Article

Not peer-reviewed version

Solve Bi-Level Optimization Model for Meta-Learning Using Method of Lagrangian Multipliers

Alicia Arora * and Palack Batra

Posted Date: 28 April 2025

doi: 10.20944/preprints202504.2285.v1

Keywords: Meta-learning; Bilevel optimization; Method of Lagrangian Multipliers

Article

Solve Bi-Level Optimization Model for Meta-Learning Using Method of Lagrangian Multipliers

Alicia Arora * and Palack Batra

Independent Researcher
* Correspondence: vivienbian624@gmail.com

Abstract: Optimization-based meta-learning has emerged as a powerful framework for improving model generalization, especially in domains with diverse and heterogeneous data distributions. In this work, we propose a bilevel optimization model for meta-learning, explicitly framed through an optimal control perspective. Our approach formulates the meta-training process as a constrained optimization problem, where the lower-level updates task-specific models using a learnable unrolling network, and the upper-level adjusts hyperparameters to minimize validation losses across tasks. By applying the Method of Lagrangian Multipliers (MLM), we model both the primal reconstruction variables and the dual multipliers, ensuring that updates respect the dynamic constraints of the optimization process. We prove the theoretical equivalence between direct loss minimization and Lagrangian-based optimization and develop an efficient algorithm for network training. Experimental motivations drawn from magnetic resonance imaging (MRI) reconstruction suggest that our framework offers scalable and principled solutions, with potential for broader impact in general inverse problems and meta-learning scenarios.

Keywords: meta-learning; bilevel optimization; method of lagrangian multipliers

1. Introduction

Meta-learning, often referred to as "learning to learn," [1–3] has emerged as a powerful framework for improving model generalization, particularly in domains where data distributions vary significantly across tasks.

Inspired by these advances[4], our proposed model integrates task correlation learning directly into the optimization procedure, seeking not only to minimize training loss but also to enhance the quality of downstream decision-making through better generalization. In magnetic resonance imaging (MRI) reconstruction, where datasets can differ widely in anatomy, acquisition protocols, and imaging artifacts, meta-learning provides a promising avenue to build adaptable models. Building on the foundation of optimization-based deep learning methods for MRI reconstruction[5,6] , we propose a bi-level optimization model for meta-learning that explicitly captures the relationship between training and validation datasets through learnable task correlations.

This work is motivated by a line of recent studies exploring optimization-driven and meta-learning approaches, including an optimization-based meta-learning model for diverse datasets[7–9] and a learnable variational model for joint multimodal MRI reconstruction and synthesis[10]. These advances illustrate that incorporating optimization formulations into the learning process enhances model robustness across diverse conditions.

Our framework formulates the meta-training as a constrained bi-level optimization problem, where the lower-level problem learns task-specific models weighted by a normalized hyperparameter matrix, and the upper-level problem adjusts these hyperparameters to minimize validation losses. Furthermore, we extend the training dynamics using the method of Lagrangian multipliers (MLM) to provide theoretical equivalence between direct loss minimization and primal-dual optimization updates, following similar optimization principles as used in optimal control frameworks for image processing[11,12].

By synthesizing principles from optimization theory, control dynamics[13], and deep learning, our approach offers a principled and scalable solution for meta-learning training algorithm, potentially improving performance on heterogeneous, real-world imaging datasets. This work continues the trajectory of optimization-based learning[9,12,14].

1.1. Optimization-Based Meta-Learning: Frameworks and Algorithms

In optimization-based meta-learning, the learner is not just learning parameters — it's learning how to optimize.

Instead of using a fixed optimization method (like SGD or Adam), you train an optimizer (or regularizer) so that across many tasks, it adapts quickly and solves new tasks better. This is typically framed as a bi-level optimization problem:

$$\min_{\theta} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}[\mathcal{L}_{\text{meta}}(\theta;\mathcal{T})]$$

where:

- θ denotes the parameters of the optimizer, model, or regularizer,
- $\mathcal{T} \sim p(\mathcal{T})$ represents tasks sampled from a distribution $p(\mathcal{T})$,
- $\mathcal{L}_{\text{meta}}(\theta;\mathcal{T})$ measures how well the task is solved after a few optimization steps.

At each meta-training iteration:

1. Sample a batch of tasks $\mathcal{T}_i$ from the task distribution.
2. For each task $\mathcal{T}_i$:
   - Initialize model parameters x (e.g., randomly or from a pre-trained model).
   - **Inner Loop:** Solve the task-specific optimization problem for a few steps using the current optimizer (parameterized by θ), yielding adapted parameters x_i'.
   - Compute the task loss $\mathcal{L}_{\text{task}}(x_i';\mathcal{T}_i)$.
3. **Outer Loop:**
   - Aggregate all the task losses to compute a meta-loss.
   - Update θ (the optimizer or model parameters) via gradient descent to improve performance on future tasks.

2. Problem Settings

Suppose we are given a set of N training examples, denoted as $\{(\mathbf{x}^{(j)}, \mathbf{u}^{*(j)})\}_{j=1}^{N}$, where for each index $j \in \{1,\ldots,N\}$:

- $\mathbf{x}^{(j)}$ represents the input data (e.g., an undersampled measurement or an input image),
- $\mathbf{u}^{*(j)}$ denotes the corresponding ground-truth output (e.g., a fully sampled or high-quality image).

We aim to train a neural network parameterized by Θ, where the objective is to learn Θ by minimizing a loss function ℓ that measures the discrepancy between the network's output and the ground truth across the training set.

This training procedure can be formulated as a **bilevel optimization problem**[15], consisting of two levels:

- **Lower-level optimization:** For a fixed set of trainable parameters Θ, we update the reconstruction variable $\mathbf{u}$ by solving a task-specific optimization problem guided by the network.
- **Upper-level optimization:** After updating $\mathbf{u}$, we update the network parameters Θ by minimizing the empirical loss ℓ over the training dataset.

Thus, the network training involves alternating between solving for $\mathbf{u}$ given Θ, and optimizing Θ based on the performance measured by the loss function ℓ.

$$\begin{aligned}
\min_{\Theta} \quad & \ell(\mathbf{u}(T)), && \text{(1a)}\\
\text{s.t.} \quad & \mathbf{u}(t) = \mathbf{g}(\mathbf{u}(t-1), \theta_t), \quad t = 1 \cdots T, && \text{(1b)}\\
& \mathbf{u}(0) = \mathbf{u}_0, && \text{(1c)}
\end{aligned}$$

Set $\mathbf{U} = (\mathbf{u}(0), \cdots, \mathbf{u}(T))^\intercal$ and $\Theta = (\theta_1, \cdots, \theta_T)^\intercal$ be the collection of states $\mathbf{u}(t)$ and controls θ_t at all time steps respectively. $\mathbf{g}$ is a multi-phase unrolling network inspired by the proximal gradient algorithm, and the output of $\mathbf{g}(\cdot) \in \mathbb{C}^{m \times n \times c}$ is the updated multi-coil MRI data from each phase. In our framework, we introduce a neural network $\mathbf{g}$ that serves as an intermediate mapping between consecutive states of the reconstruction variable. Specifically, at each iteration t (where $t = 0, \ldots, T-1$), the network $\mathbf{g}$ takes the current estimate $\mathbf{u}^{(t)}$ and produces an updated estimate $\mathbf{u}^{(t+1)}$:

$$\mathbf{u}^{(t+1)} = \mathbf{g}(\mathbf{u}^{(t)}; \theta(t)),$$

where $\theta(t)$ denotes the set of trainable parameters at iteration t.

To initialize the reconstruction process, we define a separate network $\mathbf{g}_0$, which operates with an initial set of control parameters $\theta(0)$. The role of $\mathbf{g}_0$ is to map the given partial k-space measurements $\mathbf{f}$ to an initial image reconstruction $\mathbf{u}^{(0)}$:

$$\mathbf{u}^{(0)} = \mathbf{g}_0(\mathbf{f}; \theta(0)).$$

This initial estimate $\mathbf{u}^{(0)}$ then serves as the starting point for the iterative reconstruction process governed by the optimal control system, where $\mathbf{g}$ is applied sequentially to refine the reconstruction across T iterations.

Suppose the training data $\mathcal{D}^{tr} = \{\mathcal{D}^{tr}_{\tau_i}\}_{i=1}^{\mathcal{N}}$ which consists of $\mathcal{N}$ batches.

And the validation data $\mathcal{D}^{val} = \{\mathcal{D}^{val}_{v_j}\}_{j=1}^{\mathcal{M}}$ which consists of $\mathcal{M}$ batches.

Let θ be the collection of the network parameters and α be the set of hyper-parameters. We optimize the following bi-level model:

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{j=1}^{\mathcal{M}} \ell_{v_j}(\theta(\alpha); \mathcal{D}^{val}_{v_j}) \tag{2a}$$

$$s.t. \;\; \theta(\alpha) = \arg\min_{\bar{\theta}} \sum_{i=1}^{\mathcal{N}} \sigma(\alpha)_i \cdot \ell_{\tau_i}(\bar{\theta}; \mathcal{D}^{tr}_{\tau_i}), \tag{2b}$$

$$\text{where} \;\; \sum_i \sigma(\alpha)_i = 1, \;\; 0 \le \sigma(\alpha)_i \le 1. \tag{2c}$$

where ℓ denotes the loss function which can be taken as cross entropy loss. And the function σ is to normalize the hyperparameter vector α, here we can simply take σ as the softmax function.

Also we propose a generalized model where $\alpha \in \mathcal{R}^{\mathcal{N} \times \mathcal{M}}$, and we denote $\alpha_j \in \mathcal{R}^{\mathcal{N}}$ denotes the j-th column. After we apply the normalization function σ (in (3c)), the matrix $\sigma(\alpha)$ can be regarded as a weight matrix. Intuitively and ideally, each component $\sigma(\alpha_j)_i$ can represent the correlation between $\mathcal{D}^{tr}_{\tau_i}$ and $\mathcal{D}^{val}_{v_j}$.

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{j=1}^{\mathcal{M}} \ell_{v_j}(\theta(\alpha_j); \mathcal{D}^{val}_{v_j}) \tag{3a}$$

$$s.t.\ \ \theta(\alpha_j) = \arg\min_{\overline{\theta}} \sum_{i=1}^{\mathcal{N}} \sigma(\alpha_j)_i \cdot \ell_{\tau_i}(\overline{\theta}; \mathcal{D}^{tr}_{\tau_i}), \tag{3b}$$

$$\text{where}\ \ \sum_i \sigma(\alpha_j)_i = 1,\ \ 0 \leq \sigma(\alpha_j)_i \leq 1. \tag{3c}$$

Algorithm 1:

```
Input: Initialize hyper-parameters $\alpha^0$ and network parameters $\theta^0$.
for epoch $p = 1, 2, \ldots, P$ do
    $\boldsymbol{\theta^p = \arg\min_{\overline{\theta}} \sum_{i=1}^{\mathcal{N}} \sigma(\alpha^{p-1})_i \cdot \ell_{\tau_i}(\overline{\theta}; \mathcal{D}^{tr}_{\tau_i})}$
        ▷ Perform (k steps) gradient update steps in SGD on $\boldsymbol{\mathcal{D}^{tr}_{\tau_i}}$.
    $\boldsymbol{\theta^p_i = \arg\min_{\theta} \alpha_i^{p-1} \cdot \ell_{\tau_i}(\theta, \mathcal{D}^{tr}_{\tau_i})}$
    $\boldsymbol{\alpha^p = \arg\min_{\alpha} \sum_{j=1}^{\mathcal{M}} \ell_{v_j}(\theta^p; \mathcal{D}^{val}_{v_j})}$    ▷ Perform gradient update steps in SGD on $\boldsymbol{\mathcal{D}^{val}_{v_i}}$.
end for
Output: $\alpha^P, \theta^P$.
```

3. Network Training from the View of Method of Lagrangian Multipliers (MLM)

The network parameters to be solved from (3) are $\Theta = \{\theta_t : t = 1, \cdots T\}$. First, we apply MLM to solve the control problem in section 2. The Lagrangian of the control problem (3) is

$$\begin{aligned} L(\mathbf{U}, \Theta; \Lambda) = \ell(\mathbf{u}(T)) + \textstyle\sum_{t=1}^{T} \langle \lambda_t, \mathbf{u}(t) - \mathbf{g}(\mathbf{u}(t-1), \theta_t) \rangle \\ + \langle \lambda_0, \mathbf{u}(0) - \mathbf{u}_0 \rangle, \end{aligned} \tag{4}$$

where $\Lambda = (\lambda_0, \cdots, \lambda_T)^{\intercal}$ are Lagrangian multipliers of (3).

We want to find primal solution $(\mathbf{U}^*, \Theta^*)$ and dual solution Λ^* to solve (3), if $(\mathbf{U}^*, \Theta^*; \Lambda^*)$ minimize (4), by the first order optimality condition, we have

$$\begin{cases} \partial_{\mathbf{U}} L(\mathbf{U}^*, \Theta^*; \Lambda^*) = 0 & (5a) \\ \partial_{\Theta} L(\mathbf{U}^*, \Theta^*; \Lambda^*) = 0 & (5b) \\ \partial_{\Lambda} L(\mathbf{U}^*, \Theta^*; \Lambda^*) = 0 & (5c) \end{cases}$$

The algorithm is conducted in the following manner:

1. Fix Θ and define $(\overline{\mathbf{U}}, \overline{\Lambda}) := \min_{\mathbf{U}, \Lambda} L(\mathbf{U}, \Theta; \Lambda)$, then by the first order optimality condition of $\lambda_t, t = 0, \cdots, T$ we get

$$\partial_{\lambda_0} \langle \lambda_0, \mathbf{u}(0) - \mathbf{u}_0 \rangle = 0 \implies \bar{\mathbf{u}}(0) = \mathbf{u}_0 \tag{6a}$$

$$\partial_{\lambda_t} \langle \lambda_t, \mathbf{u}(t) - \mathbf{g}(\mathbf{u}(t-1), \theta_t) \rangle = 0 \implies \bar{\mathbf{u}}(t) = \mathbf{g}(\bar{\mathbf{u}}(t-1), \theta_t),\ t = 1, \cdots, T, \tag{6b}$$

By the first order optimality condition of $\mathbf{u}(t), t = 1, \cdots, T$ we get

$$\partial_{\mathbf{u}(T)} [\ell(\mathbf{u}(T)) + \langle \lambda_T, \mathbf{u}(T) \rangle] = 0 \implies \bar{\lambda}_T = -\partial_{\mathbf{u}(T)} \ell(\bar{\mathbf{u}}(T)), \tag{7a}$$

$$\partial_{\mathbf{u}(t)} [\langle \lambda_t, \mathbf{u}(t) \rangle + \langle \lambda_{t+1}, -\mathbf{g}(\mathbf{u}(t), \theta_{t+1}) \rangle] = 0 \implies \bar{\lambda}_t = \langle \bar{\lambda}_{t+1}, \partial_{\mathbf{u}(t)} \mathbf{g}(\bar{\mathbf{u}}(t), \theta_{t+1}) \rangle,\ t = 1, \cdots, T-1, \tag{7b}$$

$$\partial_{\mathbf{u}(0)} [\langle \lambda_0, \mathbf{u}(0) \rangle + \langle \lambda_1, -g(\mathbf{u}(0), \theta_1) \rangle] = 0 \implies \bar{\lambda}_0 = \langle \bar{\lambda}_1, \partial_{\mathbf{u}(0)} \mathbf{g}(\bar{\mathbf{u}}(0), \theta_1) \rangle. \tag{7c}$$

Therefore for fixed Θ we get the optimal solutions $(\overline{\mathbf{U}}, \overline{\Lambda})$ satisfy (6) and (7).

2. For fixed $(\mathbf{U}, \Lambda)$, we compute the gradient $\partial_\Theta L(\mathbf{U}, \Theta; \Lambda)$:

$$\partial_\Theta L(\mathbf{U}^k, \Theta^k; \Lambda^k) = -\langle \lambda_0^k, \partial_{\theta_0}(\mathbf{u}_k^{(0)} - \mathbf{u}_0^k)\rangle = 0, \text{ for } t = 0. \tag{8a}$$

$$\partial_{\theta_t} L(\mathbf{U}, \theta_t; \Lambda) = \partial_{\theta_t}[-\langle \lambda_t, \mathbf{g}(\mathbf{u}(t-1), \theta_t)\rangle] \tag{8b}$$

$$= -\langle \lambda_t, \partial_{\theta_t}\mathbf{g}(\mathbf{u}(t-1), \theta_t)\rangle \text{ for } t = 1, \cdots, T. \tag{8c}$$

Recall the conventional SGD method, we update $\Theta^{k+1} = \Theta^k - \frac{\eta}{n}\sum_{i=1}^{n} \partial_\Theta \ell(\mathbf{u}_i^k(T)(\Theta^k))$, where $\mathbf{u}^k(T)$ is interpreted as a function of $\Theta^k = (\theta_1^k, \cdots, \theta_T^k)^\intercal$.

```
Algorithm 2: Network Training by SGD
Hyperparameter: $K$ (Numbers of Iterations), $\eta$ (learning rate);
initialize: Initial guess $\Theta^0$ ;
for $k = 0$ to $K - 1$ do
    $\Theta^{k+1} = \Theta^k - \frac{\eta}{n}\sum_{i=1}^{n} \partial_\Theta \ell(\mathbf{u}_i^k(T)(\Theta^k))$.
end
Output: $\Theta^K$.
```

In the following theorem, we show the equivalence between $\partial_\Theta L(\overline{\mathbf{U}}, \Theta; \overline{\Lambda})$ and $\partial_\Theta \ell(\bar{\mathbf{u}}(T)(\Theta))$.

Theorem 1. $\partial_\Theta L(\overline{\boldsymbol{U}}, \Theta; \overline{\Lambda}) = \partial_\Theta \ell(\bar{\mathbf{u}}(T)(\Theta))$.

Proof. From (6) we have

$$\partial_{\theta_t}\mathbf{g}(\bar{\mathbf{u}}(t-1), \theta_t) = \partial_{\theta_t}\bar{\mathbf{u}}(t), \; t = 1, \cdots, T \tag{9a}$$

From (7b), we have

$$\bar{\lambda}_{T-1} = \langle \bar{\lambda}_T, \partial_{\mathbf{u}(T-1)}\mathbf{g}(\bar{\mathbf{u}}(T-1), \theta_T)\rangle \tag{10a}$$

$$= \langle -\partial_{\mathbf{u}(T)}\ell(\bar{\mathbf{u}}(T)), \partial_{\mathbf{u}(T-1)}\bar{\mathbf{u}}(T)\rangle \text{ by (7a), (6b)} \tag{10b}$$

$$= -\partial_{\mathbf{u}(T-1)}\ell(\bar{\mathbf{u}}(T)) \tag{10c}$$

Similarly, we can show (10) holds for $t = 0, \cdots, T-1$

$$\bar{\lambda}_t = -\partial_{\mathbf{u}(t)}\ell(\bar{\mathbf{u}}(T)), \tag{11}$$

together with (7a), we have

$$\bar{\lambda}_t = -\partial_{\mathbf{u}(t)}\ell(\bar{\mathbf{u}}(T)), \; t = 0, \cdots, T. \tag{12}$$

Hence, (8c) reduces to

$$\partial_{\theta_t} L(\overline{\mathbf{U}}, \theta_t; \overline{\Lambda}) = -\langle \bar{\lambda}_t, \partial_{\theta_t}\, \mathbf{g}(\bar{\mathbf{u}}(t-1), \theta_t)\rangle \tag{13a}$$

$$= \langle \partial_{\mathbf{u}(t)}\ell(\bar{\mathbf{u}}(T)), \partial_{\theta_t}\bar{\mathbf{u}}(t)\rangle \text{ by (12), (9a)} \tag{13b}$$

$$= \partial_{\theta_t}\ell(\bar{\mathbf{u}}(T)(\theta_t)), \; t = 1, \cdots, T. \tag{13c}$$

Thus, we derive $\partial_\Theta L(\overline{\mathbf{U}}, \Theta; \overline{\Lambda}) = \partial_\Theta \ell(\bar{\mathbf{u}}(T)(\Theta))$. □

This theorem shows that applying SGD on the loss function ℓ is equivalent to performing SGD on L. We want to reach the optimal solutions $(\overline{\mathbf{U}}, \overline{\Lambda})$ with fixed Θ, so we proceed the algorithm to update

Θ in the iterative scheme by replacing notation $\overline{\mathbf{U}}, \overline{\Lambda}$ as $\mathbf{U}^k, \Lambda^k$. This training process is summarized in Algorithm 3, where $\Lambda^k = (\lambda_0^k, \lambda_1^k, \cdots, \lambda_T^k)^\intercal$, $\Theta^k = (\theta_1^k, \cdots, \theta_T^k)^\intercal$, and $\mathbf{U}^k = (\mathbf{u}^k(0), \mathbf{u}^k(1), \cdots, \mathbf{u}^k(T))^\intercal$ for each iteration $k = 0, 1, \cdots, K$. Now we can use one step of SGD optimization methods such as ADAM to update Θ by replacing $\partial_\Theta \ell(\mathbf{u}^k(T)(\Theta))$ to be $\partial_\Theta L(\mathbf{U}^k, \Theta; \Lambda^k)$ with gradient computed in (8).

Algorithm 3: Network Training by MLM

```
**Hyperparameter:** $K$ (Number of Iterations);
**Initialize:** Initial guess $\theta_t^0 \in \Theta^0, t = 1, \cdots T-1$;
for $k = 0$ to $K-1$ do
    Set $\mathbf{u}^k(0) = \mathbf{u}_0$;
    for $t = 1$ to $T$ do
        $\mathbf{u}^k(t) = \mathbf{g}(\mathbf{u}^k(t-1), \theta_t^k)$ ;
    end
    Set $\lambda_T^k = -\partial_{\mathbf{u}(T)} \ell(\mathbf{u}^k(T))$ ;
    for $t = T-1$ to $0$ do
        $\lambda_t^k = \langle \lambda_{t+1}^k, \partial_{\mathbf{u}(t)} \mathbf{g}(\mathbf{u}^k(t), \theta_{t+1}^k) \rangle$;
    end
    for $t = 1$ to $T$ do
        Update $\theta_t^{k+1} = \arg\min_{\theta_t} L(\mathbf{U}^k, \theta_t; \Lambda^k)$ by (8) and SGD algorithms;
    end
end
**Output:** $\theta_t^K, t = 1, \cdots, T$.
```

Algorithm 3 trains a meta-learning model by solving a bilevel optimization problem using the Method of Lagrangian Multipliers (MLM).

Instead of naively updating the network parameters by only minimizing the loss, this algorithm:

- Explicitly tracks the **primal variables** (the reconstruction variables $\mathbf{u}(t)$),
- And the **dual variables** (the Lagrange multipliers λ_t),

to ensure that the updates respect the dynamic constraint at each iteration:

$$\mathbf{u}(t+1) = \mathbf{g}(\mathbf{u}(t), \theta_t),$$

where $\mathbf{g}$ denotes the learned update network and θ_t are the trainable control parameters at iteration t.

The forward pass simulates the network behavior over T iterations (like "unrolling" the optimization).

The backward pass computes how the loss at the end depends on every single step (using the chain rule and multipliers). The parameter update ensures that the learned optimizer $\mathbf{g}$ not only reduces loss but respects the dynamic constraints (i.e., how $\mathbf{u}(t)$ evolves over time).

Thus, it mimics optimal control theory where we treat the evolution of the system as a dynamic process, not just static minimization.

4. Discussion

Optimization is fundamental to solving inverse problems, yet traditional methods often require careful tuning and lack flexibility across domains. Meta-learning addresses this by enabling optimizers to learn from data how to adaptively solve new tasks, capturing task-specific structures, and accelerating convergence[16,17]. This is especially important in applications such as quantitative MRI reconstruction, where diverse data distributions[18] require robust and adaptable optimization strategies.

Optimization and meta-learning are increasingly intertwined with broader developments in decision-making under uncertainty. Recent works in predict-then-optimize frameworks[19], fair-

doi:10.20944/preprints202504.2285.v1

ness in resource allocation[20], and integrated estimation-optimization perspectives[21] highlight the importance of aligning learning models with downstream optimization goals[22–24].

5. Conclusion

In this work, we proposed a bilevel optimization framework for meta-learning, where the training process is viewed through the lens of optimal control theory. By employing the Method of Lagrangian Multipliers (MLM), we explicitly modeled both the primal reconstruction variables and the dual Lagrange multipliers, enabling principled updates that respect the underlying dynamics of the optimization process[25]. Our approach bridges classical optimization techniques with modern deep learning, offering a scalable and theoretically grounded method for learning adaptive optimization strategies.

This formulation not only improves the training stability but also enhances generalization across diverse tasks and domains, making it particularly suitable for challenging inverse problems such as quantitative MRI reconstruction. The equivalence established between direct loss minimization and Lagrangian-based optimization further validates the correctness and efficiency of our method.

Future work will explore extending this framework to more complex scenarios, such as multi-stage reconstruction, uncertainty-aware modeling, and real-time adaptation in dynamic environments. We believe that optimization-driven meta-learning approaches like ours hold strong potential for advancing learning systems in both medical imaging and broader scientific computing domains.

References

1. T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, “Meta-learning in neural networks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
2. M. Huisman, J. N. van Rijn, and A. Plaat, “A survey of deep meta-learning,” *Artificial Intelligence Review*, pp. 1–59, 2021.
3. C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, “Online meta-learning,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 1920–1930.
4. W. Bian, Y. Chen, and X. Ye, “An optimal control framework for joint-channel parallel mri reconstruction without coil sensitivities,” *Magnetic Resonance Imaging*, 2022.
5. D. Kiyasseh, A. Swiston, R. Chen, and A. Chen, “Segmentation of left atrial mr images via self-supervised semi-supervised meta-learning,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24*. Springer, 2021, pp. 13–24.
6. W. Bian, A. Jang, and F. Liu, “Multi-task magnetic resonance imaging reconstruction using meta-learning,” *Magnetic Resonance Imaging*, vol. 116, p. 110278, 2025.
7. Y. Chen, C.-B. Schönlieb, P. Liò, T. Leiner, P. L. Dragotti, G. Wang, D. Rueckert, D. Firmin, and G. Yang, “Ai-based reconstruction for fast mri—a systematic review and meta-analysis,” *Proceedings of the IEEE*, vol. 110, no. 2, pp. 224–245, 2022.
8. W. Bian, Y. Chen, X. Ye, and Q. Zhang, “An optimization-based meta-learning model for mri reconstruction with diverse dataset,” *Journal of Imaging*, vol. 7, no. 11, p. 231, 2021.
9. Q. Liu, Q. Dou, and P.-A. Heng, “Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*. Springer, 2020, pp. 475–485.
10. W. Bian, Q. Zhang, X. Ye, and Y. Chen, “A learnable variational model for joint multimodal mri reconstruction and synthesis,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 354–364.
11. W. Bian and Y. K. Tamilselvam, “A review of optimization-based deep learning models for mri reconstruction,” *AppliedMath*, vol. 4, no. 3, pp. 1098–1127, 2024.
12. A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, “Meta-learning with implicit gradients,” *Advances in neural information processing systems*, vol. 32, 2019.
13. S. Wang, J. Chen, X. Deng, S. Hutchinson, and F. Dellaert, “Robot calligraphy using pseudospectral optimal control in conjunction with a novel dynamic brush model,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6696–6703.

14. W. Bian, "Optimization-based deep learning methods for magnetic resonance imaging reconstruction and synthesis," Ph.D. dissertation, University of Florida, 2022.
15. ——, "A brief overview of optimization-based algorithms for mri reconstruction using deep learning," *arXiv preprint arXiv:2406.02626*, 2024.
16. Z. Ding, P. Li, Q. Yang, S. Li, and Q. Gong, "Regional style and color transfer," in *2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*. IEEE, 2024, pp. 593–597.
17. W. Bian, A. Jang, L. Zhang, X. Yang, Z. Stewart, and F. Liu, "Diffusion modeling with domain-conditioned prior guidance for accelerated mri and qmri reconstruction," *IEEE Transactions on Medical Imaging*, 2024.
18. Z. Ke, S. Zhou, Y. Zhou, C. H. Chang, and R. Zhang, "Detection of ai deepfake and fraud in online payments using gan-based models," *arXiv preprint arXiv:2501.07033*, 2025.
19. S. Verma, Y. Zhao, S. Shah, N. Boehmer, A. Taneja, and M. Tambe, "Group fairness in predict-then-optimize settings for restless bandits," in *The 40th Conference on Uncertainty in Artificial Intelligence*.
20. N. Boehmer, Y. Zhao, G. Xiong, P. Rodriguez-Diaz, P. D. C. Cibrian, J. Ngonzi, A. Boatin, and M. Tambe, "Optimizing vital sign monitoring in resource-constrained maternal care: An rl-based restless bandit approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 28, 2025, pp. 28 843–28 849.
21. A. N. Elmachtoub, H. Lam, H. Zhang, and Y. Zhao, "Estimate-then-optimize versus integrated-estimation-optimization: A stochastic dominance perspective," *arXiv preprint arXiv:2304.06833*, vol. 107, 2023.
22. W. Bian, P. Li, M. Zheng, C. Wang, A. Li, Y. Li, H. Ni, and Z. Zeng, "A review of electromagnetic elimination methods for low-field portable mri scanner," in *2024 5th International Conference on Machine Learning and Computer Application (ICMLCA)*. IEEE, 2024, pp. 614–618.
23. Z. Li, S. Qiu, and Z. Ke, "Revolutionizing drug discovery: Integrating spatial transcriptomics with advanced computer vision techniques," in *1st CVPR Workshop on Computer Vision For Drug Discovery (CVDD): Where are we and What is Beyond?*
24. Z. Ke and Y. Yin, "Tail risk alert based on conditional autoregressive var by regression quantiles and machine learning algorithms," in *2024 5th International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*. IEEE, 2024, pp. 527–532.
25. W. Bian, A. Jang, and F. Liu, "Improving quantitative mri using self-supervised deep learning with model reinforcement: Demonstration for rapid t1 mapping," *Magnetic Resonance in Medicine*, vol. 92, no. 1, pp. 98–111, 2024.

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.