradient flow dynamics with the Method of Successive Approximations (MSA) to train the network parameters in a theoretically grounded manner. Our analysis established connections between MSA and classical back-propagation, while highlighting the scalability challenges associated with memory growth. To address these limitations, we developed memory-efficient variants of MSA, including augmented reverse-state formulations and modified backward control flows, enabling constant memory cost while preserving convergence properties.

Through discrete-time implementations and the design of a learnable regularizer via deep convolutional networks, we demonstrated how this framework provides a flexible, scalable, and interpretable approach to tackle challenging inverse problems such as sparse-view CT and accelerated MRI reconstruction. Future directions include further improvements in algorithmic stability, integration with multi-physics models, and applications to broader classes of inverse problems beyond medical imaging.

References

1. Iqbal, M.; Naeem, M.; Anpalagan, A.; Ahmed, A.; Azam, M. Wireless sensor network optimization: Multi-objective paradigm. *Sensors* **2015**, *15*, 17572–17620.
2. Fei, Z.; Li, B.; Yang, S.; Xing, C.; Chen, H.; Hanzo, L. A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems. *IEEE Communications Surveys & Tutorials* **2016**, *19*, 550–586.
3. Bian, W.; Tamilselvam, Y.K. A Review of Optimization-Based Deep Learning Models for MRI Reconstruction. *AppliedMath* **2024**, *4*, 1098–1127.
4. Bian, W. An Optimal Control Approach for Inverse Problems with Deep Learnable Regularizers. *arXiv preprint arXiv:2409.00498* **2024**.
5. Teschl, G. *Ordinary differential equations and dynamical systems*; Vol. 140, American Mathematical Soc., 2012.
6. Bian, W.; Chen, Y.; Ye, X. Deep parallel MRI reconstruction network without coil sensitivities. In Proceedings of the Machine Learning for Medical Image Reconstruction: Third International Workshop, MLMIR 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 3. Springer, 2020, pp. 17–26.
7. Ambrosio, L.; Gigli, N.; Savare, G. Gradient Flows in Metric Spaces and in the Space of Probability Measures. *Lectures in Mathematics ETH Zurich* **2005**. <https://doi.org/10.1007/978-3-7643-8722-8>.
8. Li, Q.; Chen, L.; Tai, C.; E, W. Maximum Principle Based Algorithms for Deep Learning. *Journal of Machine Learning Research* **2018**, *18*, 1–29.
9. Bian, W.; Chen, Y.; Ye, X.; Zhang, Q. An optimization-based meta-learning model for mri reconstruction with diverse dataset. *Journal of Imaging* **2021**, *7*, 231.
10. Chen, R.T.Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D.K. Neural Ordinary Differential Equations. In Proceedings of the Advances in Neural Information Processing Systems; Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; Garnett, R., Eds. Curran Associates, Inc., 2018, Vol. 31.
11. Li, Q.; Hao, S. An Optimal Control Approach to Deep Learning and Applications to Discrete-Weight Neural Networks. In Proceedings of the Proceedings of the 35th International Conference on Machine Learning; Dy, J.; Krause, A., Eds. PMLR, 10–15 Jul 2018, Vol. 80, *Proceedings of Machine Learning Research*, pp. 2985–2994.

12. Bian, W.; Zhang, Q.; Ye, X.; Chen, Y. A learnable variational model for joint multimodal MRI reconstruction and synthesis. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2022, pp. 354–364.
13. Smale, S. Differentiable dynamical systems. *Bulletin of the American mathematical Society* **1967**, *73*, 747–817.
14. Ascher, U.; Petzold, L. Computer methods for ordinary differential equations and differential-algebraic equations. SIAM, 1998.
15. Bian, W.; Chen, Y.; Ye, X. An optimal control framework for joint-channel parallel MRI reconstruction without coil sensitivities. *Magnetic Resonance Imaging* **2022**.
16. Bian, W. Optimization-Based Deep learning methods for Magnetic Resonance Imaging Reconstruction and Synthesis. PhD thesis, University of Florida, 2022.
17. Elfving, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks* **2018**, *107*, 3–11.
18. Bian, W.; Jang, A.; Liu, F. Improving quantitative MRI using self-supervised deep learning with model reinforcement: Demonstration for rapid T1 mapping. *Magnetic Resonance in Medicine* **2024**, *92*, 98–111.
19. Bian, W.; Jang, A.; Zhang, L.; Yang, X.; Stewart, Z.; Liu, F. Diffusion modeling with domain-conditioned prior guidance for accelerated mri and qmri reconstruction. *IEEE Transactions on Medical Imaging* **2024**.
20. Gao, L.; Cordova, G.; Danielson, C.; Fierro, R. Autonomous multi-robot servicing for spacecraft operation extension. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2023, pp. 10729–10735.
21. Yang, J.; Liu, J.; Yao, Z.; Ma, C. Measuring digitalization capabilities using machine learning. *Research in International Business and Finance* **2024**, *70*, 102380. <https://doi.org/https://doi.org/10.1016/j.ribaf.2024.102380>.
22. Gao, L.; Danielson, C.; Fierro, R. Adaptive Robot Detumbling of a Non-Rigid Satellite. *arXiv preprint arXiv:2407.17617* **2024**.
23. Bian, W.; Jang, A.; Liu, F. Multi-task magnetic resonance imaging reconstruction using meta-learning. *Magnetic Resonance Imaging* **2025**, *116*, 110278.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.