Task-Specific Joint Learning in Inverse Problems: Efficient Deep Unrolling networks and Bilevel Optimization Methods

Project III (EC57003) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Integrated Bachelor and Master of Technology (Dual 5.Y)

in

Vision and Intelligent Systems

by

Kadagala Sai Siva Sankar

(20EC39017)

Under the supervision of

Professor **Subhadip Mukherjee**

Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2024-25

April 29, 2025

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: April 29, 2025 (Kadagala Sai Siva Sankar)

Place: Kharagpur (20EC39017)

ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Task-Specific Joint Learning in Inverse Problems: Efficient Deep Unrolling networks and Bilevel Optimization Methods** " submitted by **Kadagala Sai Siva Sankar** (Roll No. 20EC39017) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Integrated Bachelor and Master of Technology (Dual 5.Y) in Vision and Intelligent Systems is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2024-25.

29/04/25

Professor Subhadip Mukherjee
Electronics and Electrical
Communication Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Date: April 29, 2025

Place: Kharagpur

Abstract

Name of the student: **Kadagala Sai Siva Sankar** Roll No: **20EC39017**

Degree for which submitted: **Integrated Bachelor and Master of Technology (Dual 5.Y)**

Department: **Electronics and Electrical Communication Engineering**

Thesis title: **Task-Specific Joint Learning in Inverse Problems: Efficient Deep Unrolling networks and Bilevel Optimization Methods**

Thesis supervisor: ***Professor*** **Subhadip Mukherjee**

Month and year of thesis submission: **April 29, 2025**

Reconstruction of images from indirect and noisy measurements is a fundamental challenge in inverse problems. Traditionally, reconstruction and downstream tasks such as segmentation or classification are performed sequentially, often resulting in suboptimal performance due to error propagation. This thesis addresses the need for integrated pipelines that jointly optimize image reconstruction and task objectives. Two complementary approaches are proposed. First, a task adapted strategy is implemented by combining a computationally efficient variant of the Learned Primal-Dual (LPD) network with task-specific joint loss. Second, a bilevel optimization framework is explored, where image reconstruction is treated as a lower-level problem and task performance is optimized at the upper level using implicit differentiation. While the task adapted approach is applied to CT reconstruction tasks, the bilevel framework is demonstrated for denoising and classification problems. Experimental results highlight the effectiveness of both methods in improving task performance.

Acknowledgements

I would like to express my deepest gratitude to my M.Tech project guide, **Prof. Subhadip Mukherjee**, for providing me with such an engaging and intellectually stimulating problem statement. His commitment, valuable guidance, and regular weekly discussions have been pivotal in shaping this work. I am truly grateful for his constant support, insightful feedback, and encouragement throughout this semester, which have been instrumental in the successful progression of this project

Kadagala Sai Siva Siva Sankar
November, 2024
Indian Institute of Technology, Kharagpur

Contents

List of Figures

Abbreviations

CT	**C**omputed **T**omography
FBP	**F**iltered **B**ack**P**rojection
PDHG	**P**rimal–**D**ual **H**ybrid **G**radient
LPD	**L**earned **P**rimal–**D**ual
LSPD	**L**earned **S**tochastic **P**rimal–**D**ual
Sk-LSPD	**S**ketched **L**earned **S**tochastic **P**rimal–**D**ual
TV	**T**otal **V**ariation
HOAG	**H**yperparameter **O**ptimization **A**pproximate **G**radient
IFT	**I**mplicit **F**unction **T**heorem
MNIST	**M**odified **N**ational **I**nstitute **S**tandards **T**echnology
ODL	**O**perator **D**iscretization **L**ibrary
ASTRA	**A**STRA Toolbox
CNN	**C**onvolutional **N**eural **N**etwork
MSE	**M**ean **S**quared **E**rror
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
SSIM	**S**tructural **S**imilarity **I**ndex **M**easure
CG	**C**onjugate **G**radient
L-BFGS	**L**imited-memory **B**royden–**F**letcher–**G**oldfarb–**S**hanno

Chapter 1

Introduction

1.1 Background

X-ray computed tomography is an essential component in medical imaging, enabling visualization of internal anatomical structures. Traditional CT image reconstruction relies on analytic methods like Filtered back-projection (FBP), which, although computationally efficient, degrades in performance when faced with limited or noisy measurements. Sparse-view and low-dose imaging, motivated by the need to reduce radiation exposure, deepen the ill-posedness of the reconstruction problem, leading to artifacts and loss of diagnostic quality.

The last decade has witnessed a paradigm shift with the development of deep learning-based approaches for image reconstruction. Deep unrolling networks, such as the Learned Primal-Dual (LPD) network, combine model-based iterative schemes with learnable components, leveraging the expressive power of neural networks while retaining the physics of image formation. Despite their success, these networks face challenges of computational burden and scalability, particularly in high-dimensional (3D) imaging scenarios. Moreover, the conventional sequential pipeline, where reconstruction and downstream tasks (like segmentation or classification) are treated

separately, often leads to suboptimal performance. Integrating task-specific objectives directly into the reconstruction process through task adapted reconstruction or bilevel optimization frameworks offers a promising direction to overcome these limitations. This thesis focuses on developing efficient, scalable, and task-driven CT reconstruction frameworks, bridging the gap between image recovery and downstream decision-making.

1.2 Problem statement and research question

While deep learning has significantly advanced CT image reconstruction, challenges remain in ensuring efficiency, reliability, and task adaptability. Sequential pipelines that reconstruct images independently of downstream tasks risk optimizing for metrics that do not correlate with final objectives, like segmentation accuracy. Thus, the central research question of this thesis is: How can we design efficient, scalable deep learning-based CT reconstruction methods that are directly optimized for task-specific performance, overcoming the limitations of traditional sequential pipelines. We hypothesize that task adapted joint learning strategies and bilevel optimization frameworks, can significantly enhance reconstruction and downstream task outcomes compared to purely sequential approaches.

To address the research question, this thesis is structured as follows. Chapter 2 provides a comprehensive literature review covering classical and modern approaches to inverse problems, including filtered backprojection, deep unrolling networks, and bilevel optimization frameworks. Chapter 3 details the proposed methodology and experimental setup, including implementation of various reconstruction techniques such as Learned Primal-Dual (LPD), its stochastic and sketched variants, and their integration with downstream tasks through joint and bilevel learning. Chapter 4 presents the key insights and conclusions drawn from the experiments and highlights potential directions for future research.

Chapter 2

Literature Review

2.1 Filtered backprojection

Filtered Back Projection (FBP) is a core technique used in medical imaging, especially for reconstructing images in X-ray computed tomography (CT). The method starts by capturing projection data from various angles around the patient. This data is then processed using a high-pass filter, which helps to reduce low-frequency noise and unwanted artifacts while keeping important image details intact. Once filtered, the data is projected back across an image grid — essentially "smearing" the data along the paths it was originally collected. By repeating this process for each angle and combining the results, FBP reconstructs a full image of the scanned area. One of its biggest strengths is its speed, which is critical in clinical settings where fast image turnaround is essential. However, FBP does have its drawbacks, particularly when the available data is sparse or when the object being imaged strongly attenuates the X-rays, which can lead to artifacts and a drop in image quality.

2.2 Deep unrolling networks

The concept of unrolling originates from classical variational methods such as total variation regularization, which are typically solved with iterative solvers. Deep unrolling networks are a class of neural networks used primarily for inverse problems in image reconstruction and signal processing. They are designed to address complex optimization tasks by unrolling iterative algorithms into neural network architectures. These networks leverage the concept of ”unrolling” iterative algorithms, such as iterative optimization methods like gradient descent or alternating minimization, into a fixed number of steps. Each step of the algorithm is represented by a layer in the neural network, and the parameters of these layers are learned during training. By unrolling the iterative process into a neural network, deep unrolling networks can learn to solve inverse problems directly from data, without requiring explicit knowledge of the forward model or the inverse problem itself. This makes them highly flexible and adaptable to various tasks, including denoising, and medical image reconstruction. One significant advantage of deep unrolling networks is their ability to incorporate prior knowledge about the problem domain into the network architecture, allowing for improved performance and generalization. Additionally, they can handle ill-posed inverse problems more effectively compared to traditional optimization methods.

2.3 Learned primal dual

The saddle-point problem can be efficiently solved by the primal-dual hybrid gradient (PDHG) method, which is also known as the Chambolle-Pock algorithm (Chambolle and Pock (2010)) in the optimization literature. The PDHG method for solving the

saddle-point problem obeys the following updating rule:

Primal-Dual Hybrid Gradient (PDHG)

Initialize $x_0, \bar{x}_0 \in \mathbb{R}^d, \quad y_0 \in \mathbb{R}^p$

For $k = 0, 1, 2, \ldots, K$

$$y_{k+1} = \text{prox}_{\sigma f^*}(y_k + \sigma A \bar{x}_k);$$

$$x_{k+1} = \text{prox}_{\tau g}(x_k - \tau A^\top y_{k+1});$$

$$\bar{x}_{k+1} = x_{k+1} + \beta(x_{k+1} - x_k);$$

The PDHG algorithm takes alternatively the gradients regarding the primal variable x and dual variable y and performs the updates. The state-of-the-art unrolling scheme – learned primal-dual network (Adler and Oktem (2018)) is based on unfolding the iteration of PDHG by replacing the proximal operators with multilayer convolutional neural networks applied on the both primal and dual spaces. The step sizes at each steps are also set to be trainable.

Learned Primal-Dual (LPD)

Initialize $x_0 \in \mathbb{R}^d, \quad y_0 \in \mathbb{R}^p$

For $k = 0, 1, 2, \ldots, K-1$

$$y_{k+1} = D_\theta^k(y_k, \sigma_k, Ax_k, b);$$

$$x_{k+1} = P_\theta^k(x_k, \tau_k, A^\top y_{k+1});$$

where x is the primal variable, y is the dual variable, D_θ^k is the kth dualNet and P_θ^k is the kth PrimalNet. A, A^T are the forward and adjoint operators and σ_k, τ_k are the trainable step sizes for kth Dual and Primal networks respectively.

2.4 Learned stochastic primal dual and Sketched learned stochastic primal dual

In Learned stochastic primal dual (Tang et al. (2022b), we replace the forward and adjoint operators in the full-batch LPD network with only subsets of it. We partition the forward and adjoint operators into m subsets, and also the corresponding measurement data. In each layer, we use only one of the subsets, in a cycling order. The general framework of LSPD is:

Learned Stochastic Primal-Dual (LSPD)

Initialize $x_0 \in \mathbb{R}^d, \quad y_0 \in \mathbb{R}^{p/m}$

For $k = 0, 1, 2, \ldots, K-1$

$i = \mathrm{mod}(k, m)$;

(or pick i from $[0, m-1]$ uniformly at random)

$y_{k+1} = D_\theta^k(y_k, \sigma_k, (S_i A)x_k, S_i b)$;

$x_{k+1} = P_\theta^k(x_k, T_k, (S_i A)^\top y_{k+1})$;

where S := $[S_0, S_1, S_2, ..., S_{m-1}]$ are the set of sub-sampling operators. For the same number of layers, the LSPD network is approximately m-time more efficient than the full-batch LPD network in terms of computational complexity.

In the accelerated variant of LSPD (Tang et al. (2022a)), the main idea is to speedily approximate the products $Ax_k, A^T y_{k+1}$:

$$Ax_k \approx A_{s_k} S_{\theta_s^k}(x_k), A^T y_{k+1} \approx U_{\theta_u^k}(A^T y_{k+1})$$

The Sketched LPD network is written as:

Sketched-LPD

Initialize $x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^{p/m}$

For $k = 0, 1, 2, \ldots, K-1$

$i = \mod(k, m)$;

(or pick i from $[0, m-1]$ uniformly at random)

$y_{k+1} = D_{\theta_d^k}(y_k, \sigma_k, A_{s_k} S_{\theta_s^k}(x_k), b)$;

$x_{k+1} = P_{\theta_p^k}(x_k, \tau_k, U_{\theta_u^k}(A_{s_k}^T y_{k+1}))$;

Again, we can use the same approximation for stochastic gradient steps:

$$(S_i A)x_k \approx (S_i A_{s_k}) S_{\theta_u^k}(x_k),$$
$$(S_i A)^T y_{k+1} \approx U_{\theta_u^k}((S_i A_{s_k})^T y_{k+1}),$$

and hence we can write Sketched LSPD (SkLSPD) network as:

Sk-LSPD(option-2)

Initialize $x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^{p/m}$

For $k = 0, 1, 2, \ldots, K-1$

$i = \mod(k, m)$;

(or pick i from $[0, m-1]$ uniformly at random)

$y_{k+1} = D_{\theta_d^k}(y_k, \sigma_k, (S_i A_{s_k}) S_{\theta_s^k}(x_k), S_i b)$;

$x_{k+1} = U_{\theta_u^k}(P_{\theta_p^k}(S_{\theta_s^k}(x_k), T_k, (S_i A_{s_k})^T y_{k+1}))$;

In practice, we use the most simple off-the-shelf up/down-sampling operators in Pytorch for example the bilinear interpolation delivers excellent performance for the sketched unrolling networks. We use a "coarse-to-fine" strategy for skLPD and skLSPD. We use more aggressive sketch at the beginning for efficiency, while conservative sketch or non-sketch at latter iterations for accuracy. One possible choice is:

for the last few unrolling layers of SkLPD and SkLSPD, we switch to usual LPD/L-SPD (say if the number of unrolling layers is 12, we can choose last 4 unrolling layers to be unsketched, such that the reconstruction accuracy is best preserved.

2.5 Regularizers

Inverse problems, such as computed tomography (CT) reconstruction, are often characterized by their ill-posed nature. This property implies that small perturbations in the measured data can lead to significant deviations in the reconstructed solution, necessitating the incorporation of *apriori* information or constraints to stabilize the inversion process and guide the solution towards a physically plausible state. This process is formally known as regularization.

A widely adopted regularization functional in image reconstruction is the Total Variation (TV). TV regularization promotes solutions that are piecewise-smooth by penalizing the integrated magnitude of the image gradient. Mathematically, the discrete isotropic TV of an image x is typically defined as the sum of the magnitudes of the spatial gradients:

$$\mathrm{TV}(x) = \sum_{i,j} \sqrt{(\mathrm{diff}_x[i,j])^2 + (\mathrm{diff}_y[i,j])^2}$$

where $\mathrm{diff}_x[i,j]$ and $\mathrm{diff}_y[i,j]$ represent discrete approximations of the partial derivatives in the horizontal and vertical directions, respectively. TV is particularly effective at preserving sharp discontinuities (edges) while simultaneously suppressing noise within relatively homogeneous regions, making it valuable for applications like medical imaging. A significant challenge associated with the standard TV functional is its non-differentiability at points where the gradient is zero. This property complicates the direct application of gradient-based optimization algorithms, which are foundational to many modern machine learning and optimization frameworks, including deep learning.

To facilitate the use of gradient-based methods, smoothed variants of TV regularization are commonly employed. These variants introduce a small smoothing parameter to yield a differentiable approximation of the TV norm. This smoothing operation ensures that the gradient is well-defined everywhere, thereby enabling efficient computation of gradients via backpropagation. This characteristic is essential for integrating TV-like priors into deep learning architectures, such as unrolled optimization networks or bilevel optimization formulations where the regularizer's gradient is required.

Given an input tensor $x \in \mathbb{R}^{B\times C\times H\times W}$ (representing a batch of images with B batch elements, C channels, H height, and W width) and scalar parameters $\theta = (\theta_0, \theta_1) \in \mathbb{R}^2$, the forward differences are defined as:

$$\text{diff}_x[i,j] = x[i,j+1] - x[i,j] \quad \text{(horizontal difference)}$$

$$\text{diff}_y[i,j] = x[i+1,j] - x[i,j] \quad \text{(vertical difference)}$$

These differences are computed element-wise across batch and channel dimensions. The smoothed gradient magnitude at spatial location (i, j) for a given batch element and channel is then computed using the smoothing parameter e^{θ_1}:

$$\text{norm\_grad}[i,j] = \sqrt{(\text{diff}_x[i,j])^2 + (\text{diff}_y[i,j])^2 + e^{\theta_1}}$$

The term e^{θ_1} ensures that the term inside the square root is strictly positive, guaranteeing differentiability.

The Smoothed Total Variation regularizer, incorporating a learnable scaling factor e^{θ_0} and averaging over all elements, is formally defined as:

$$\text{Smoothed-TV}(x,\theta) = e^{\theta_0} \times \frac{1}{BCHW} \sum_{b=1}^{B}\sum_{c=1}^{C}\sum_{i=1}^{H}\sum_{j=1}^{W} \text{norm\_grad}_b^c[i,j]$$

The parameters θ_0 and θ_1 can be treated as hyperparameters or learned during an optimization process. Beyond conventional handcrafted regularizers like TV, contemporary research explores data-driven or learned regularizers. These methods leverage the representational power of neural networks to implicitly learn and impose complex priors directly from training data, offering potential for enhanced performance in diverse and intricate inverse problems.

2.6 Sequential learning

Sequential learning in the context of inverse problems refers to a process where the problem is decomposed into multiple stages, and different parts are solved in a specific order. In task-adapted reconstruction, for instance, a common sequential approach involves first performing the reconstruction of the model parameters from the noisy data, and then subsequently applying a task-specific operator to these reconstructed parameters. Another example is learned post-processing, where an initial reconstruction is obtained using a classical or learned method, and then a deep learning model is employed to further refine and improve the quality of this reconstruction. Sequential learning offers several advantages in the context of inverse problems. Its modular nature enhances understanding and facilitates debugging of individual components. This modularity also allows for the incorporation of domain expertise in the design of each specific step in the sequence. The flexibility of this approach enables the use of different techniques tailored to the requirements of each stage, allowing for the creation of customized solutions. Furthermore, sequential learning can leverage existing, well-established algorithms for reconstruction or feature extraction, building upon the vast body of research in traditional inverse problem-solving. Despite its benefits, sequential learning also presents certain disadvantages. Each step in the sequence is susceptible to introducing approximations that might not be properly accounted for by subsequent steps. This can lead to error propagation between stages, where inaccuracies in the reconstruction step can

negatively impact the performance of the task, and vice versa. Moreover, the reconstruction performed in the initial stages might not be optimally suited for the specific end task that needs to be accomplished. Similarly, if feature extraction is involved, it might not optimally consider the original measured data or the ultimate goal of the task.

2.7 End-to-End learning in inverse problems

End-to-end learning in the context of inverse problems refers to the approach of training a single model to perform the entire task, directly mapping from the raw input measurements to the final desired output, such as a reconstructed parameter or a task-specific result. This methodology bypasses the need for explicit intermediate steps like manual feature engineering or separate reconstruction stages. In task-adapted reconstruction, end-to-end learning involves directly learning the relationship between the observed data and the output of the task, such as a classification label or a segmentation mask. Deep neural networks are commonly employed to learn this direct and often complex mapping. The core idea behind end-to-end learning is to enable the model to automatically learn the optimal internal representations and mappings required to achieve the final goal, potentially capturing intricate dependencies that might be overlooked in sequential approaches. End-to-end learning offers several compelling advantages for tackling inverse problems. The model is directly optimized to minimize a loss function that is defined with respect to the final task, which can lead to superior performance compared to approaches where intermediate stages are optimized independently.

Despite these benefits, end-to-end learning also presents several disadvantages. A significant drawback is the high data requirement; these models typically need vast amounts of labeled training data to learn effectively. Without sufficient data, the

model might overfit to the training set and perform poorly on unseen data. Another major challenge is the lack of interpretability; the complex internal workings of deep neural networks often make it difficult to understand the reasoning behind the model's predictions. End-to-end models might also exhibit poor generalization if the test data significantly differs from the data used during training. They can also be sensitive to noise and perturbations in the input data, potentially leading to unstable and inaccurate reconstructions. Unlike analytical methods, incorporating prior domain knowledge or physical constraints explicitly into the network architecture or training process can be challenging.

2.8 Joint learning strategies

Inverse problems are fundamental in many scientific fields, involving estimating model parameters to match observed data. Traditionally, these problems are addressed using a sequential pipeline, where reconstruction and decision-making are treated as separate steps. However, this approach introduces approximations at each stage, often neglecting the final task during reconstruction and ignoring measured data during feature extraction. Integrating task-aware strategies into the reconstruction process can mitigate these issues by aligning intermediate steps with the end objective.

The paper *Task Adapted Reconstruction for Inverse Problems* (Adler et al. (2022)) introduces a joint learning paradigm to tackle these challenges. It focuses on estimating model parameters from noisy, indirect measurements while integrating reconstruction and task-specific decision-making into a single end-to-end process. The framework balances reconstruction fidelity with task performance using a shared loss function. Using neural networks ensures scalability and adaptability across various tasks and inverse problems, enhancing task accuracy, reducing computational overhead, and incorporating inherent regularisation for robust and interpretable results.

While effective, the joint learning approach lacks the mathematical flexibility of a bilevel framework (Crockett and Fessler (2022)), which explicitly separates the reconstruction and task performance objectives into a nested optimization structure. The bilevel setup addresses several challenges inherent in joint learning by decoupling reconstruction (lower level) and task-specific optimization (upper level). This separation allows reconstruction parameters to be optimized for fidelity while accounting for their downstream impact. Furthermore, the framework supports the inclusion of domain-specific regularizers at the lower level and directly optimizes task-related metrics at the upper level. It also provides greater control over hyperparameter tuning and systematically handles non-convex objectives, offering enhanced flexibility and robustness for complex inverse problems.

2.8.1 Notation

The upper-level loss function, denoted as $\ell(\gamma) \mapsto \mathbb{R}$ or $\ell(\gamma; \mathbf{x}^*) \mapsto \mathbb{R}$, is used as a fitness measure of γ. While ℓ is primarily a function of γ, it is often useful to express it with two inputs, where typically $\mathbf{x} = \hat{\mathbf{x}}(\gamma)$ represents the solution to the inner optimization problem (reconstructed image). The lower-level cost function, $\Phi(\mathbf{x}; \gamma) \mapsto \mathbb{R}$, is used for reconstructing an image, and is typically parameterized by γ. Regularization is represented by $R_\gamma(\mathbf{x}) \mapsto \mathbb{R}$, which incorporates prior information about the likely characteristics of an image. We use ∇ to denote the gradient of a real-valued function. If the function has multiple input arguments, ∇_i represents the gradient with respect to the argument i. Similarly, ∇^2 denotes the Hessian, and $\nabla^2_{i,j}$ represents the second-order differential with respect to variables i and j. In this context, the task labels are represented by s (e.g., segmentation masks or class labels), and the ground truth image/data is represented by $\mathbf{x}^*$. The noisy measurement of the image is denoted by $\mathbf{y}$. The solution to the inner optimization problem is $\hat{\mathbf{x}}(\gamma)$, and it is distinct from a generic $\mathbf{x}$. The task-specific loss is represented as $\ell(\gamma; \hat{x}(\gamma))$, and the reconstruction loss (lower-level loss) is represented by $\Phi(\mathbf{x}; \gamma)$,

with the regularization term $R_\gamma(\mathbf{x})$ governing the smoothness or prior structure of the solution.

2.8.2 Bilevel optimization framework

Many optimization problems, especially in hyperparameter tuning and inverse problems, aim to solve a bilevel optimization problem. The outer optimization seeks to find the optimal parameter γ^* that minimizes an objective function $\ell(\gamma; \hat{x}(\gamma))$, expressed as:

$$\hat{\gamma} = \arg\min_{\gamma \in \mathbb{F}^R} \ell(\gamma; \hat{x}(\gamma))$$

where $\hat{x}(\gamma)$ is the solution to an inner optimization problem. The inner problem minimizes another objective $\Phi(x; \gamma)$ with respect to x, for a given γ, defined as:

$$\hat{x}(\gamma) = \arg\min_{x \in \mathbb{F}^N} \Phi(x; \gamma)$$

Here, $\mathbb{F}^R$ represents the space of the parameters γ, and $\mathbb{F}^N$ represents the space of the variables x. This nested structure ensures that the outer optimization leverages the solution of the inner problem to achieve its objective.

The dataset comprises images, their corresponding noisy measurements, and corresponding task labels. We aim to develop an integrated pipeline that combines image reconstruction and task execution within a bilevel optimization framework. To implement this, we employ the Hyperparameter Optimization with Approximate Gradient algorithm (Pedregosa (2022)). This approach could be enhanced in the future by adopting a more robust algorithm to achieve improved performance(Salehi et al. (2024)).

2.8.2.1 Hyperparameter optimization with approximate gradient algorithm (HOAG)

When the lower-level optimization problem has a closed-form solution, $\hat{x}$, one can substitute that solution into the upper-level loss function. In this case, the bilevel problem is equivalent to a single-level problem and one can use classic single-level optimization methods to minimize the upper-level loss. But our project focuses on the more typical bilevel problems that lack a closed-form solution for $\hat{x}$. Using the chain rule, the gradient of the upper-level loss function with respect to the hyperparameters is

$$\nabla \ell(\gamma) = \nabla_\gamma \ell(\gamma; \hat{x}(\gamma)) + (\nabla_\gamma \hat{x}(\gamma))' \nabla_x \ell(\gamma; \hat{x}(\gamma))$$

To calculate $\nabla_\gamma \hat{x}(\gamma)$ we assume the gradient of Φ at the minimizer is zero. We use the implicit function theorem (IFT) perspective to arrive at the final expression.

2.8.2.2 Implicit differentiation

1. **Stationarity condition** For smooth and convex $\Phi(\mathbf{x}; \gamma)$, the solution $\hat{\mathbf{x}}(\gamma)$ satisfies the stationarity condition:

$$\nabla_{\mathbf{x}} \Phi(\hat{\mathbf{x}}(\gamma); \gamma) = 0$$

   This implicitly defines $\hat{\mathbf{x}}(\gamma)$ as a function of γ.

2. **Derivative of $\hat{\mathbf{x}}(\gamma)$ w.r.t. γ** Differentiating the stationarity condition with respect to γ:

$$\nabla^2_{\mathbf{x},\gamma} \Phi + \nabla^2_{\mathbf{x},\mathbf{x}} \Phi \cdot \frac{\partial \hat{\mathbf{x}}}{\partial \gamma} = 0$$

   If $\nabla^2_{\mathbf{x},\mathbf{x}} \Phi$ (the Hessian w.r.t. $\mathbf{x}$) is invertible:

$$\frac{\partial \hat{\mathbf{x}}}{\partial \gamma} = -\left(\nabla^2_{\mathbf{x},\mathbf{x}} \Phi\right)^{-1} \nabla^2_{\mathbf{x},\gamma} \Phi$$

3. **Gradient of $\ell(\gamma)$** Using the chain rule, the gradient of the upper-level loss $\ell(\gamma; \hat{\mathbf{x}}(\gamma))$ is given by:

$$\nabla \ell = \nabla_\gamma \ell + \left(\frac{\partial \hat{\mathbf{x}}}{\partial \gamma}\right)^\top \nabla_{\mathbf{x}} \ell$$

Substituting $\frac{\partial \hat{\mathbf{x}}}{\partial \gamma}$:

$$\nabla \ell = \nabla_\gamma \ell - \left(\nabla^2_{\mathbf{x},\gamma} \Phi\right)^\top \left(\nabla^2_{\mathbf{x},\mathbf{x}} \Phi\right)^{-1} \nabla_{\mathbf{x}} \ell$$

2.8.2.3 Implementation

To solve the bilevel optimization problem using implicit differentiation:

1. **Solve the inner problem:** Approximate the solution $\hat{\mathbf{x}}(\gamma)$ by minimizing the inner objective $\Phi(\mathbf{x}; \gamma)$ with respect to $\mathbf{x}$. The optimization should achieve ϵ-optimality, where the gradient norm or objective value changes fall below a predefined tolerance. Gradient-based methods, such as Adam or gradient descent or second order quasi-Newton methods like L-BFGS, can be used to balance computational efficiency and accuracy.

2. **Compute gradients:** Calculate the necessary gradients for implicit differentiation:

   - $\nabla_{\mathbf{x}} \ell$: The gradient of the outer objective ℓ with respect to $\mathbf{x}$, computed via chain rule.
   - $\nabla^2_{\mathbf{x},\gamma} \Phi$: The mixed partial derivative of the inner loss with respect to $\mathbf{x}$ and γ, evaluated as a vector-Jacobian product.
   - $\nabla^2_{\mathbf{x},\mathbf{x}} \Phi$: The Hessian of the inner objective with respect to $\mathbf{x}$, accessed efficiently using Hessian-vector products through PyTorch's `autograd.grad`. A Hessian-vector product is simply the product of the Hessian matrix with a vector. The key is that we often don't need the full Hessian matrix itself. We only need its action on a vector. We take advantage of this to reduce the expensive computation required to calculate full hessians.

3. **Solve linear system:** Solve the linear system $\nabla^2_{\mathbf{x},\mathbf{x}}\Phi \cdot \mathbf{z} = \nabla_{\mathbf{x}}\ell$ to compute $\mathbf{z}$. The conjugate gradient (CG) method is recommended for efficiency, as it avoids explicit Hessian formation:

   - Define $H(p)$, a function to compute $\nabla^2_{\mathbf{x},\mathbf{x}}\Phi \cdot p$, using Hessian-vector products.
   - Use CG (`conjugate_gradient`) to iteratively solve the system, stopping when the residual norm is below a specified tolerance. CG ensures scalability for high-dimensional problems by leveraging implicit Hessian computation and sparsity.

4. **Update γ:** Compute the gradient

$$\nabla\ell = \nabla_\gamma\ell - \left(\nabla^2_{\mathbf{x},\gamma}\Phi\right)^\top \left(\nabla^2_{\mathbf{x},\mathbf{x}}\Phi\right)^{-1} \nabla_{\mathbf{x}}\ell$$

   Where, the first term captures the indirect effect of γ on the outer loss through the inner solution while the second term represents the direct sensitivity of the inner objective to γ. Update γ using a projected gradient descent step:

$$\gamma \leftarrow \text{Project}(\gamma - \eta\nabla_\gamma\ell)$$

   where η is the learning rate, and `Project` ensures γ remains within feasible bounds (e.g., non-negativity or upper limits).

procedure HOAG($\{\epsilon^{(u)}, u = 1, 2, \ldots\}$, $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{x}^{(0)}$, $\boldsymbol{y}$)
  for u **do**=0,1,... ▷ Upper-level iteration counter
    $t = 0$ ▷ Lower-level iteration counter
    while $\|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)}) - \boldsymbol{x}^{(t)}(\boldsymbol{\gamma}^{(u)})\| \geq \epsilon^{(u)}$ **do**
      $\boldsymbol{x}^{(t+1)} = \Psi(\boldsymbol{x}^{(t)}; \boldsymbol{\gamma}^{(u)})$ ▷ Lower-level optimization step
      $t = t + 1$
    end while
    Compute gradient $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}^{(u)}; \boldsymbol{x}^{(t)})$ and
    Jacobian $\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}^{(t)}; \boldsymbol{\gamma}^{(u)})$
    Using CG, find $\boldsymbol{q}$ such that
    $\|\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}^{(t)}; \boldsymbol{\gamma}^{(u)})\boldsymbol{q} - \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}^{(u)}; \boldsymbol{x}^{(t)})\| \leq \epsilon^{(u)}$
    $\boldsymbol{g} = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}^{(u)}; \boldsymbol{x}^{(t)}) - \left(\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}^{(t)}; \boldsymbol{\gamma}^{(u)})\right)' \boldsymbol{q}$
    $\boldsymbol{\gamma}^{(u+1)} = \boldsymbol{\gamma}^{(u)} - \frac{1}{L}\boldsymbol{g}$ ▷ L is a Lipschitz constant of $\nabla \ell(\boldsymbol{\gamma})$
  end for
  return $\boldsymbol{\gamma}^{(u+1)}$
end procedure

FIGURE 2.1: Hyperparameter optimization with approximate gradient (Crockett and Fessler (2022))

Chapter 3

Methodology and Experimental results

3.1 Experiments with learned primal-dual

To evaluate the performance of the learned primal-dual (LPD) network for tomographic reconstruction, experiments were conducted on two distinct datasets: the MNIST dataset of handwritten digits as an initial proof-of-concept and the more complex Mayo-Clinic dataset of clinical CT scans.

3.1.1 Training on the MNIST dataset

As an initial validation of the LPD network's functionality, the MNIST dataset was utilized. This dataset comprises 28×28 grayscale images of handwritten digits. A subset of 51,200 samples was randomly selected and partitioned into training (80%, 40,960 samples) and testing (20%, 10,240 samples) sets.

Forward and adjoint operators were simulated using the ODL (Operator Discretization Library) toolbox, with the ASTRA toolbox providing GPU acceleration for

improved computational efficiency during training and inference. The projection geometry was configured with 24 angles uniformly sampled in the range $(0, \pi)$, simulating a parallel beam setup. Ground truth images from the dataset were forward-projected using ODL to generate corresponding sinograms. A custom dataset class was implemented to pair the ground truth images with their simulated sinograms for training and testing.

The learned primal-dual algorithm was implemented using the PyTorch framework, integrating deep convolutional neural networks (CNNs) within both the primal (image) and dual (data) spaces. These networks are iteratively connected via the simulated forward operator and its adjoint. The training objective was to minimize the mean squared error (MSE) between the reconstructed images and the ground truth images. Key hyperparameters for the MNIST experiment included the use of 5 DualNet and 5 PrimalNet blocks. The batch size was set to 1024. The initial learning rate was 0.001. Each individual dual and primal network block consisted of 3 convolutional layers followed by a PReLU activation function. A skip connection was implemented, adding the initial input channel directly to the output of the block. The convolutional layers had 32 output channels and a kernel size of 3×3. Momentum was omitted for simplicity and reduced memory footprint. The dual variable was initialized to a tensor of zeros.

During training, the model's performance was periodically evaluated on the test set. The model weights corresponding to the epoch with the lowest test loss were saved. The Peak Signal-to-Noise Ratio (PSNR) metric was employed to quantitatively assess the reconstruction quality. PSNR values were calculated by comparing the LPD network's predictions to the ground truth images, and also by comparing traditional filtered backprojection (FBP) reconstructions (generated from the simulated sinograms) to the ground truth images.

Quantitative evaluation on 5 random test samples yielded an average PSNR of 30.18 dB for the LPD model's reconstructions, significantly surpassing the average PSNR

of 10.23 dB achieved by the filtered backprojection method on the same samples. An illustrative example demonstrating the visual performance is shown in Figure 3.1.

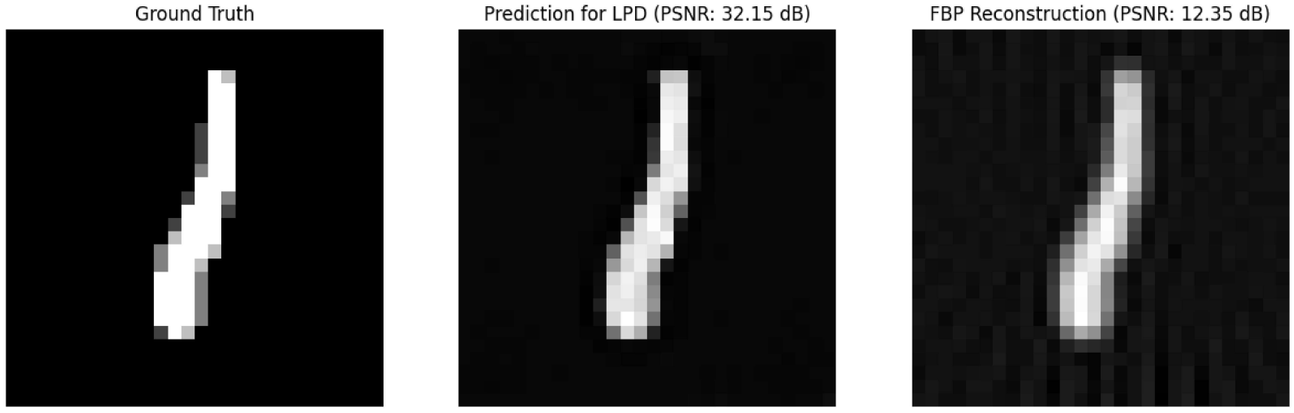

FIGURE 3.1: Example of LPD network performance on MNIST dataset compared to ground truth and FBP.

3.1.2 Training on the Mayo-Clinic dataset

Following the successful validation on the simpler MNIST dataset, the LPD network was applied to the more challenging Mayo-Clinic dataset, which consists of clinical 2D CT image slices sized 512×512. The dataset comprises 2111 slices, split into 80% for training (1689 slices) and 20% for testing (422 slices).

Similar to the MNIST experiment, the ODL toolbox with Astra CUDA implementation was used to simulate parallel beam projections. A sparse-view scenario was created by simulating projections with 90 equally spaced angles in the range $(0, \pi)$. A dedicated dataset class was developed to provide triplets of sinograms, corresponding filtered backprojection (FBP) images (often used as an initial guess in model-based methods), and ground truth images. A sample image slice from this sparse-view dataset setup is shown in Figure 3.2.

The LPD model architecture employed for the Mayo-Clinic dataset was similar to the MNIST experiment but with specific modifications tailored for the increased

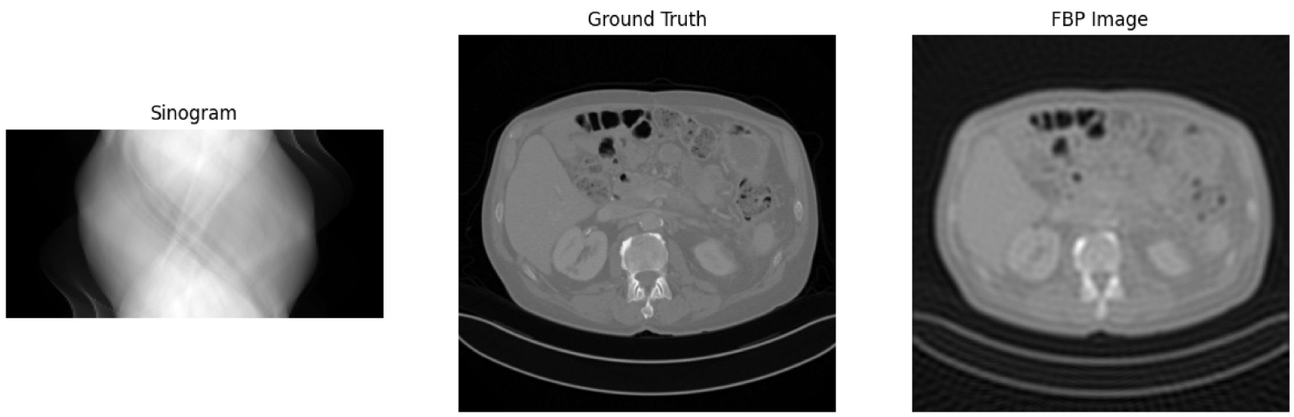

FIGURE 3.2: A sample slice from the sparse-view Mayo-Clinic dataset used for training and testing.

complexity and resolution. The number of DualNet and PrimalNet blocks was increased to 12. The kernel size for the convolutional layers within the networks was set to 5×5. The initial step sizes for the primal and dual updates, often denoted as τ and σ in primal-dual algorithms, were initialized to 0.01. Due to the larger image size, the batch size was reduced to 1. The Adam optimizer was used for training with an MSE loss function. The initial learning rate was set to 5e-5, and the β parameters for the Adam optimizer were $(0.5, 0.99)$. The network was trained for 15 epochs.

After training, the network's performance was evaluated on 100 samples from the test data. The average PSNR value for the LPD reconstructions was 38.58 dB. In comparison, the FBP reconstructions of the same test samples achieved an average PSNR of 28.06 dB. These results demonstrate a significant improvement in reconstruction quality achieved by the LPD network over traditional FBP for sparse-view CT reconstruction on clinical data. A visual comparison illustrating the performance difference between LPD and FBP is presented in Figure 3.3.

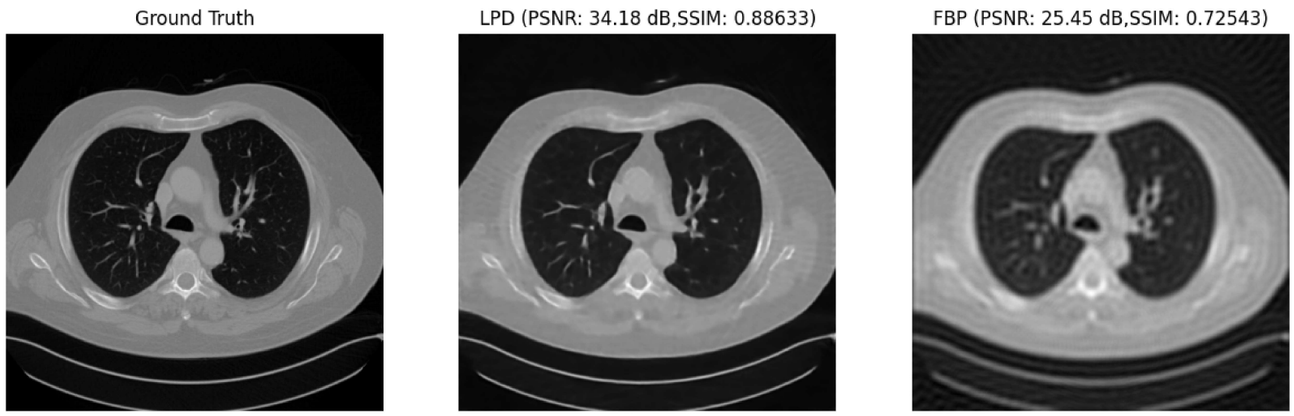

FIGURE 3.3: Comparison of reconstruction results from the LPD network and filtered backprojection (FBP) on a sample from the Mayo-Clinic test set.

3.2 Learned stochastic and Sketched primal-dual variants on Mayo-Clinic data

Building upon the functional LPD network model, this section presents the implementation and evaluation of two advanced architectures proposed in related literature: Learned Stochastic Primal-Dual (LSPD) and Sketched-LSPD (sk-LSPD). These variants aim to improve efficiency, particularly for large-scale problems, and are evaluated on the Mayo-Clinic dataset.

3.2.1 LSPD and Sketched-LSPD under sparse-view conditions

The Learned Stochastic Primal-Dual (LSPD) network modifies the full-batch LPD network Tang et al. (2022a) by replacing the full forward and adjoint operators with subsets thereof. In our implementation, we partitioned the operators and the corresponding measurement data into 4 subsets. Within each unrolling layer of the network, only one of these subsets is utilized in a cycling order. This approach involves creating four distinct operators corresponding to these sampled geometries

and using a list of these sampled forward and adjoint operators dynamically in the network.

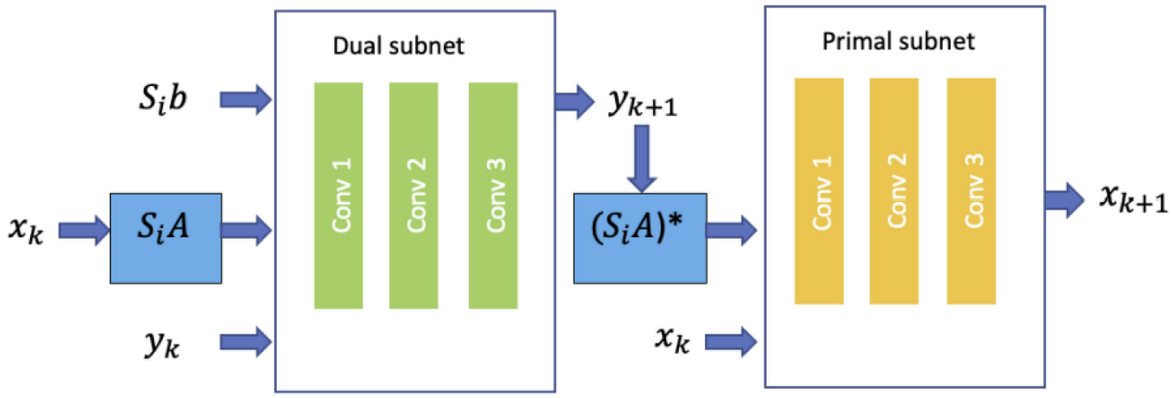

FIGURE 3.4: Illustrative structure of a single layer within the LSPD network, showcasing the use of a subset of the operator (indicated by A_i). Source: Tang et al. (2022a)

For the LSPD network trained under sparse-view conditions, we employed 12 DualNet and 12 PrimalNet blocks. A batch size of 1 was used. The hyperparameters, loss function (MSE), and optimizer (Adam with learning rate 5e-5 and betas (0.5, 0.99)) were kept consistent with the parameters used for the standard LPD training on the Mayo-Clinic dataset as described in the previous section. The LSPD network was trained for 15 epochs.

The Sketched-LSPD (sk-LSPD) variant extends the stochastic idea by operating on sketched (reduced-dimension) representations of the data in early layers. For implementing Sk-LSPD, we utilized sampled forward and adjoint operators, conceptually similar to those in LSPD, but with the modification that the input dimensions for the sketching operation were reduced from 512×512 to 256×256. This setting reduces the computational cost for the sketched operator A_s to approximately one-quarter compared to the full operator A on the 512×512 grid. We implemented sk-LSPD using "option 2" as described in the original paper Tang et al. (2022a), which employs a coarse-to-fine sketching strategy. This approach means that the sketching

operation is used in initial layers, while the final four layers (out of 12 blocks) utilize the unsketched (full-dimension) LSPD forward and adjoint operators. Bilinear upsampling and downsampling functions from PyTorch were chosen for dimension transitions between sketched and unsketched representations. The networks again comprised 12 DualNet and 12 PrimalNet blocks and were trained for 15 epochs with a batch size of 1. The initial guess x_0 for all unrolling networks (LPD, LSPD, Sk-LSPD) was set to the standard filtered backprojection (FBP) reconstruction from the corresponding sinogram.

A quantitative comparison of FBP, LPD, LSPD, and Sk-LSPD on the sparse-view Mayo-Clinic test set is presented in Table 3.1.

TABLE 3.1: Performance metrics for FBP, LPD, LSPD, and Sk-LSPD networks on the sparse-view Mayo-Clinic test dataset.

Method	PSNR	SSIM	GPU Inference time per sample
FBP	28.06 dB	0.82329	4.888 ms
LPD	38.58 dB	0.94332	322.345 ms
LSPD	38.53 dB	0.94281	233.519 ms
Sk-LSPD	36.93 dB	0.93183	183.077 ms

Table 3.1 indicates that LSPD achieves reconstruction quality (PSNR, SSIM) very close to that of the full-batch LPD network, with a marginal reduction in training and inference times per sample. Sketched-LSPD, while exhibiting a slight decrease in PSNR and SSIM compared to LPD and LSPD, demonstrates substantial improvements in both training and inference times per sample, highlighting the computational benefits of the sketching approach. Visual comparison examples for LPD, LSPD, and FBP are provided in Figure 3.5.

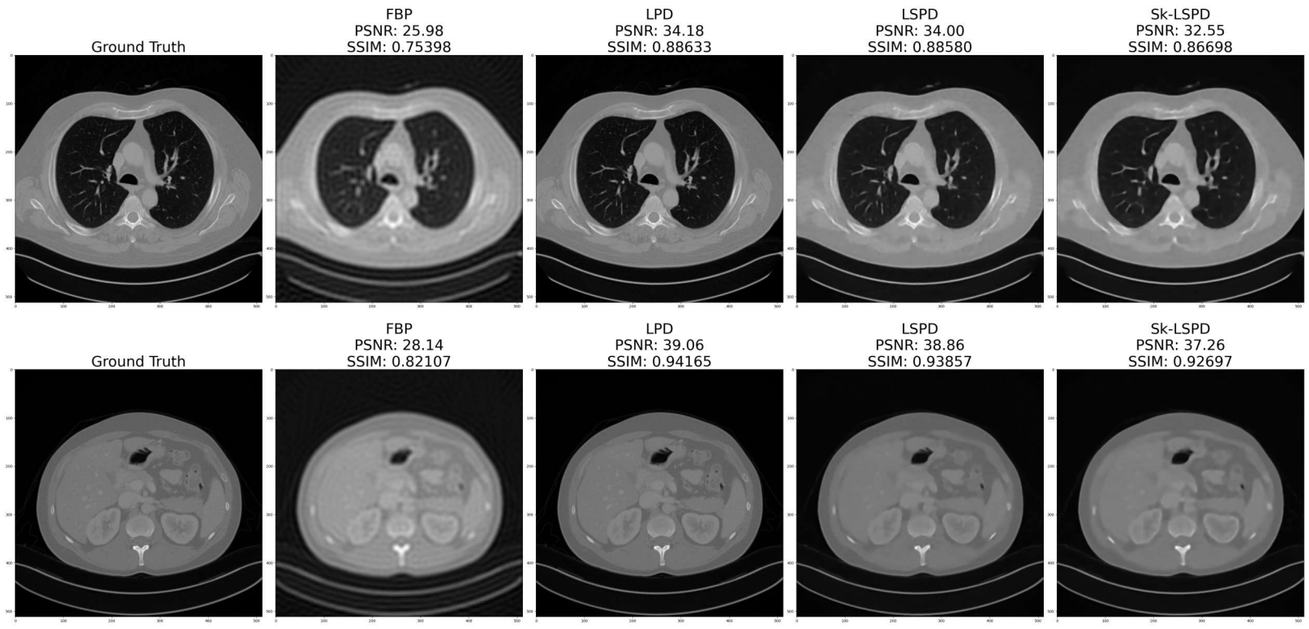

FIGURE 3.5: Examples comparing reconstructions from FBP, LPD, and LSPD networks under sparse-view conditions. Note that LSPD achieves reconstruction performance comparable to the full-batch LPD.

3.2.2 Evaluation of Sketched-LSPD under Sparse-View and Low-Dose Conditions

To further investigate the robustness of the Sketched-LSPD approach under more challenging clinical conditions, an experiment was conducted on the Mayo-Clinic CT dataset simulating a low-dose setting in addition to the sparse views. Low-dose CT is prevalent in clinical practice due to concerns regarding patient exposure to ionizing radiation, but it results in measurement data corrupted by significant noise, posing difficulties for accurate reconstruction.

To simulate low-dose CT scans, the sparse-view parallel-beam CT measurements were corrupted with Poisson noise. The noise model used was $b \sim \text{Poisson}(I_0 \exp(-Ax^*))$, where A is the forward operator, x^* is the ground truth image, and I_0 is the incident photon count. A low-dose level was simulated by setting $I_0 = 7 \times 10^4$. This Poisson noise was applied to the same sparse-view dataset derived from the Mayo-Clinic images as used in the previous subsection. A sample sinogram corrupted with this noise model is shown in Figure 3.6.

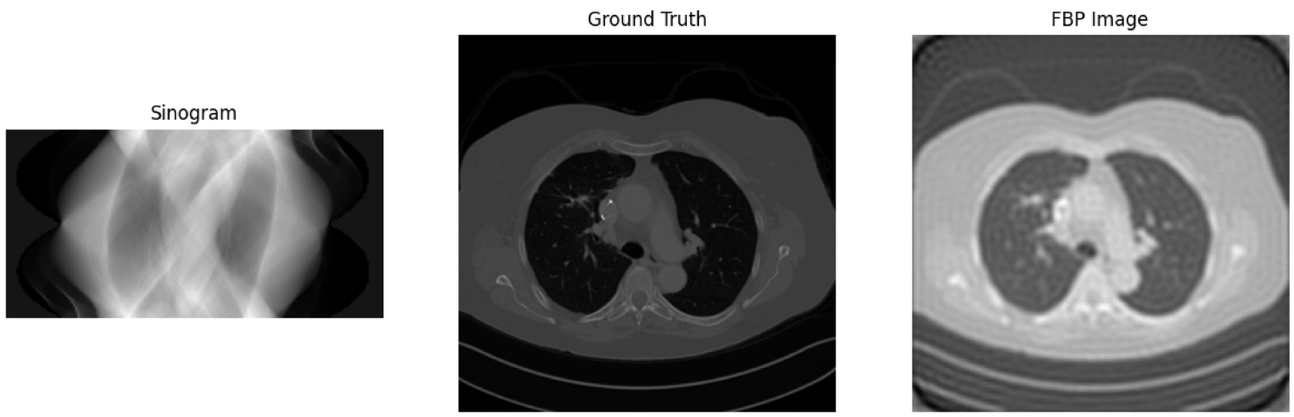

FIGURE 3.6: A sample sinogram from the Mayo-Clinic dataset corrupted with Poisson noise, simulating a sparse-view, low-dose CT acquisition.

The Sketched-LSPD network, configured with the same architecture and hyperparameters as described in the previous subsection (12 blocks, batch size 1, etc.), was trained specifically on this sparse-view, low-dose dataset for 15 epochs using the MSE loss. Visual inspection of the reconstructions demonstrates the Sk-LSPD network's capability to produce visually coherent and noise-reduced images even under these challenging measurement conditions. An example illustrating the network's performance in this sparse-view and low-dose setting is presented in Figure 3.7.

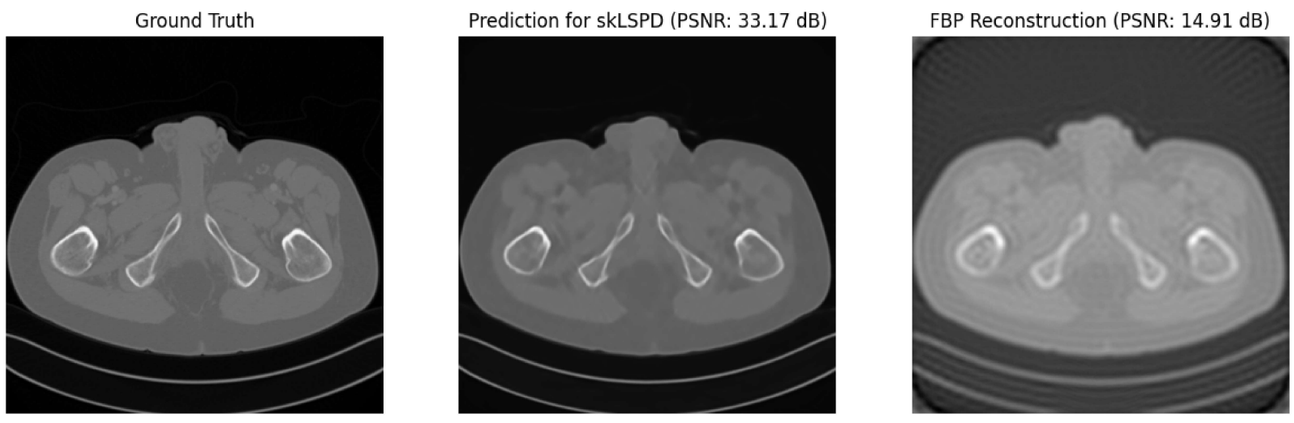

FIGURE 3.7: Example of Sketched-LSPD network reconstruction performance in a combined sparse-view and low-dose CT setting.

3.3 Task adapted reconstruction using joint loss

Errors introduced during the reconstruction phase can propagate and negatively impact the downstream task accuracy. Joint learning approaches aim to address this by simultaneously optimizing both the reconstruction and tasks within a single framework, allowing them to benefit from each other. This section details experiments conducted to evaluate the effectiveness of different approaches on liver tumour segmentation task on Medical Decathlon dataset (Simpson et al., 2019).

We compare a standard sequential pipeline, a basic segmentation model on clean images as a baseline, and joint learning frameworks incorporating different reconstruction techniques: Learned primal-dual (LPD) and Learned Stochastic Primal-Dual (LSPD). The primary objective is to demonstrate that joint learning, specifically when integrated with LSPD reconstruction method, provides performance similar to LPD which is computationally more expensive.

Each model was trained for 15 epochs. Performance was evaluated using the Dice coefficient on a separate test set. We evaluated four distinct approaches:

3.3.1 Basic UNet on Clean Images (Baseline)

This approach serves as a baseline, representing the segmentation performance achievable with ground truth data. A standard U-Net architecture was implemented and trained directly using clean images and their corresponding ground truth segmentation masks. The segmentation network was trained using a Dice loss function, which measures the overlap between the predicted and ground truth segmentation masks.

3.3.2 Sequential Pipeline

This method follows a traditional two-step process. First, the limited-view sinograms were used to reconstruct the images by training an LPD model for reconstruction. Following training of the reconstruction model, a separate U-Net model, similar to the one used in the baseline, was applied to the reconstructed images generated using the previously trained LPD model to perform segmentation. The segmentation network in this pipeline was also trained using a Dice loss function.

3.3.3 Joint Learning with LPD network

This approach implements a joint learning framework where image reconstruction and segmentation are optimized simultaneously. It integrates a Learned primal-dual (LPD) network for reconstruction with a segmentation network (likely a U-Net). The training process involved using a joint loss function, which is a weighted sum of a reconstruction loss (e.g., Mean Squared Error loss comparing the reconstructed image to the ground truth image) and the Dice loss for the segmentation masks. This allows the two tasks to influence each other during training.

3.3.4 Joint Learning with LSPD network

Similar to the LPD joint learning method, this approach simultaneously optimizes reconstruction and segmentation. The key difference is the use of Learned Stochastic Primal-Dual (LSPD) network for the reconstruction component. LSPD incorporates regularization within the learned primal-dual framework. The training is done using a combined loss function, consisting of a reconstruction loss (using the LSPD output) and the Dice loss for the segmentation output.

The performance of each approach, measured by the Dice coefficient on the test set after 15 epochs of training, is summarized in the table below:

TABLE 3.2: Performance Comparison of Different Approaches

Approach	Dice Coefficient
Basic UNet on Clean Images (Ceil)	0.8224
Joint Learning with LPD	0.7866
Joint Learning with LSPD (Our Model)	0.7762
Sequential Pipeline (Floor)	0.7624

The Basic UNet on clean images, as expected, achieved the highest Dice coefficient, the upper bound achievable without limited-view reconstruction challenges. The Sequential Pipeline yielded the lowest performance, supporting the idea that decoupling reconstruction and segmentation propagates errors and harms accuracy. Joint Learning approaches significantly outperformed the sequential pipeline, demonstrating the benefit of simultaneous optimization. This demonstrates that joint learning loss functions are indeed effective when combined with LSPD and achieve a similar performance compared to LPD joint learning method.

3.4 Bilevel Joint Learning Pipeline

Let T^* be a pretrained task operator trained on the clean dataset. We consider a dataset (x^*, y, s), where x^* represents the ground truth data, y denotes the noisy measurements, and s represents the task labels. Let R_γ be a regularizer parameterized by γ. In our case, we use the smoothed total variation (Salehi et al. (2024)) denoising regularizer. The choice of regularizer is problem-specific, and since we are addressing a denoising problem, the smoothed total variation regularizer is chosen for its ability to promote smoothness in the reconstructed data.

Inner Level Problem:

$$\hat{x}(\gamma) = \arg\min_y \|y - x^*\|_2^2 + R_\gamma(y)$$

This equation represents a denoising problem.

Outer Level Problem:

$$\hat{\gamma} = \arg\min_{\gamma} \ell(T^*(\hat{x}(\gamma)), s)$$

Where $\ell(\cdot, \cdot)$ is the loss function between the task operator output and the task labels. This pipeline can be optimized further by finetuning the pretrained task operator using the outer loop loss.

3.4.1 Image denoising and segmentation using Cityscapes Dataset

We utilize the Cityscapes dataset (Cordts et al. (2016)) for segmentation tasks, applying preprocessing steps such as resizing, normalization, and data augmentation to prepare the data for training and validation. To implement the bilevel optimization framework, we designed several helper functions. We utilize the conjugate gradient method to efficiently solve systems involving the Hessian matrix, avoiding explicit computation and thereby reducing memory and computational overhead. Additionally, we calculate Hessian-vector products using PyTorch's `autograd`, which leverages automatic differentiation to approximate Hessian operations effectively, even for high-dimensional parameter spaces. The inner optimization incorporates regularization to enhance image reconstruction. We have used smoothed total variation (TV), which penalizes abrupt changes in pixel intensity to enforce spatial smoothness. The parameters of these regularizers are treated as hyperparameters that are optimized during the outer loop.

Our implementation of the Hyperparameter Optimization via Approximate Gradients algorithm uses a bilevel optimization structure consisting of two levels:

3.4.1.1 Inner Optimization

The **inner optimization** reconstructs noisy images by minimizing a combination of reconstruction error and the smoothed total variation (TV) regularization. The objective is formulated as:

$$\hat{x}(\gamma) = \arg\min_{y} \left\{ \|y - x^*\|_2^2 + \text{Smoothed TV}(y, \gamma) \right\}$$

Where Smoothed TV(y) represents the smoothed total variation regularizer, which is defined as:

$$\text{Smoothed TV}(y, \gamma) = \exp(\gamma_0) \cdot \frac{1}{N} \sum_{i,j} \sqrt{\text{TV}(y)^2 + \exp(\gamma_1)^2}$$

where,

$$\text{TV}(y) = \sum_{i,j} \sqrt{(y_{i+1,j} - y_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2}$$

Here, $y_{i,j}$ denotes the pixel intensity at position (i, j) in the image, and the TV regularizer penalizes abrupt changes in pixel intensities, promoting smooth transitions and preserving edges in the image. The Total Variation (TV) regularizer is a popular choice for image denoising and other inverse problems because it promotes piecewise smoothness. It penalizes sharp changes in intensity while preserving important edge information. However, the standard TV regularizer is non-differentiable, which can be problematic for optimization algorithms that rely on gradients. The smoothed TV regularizer addresses this non-differentiability. It approximates the absolute value function (used in standard TV) with a smooth function, typically something like a Huber function or a hyperbola. This makes it suitable for gradient-based optimization.

3.4.1.2 Outer Optimization

The **outer optimization** refines the hyperparameters (γ) by minimizing a composite segmentation loss. The objective is given by:

$$\hat{\gamma} = \min_{\gamma} \{\mathcal{L}_{\text{segmentation}}(T^*(\hat{x}(\gamma)), \text{s})\}$$

Where the segmentation loss $\mathcal{L}_{\text{segmentation}}$ is defined as:

$$\mathcal{L}_{\text{segmentation}} = L_{\text{CE}} + L_{\text{Dice}} + L_{\text{IoU}} + L_{\text{Log-Cosh-Dice}}$$

Where L_{CE} denotes Cross-entropy loss for pixel-wise classification accuracy, L_{Dice} denotes Dice loss to measure the overlap between predicted and ground truth masks, L_{IoU} denotes Intersection-over-union loss to improve segmentation quality, $L_{\text{Log-Cosh-Dice}}$ denotes Log-cosh Dice loss for robustness to large deviations. The outer optimization adjusts the hyperparameters to optimize for better task-specific performance, balancing reconstruction quality and segmentation accuracy.

3.4.1.3 HOAG with Smoothed TV and Fine-Tuning

In an extended implementation, we combine the smoothed TV regularizer with fine-tuning of specific layers in the segmentation model. After initializing the segmentation model with pretrained weights, we allow fine-tuning of the final two upsampling layers and the output layer, while freezing the remaining layers. This approach integrates the benefits of the smoothed TV regularizer for reconstruction and the flexibility of fine-tuning for task-specific adaptation. The Adam optimizer is used to train these layers during the outer optimization, ensuring improved performance in challenging segmentation scenarios. By alternating between inner reconstruction and outer fine-tuning steps, this configuration leverages both reconstruction-based priors and task-specific learning for superior results. In our experiments, we utilized

a pretrained U-Net model as the task operator and generated a blurred dataset by introducing Gaussian noise with varying standard deviations. The training set comprised 100 samples of size 64x64, while the test set consisted of 500 samples. The inner optimization was solved for two tolerance values: 10^{-5} and 10^{-3}. The results obtained are presented in the table 3.3.

	Tolerance = 10^{-5}						Tolerance = 10^{-3}	
	= 0.1		= 0.4		= 0.8		= 0.8	
	dice	IOU	dice	IOU	dice	IOU	dice	IOU
T* on clean images	0.8539	0.7520	0.8539	0.7520	0.8539	0.7520	0.8539	0.7520
T* on noisy images	0.6504	0.4896	0.2662	0.1546	0.1764	0.0971	0.1754	0.0965
HOAG with smoothed TV	0.6786	0.5179	**0.8538**	**0.7518**	**0.8457**	**0.7398**	0.2079	0.1165
HOAG with Finetuning	**0.7239**	**0.5746**	0.8379	0.7292	0.8343	0.7239	**0.6213**	**0.4571**

TABLE 3.3: Semantic Segmentation Task using Cityscapes Dataset

3.4.2 Image denoising and classification using Stanford Dogs Dataset

We utilize the Stanford Dogs dataset Khosla et al. (2011) for the classification task, applying preprocessing steps such as resizing, normalization, and data augmentation to prepare the data for training and validation. This dataset has over 20,580 images with 120 breeds of dogs. We have used an off-the-shelf Pytorch resnet50 model as a baseline.

3.4.2.1 Inner Optimization

The **inner optimization** reconstructs noisy images by minimizing a combination of reconstruction error and the smoothed total variation (TV) regularization. The objective is formulated as:

$$\hat{x}(\gamma) = \arg\min_{y} \left\{ \|y - x^*\|_2^2 + \text{Smoothed TV}(y, \gamma) \right\}$$

Where Smoothed TV(x) represents the smoothed total variation regularizer.

3.4.2.2 Outer Optimization

The **outer optimization** refines the hyperparameters (γ) by minimizing categorical crossentropy loss. The objective is given by:

$$\hat{\gamma} = \min_{\gamma} \{\mathcal{L}_{\text{crossentropy}}(T^*(\hat{x}(\gamma)), \text{s})\}$$

The training set comprised 100 samples of size 64x64, while the test set consisted of 500 samples. The inner optimization was solved for two tolerance values: 10^{-5} and 10^{-3}. The results obtained are presented in the table 3.4.

	Tolerance = 10^{-5}			Tolerance = 10^{-3}		
	= 0.1	= 0.4	= 0.8	= 0.1	= 0.4	= 0.8
T* on clean images	48.8	48.8	48.8	48.8	48.8	48.8
T* on noisy images	16.6	5.2	2.4	15.2	4.2	2.4
HOAG with smoothed tv	20.2	**46**	**46.4**	13.3	19.5	34.6
HOAG with finetuning	**37.0**	39.6	39.8	23.2	27.2	**36.2**

TABLE 3.4: Classification Task using Stanford Dogs Dataset

Chapter 4

Conclusion

4.1 Key Insights

The experimental results presented in this thesis yield several important insights. The Learned Primal-Dual (LPD) network demonstrated strong reconstruction performance in sparse-view CT, significantly improving PSNR over traditional filtered backprojection methods. The Learned Stochastic Primal-Dual (LSPD) variant retained comparable accuracy to LPD while reducing computational cost by approximately 28%, showcasing its scalability. Furthermore, the Sketched-LSPD approach offered a practical trade-off between speed and reconstruction quality by applying aggressive downsampling in early layers and switching back to full resolution in later iterations. Joint learning frameworks, which integrate reconstruction with downstream tasks such as segmentation, consistently outperformed sequential pipelines, with the LPD and LSPD joint models yielding higher Dice scores than their decoupled counterparts. The bilevel optimization framework using HOAG effectively tuned regularization parameters to enhance segmentation performance, particularly under noisy conditions, and further gains were achieved through fine-tuning of task-specific layers. This framework also generalized well to classification tasks, as demonstrated by accuracy improvements on the Stanford Dogs dataset. Overall, these

findings confirm the benefits of jointly optimizing reconstruction and task objectives, and highlight the efficiency gains achievable through stochastic and bilevel methods.

4.2 Future Work

Based on the preliminary results, these approaches demonstrate significant potential for task-adapted joint learning in inverse problems. The ultimate goal of this research is to develop a scalable, three-dimensional joint reconstruction framework that is directly optimized for downstream tasks such as segmentation or classification. A key direction for future work is the integration of advanced regularizers based on neural networks, which can learn complex image priors directly from data and offer greater adaptability than traditional handcrafted regularizers like Total Variation. Moreover, extending the bilevel optimization framework to real-world medical imaging tasks will be crucial, especially in high-resolution 3D CT and MRI scenarios where computational efficiency and robustness are paramount. To support this, we plan to explore more efficient and numerically stable bilevel optimization algorithms tailored for large-scale imaging problems. In the long term, the aim is to build a general-purpose, modular pipeline in which only the task-specific loss function needs to be defined, allowing easy adaptation to a wide variety of clinical and scientific applications.

Bibliography

1. Adler, J., Lunz, S., Verdier, O., Schönlieb, C.-B., and Öktem, O. (2022). Task adapted reconstruction for inverse problems. *Inverse Problems*, 38(7):075006.

2. Adler, J. and Oktem, O. (2018). Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332.

3. Chambolle, A. and Pock, T. (2010). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145.

4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding.

5. Crockett, C. and Fessler, J. A. (2022). Bilevel methods for image reconstruction. *Foundations and Trends® in Signal Processing*, 15(2–3):121–289.

6. Khosla, A., Jayadevaprakash, N., Yao, B., and Fei-Fei, L. (2011). Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO.

7. Mukherjee, S., Hauptmann, A., Öktem, O., Pereyra, M., and Schönlieb, C.-B. (2023). Learned reconstruction methods with convergence guarantees: A survey of concepts and applications. *IEEE Signal Processing Magazine*, 40(1):164–182.

8. Pedregosa, F. (2022). Hyperparameter optimization with approximate gradient.

9. Salehi, M. S., Mukherjee, S., Roberts, L., and Ehrhardt, M. J. (2024). An adaptively inexact first-order method for bilevel optimization with application to hyperparameter learning.

10. Simpson, A. L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., Kopp-Schneider, A., Landman, B. A., Litjens, G., Menze, B., Ronneberger, O., Summers, R. M., Bilic, P., Christ, P. F., Do, R. K. G., Gollub, M., Golia-Pernicka, J., Heckers, S. H., Jarnagin, W. R., McHugo, M. K., Napel, S., Vorontsov, E., Maier-Hein, L., and Cardoso, M. J. (2019). A large annotated medical image dataset for the development and evaluation of segmentation algorithms.

11. Tang, J., Mukherjee, S., and Schönlieb, C.-B. (2022a). Accelerating deep unrolling networks via dimensionality reduction.

12. Tang, J., Mukherjee, S., and Schönlieb, C.-B. (2022b). Stochastic primal-dual deep unrolling.

MTP_2_first_draft-10-46.pdf

ORIGINALITY REPORT

15%	12%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	web.archive.org Internet Source	3%
2	deepblue.lib.umich.edu Internet Source	2%
3	deepai.org Internet Source	2%
4	arxiv.org Internet Source	2%
5	Submitted to University of Wollongong Student Paper	1%
6	Crockett, Caroline E.. "How Students and Algorithms Learn to Filter: Investigating Students' Understanding of Signal Processing Concepts and Bilevel Methods for Learning Filters for Image Reconstruction", University of Michigan, 2022 Publication	1%
7	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2020", Springer Science and Business Media LLC, 2020 Publication	1%
8	par.nsf.gov Internet Source	<1%

9	"Scale Space and Variational Methods in Computer Vision", Springer Science and Business Media LLC, 2017 Publication	<1 %
10	ebin.pub Internet Source	<1 %
11	Parikshit N. Mahalle, Namrata N. Wasatkar, Gitanjali R. Shinde. "Data-Centric Artificial Intelligence for Multidisciplinary Applications", CRC Press, 2024 Publication	<1 %
12	Submitted to Cranfield University Student Paper	<1 %
13	Aleksandr Kalinin, Akbar Anbar Jafari, Egils Avots, Cagri Ozcinar, Gholamreza Anbarjafari. "Generative AI-based style recommendation using fashion item detection and classification", Signal, Image and Video Processing, 2024 Publication	<1 %
14	researchportal.bath.ac.uk Internet Source	<1 %
15	web.science.mq.edu.au Internet Source	<1 %
16	Submitted to University College London Student Paper	<1 %
17	browse.arxiv.org Internet Source	<1 %
18	scholar.uwindsor.ca Internet Source	<1 %

19	Simon R. Arridge, Peter Maaß, Carola-Bibiane Schönlieb. "Deep Learning for Inverse Problems", Oberwolfach Reports, 2022 Publication	<1 %
20	indico.jlab.org Internet Source	<1 %
21	www.ukessays.com Internet Source	<1 %
22	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2019", Springer Science and Business Media LLC, 2019 Publication	<1 %
23	integraudio.com Internet Source	<1 %
24	mdpi-res.com Internet Source	<1 %
25	www.numberempire.com Internet Source	<1 %
26	Marcello Carioni, Subhadip Mukherjee, Hong Ye Tan, Junqi Tang. "Unsupervised approaches based on optimal transport and convex analysis for inverse problems in imaging", Walter de Gruyter GmbH, 2024 Publication	<1 %
27	Matthias Chung, Emma Hart, Julianne Chung, Bas Peters, Eldad Haber. "Paired Autoencoders for Likelihood-free Estimation in Inverse Problems", Machine Learning: Science and Technology, 2024 Publication	<1 %

28	Riccardo Barbano, Zeljko Kereta, Andreas Hauptmann, Simon R Arridge, Bangti Jin. "Unsupervised knowledge-transfer for learned image reconstruction", Inverse Problems, 2022 Publication	<1 %
29	"Energy Minimization Methods in Computer Vision and Pattern Recognition", Springer Science and Business Media LLC, 2011 Publication	<1 %
30	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2018", Springer Nature America, Inc, 2018 Publication	<1 %
31	B. Merialdo, J. Jiten, B. Huet. "Multi-Dimensional Dependency-Tree Hidden Markov Models", 2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings, 2006 Publication	<1 %
32	Bingsheng He, Xiaoming Yuan. "Convergence Analysis of Primal-Dual Algorithms for a Saddle-Point Problem: From Contraction Perspective", SIAM Journal on Imaging Sciences, 2012 Publication	<1 %
33	Shakir Ali, Mohammad Ashraf, Vincenzo De Filippis, Lahcen Oukhtite, Nadeem Ur Rehman. "Differential Identities in Rings and Algebras and their Applications", CRC Press, 2025 Publication	<1 %