Acceleration of 3D X-ray Microscopy (XRM) using Compressive Sensing and Deep Learning Techniques For Semiconductor Products

Tan Ying Hao

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

16/07/2024

Date

Tan Ying Hao

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

16/07/2024

Date

Assoc. Prof Nicholas Vun Chan Hua

Authorship Attribution Statement

This thesis contains material from 1 paper published in the following peer-reviewed papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Tan Ying Hao, Nicholas Vun, and Bu-Sung Lee. Sparse image measurement using deep compressed sensing to accelerate image acquisition in 3D XRM. Sixteenth International Conference on Quality Control by Artificial Vision. Vol. 12749. SPIE, 2023.

The contributions of the co-authors are as follows:

- Prof. Nicholas Vun provided the initial research direction, reviewed and edited the manuscript draft
- Prof. Bu-Sung Lee provided feedback for the research direction and manuscript draft.
- I proposed the algorithms, performed experiments and prepared the manuscript draft.

16/07/2024

.....

Date

.....

Tan Ying Hao

Acknowledgements

I wish to express my utmost gratitude to my supervisor, Prof. Nicholas Vun Chan Hua, for his invaluable guidance and unwavering support throughout my doctoral journey. His wealth of expertise and experience has not only shaped the trajectory of this research but has also been a constant source of inspiration. Most importantly, his steadfast encouragement and unwavering support during moments of uncertainty proved instrumental in keeping me motivated during challenging times. His provision of ample space and time and consistent follow-ups on my progress enabled me to effectively balance my work commitments and research pursuits, ensuring the timely achievement of milestones toward my research objectives.

I would also like to extend my deepest gratitude to my co-supervisor, Prof. Bu-Sung Lee, for his guidance and encouragement throughout these years. As a mentor from the onset of my undergraduate studies, he has consistently provided invaluable opportunities, from supporting my participation in high-performance computing competitions to facilitating my transition into graduate studies. His insightful advice has played a pivotal role in shaping both my academic and personal development. Furthermore, his constant feedback on my research progress during my doctoral journey has been pivotal in shaping the content and direction of this thesis.

I extend my sincere gratitude to the AMD Data Science and Artificial Intelligence Lab supervisor group: Prof. Nicholas Vun Chan Hua, Prof. Bu-Sung Lee, Prof. Deepu Rajan, and Dr. Sourav. Their collective guidance and support have been invaluable throughout my doctoral journey. The weekly meetings facilitated enriching discussions that not only provided my research with much-needed direction but also fostered a collaborative environment conducive to intellectual growth and exploration.

To my fellow doctoral colleagues from AMD, Mr. Sun Jianxiao, Mr. Chang Junqing, and Ms. Zhao Lin, I am eternally grateful to have met each of you. Your

encouragement, concern, and regular check-ins have been a constant source of strength and motivation throughout my doctoral journey. Moreover, your expertise and guidance in various aspects, including setting up experiments, debugging codes, and sharing academic resources, have made invaluable contributions toward the enrichment of my research.

To my colleagues from AMD, Mr. Sanyal Saptak, Mr. Eich Zachary, Mr. Swamy Sridhar and Ms. Alice Chua Qin Hui, thank you for your encouragement and support throughout these years. In particular, I wish to express my sincere appreciation to my manager, Mr. Sanyal Saptak, for his kind understanding and exceptional management skills. His ability to navigate timelines with such clarity and foresight has enabled us to strike a harmonious balance between our professional responsibilities and research endeavors, thereby facilitating a conducive environment for growth and productivity.

To Mr. Chan Li Long, I would like to extend my deepest gratitude for his invaluable contribution to my thesis. His generous sharing of the NeRF concept played a pivotal role in shaping the final part of my research. His insights have significantly enriched the depth and scope of my work, for which I am immensely thankful.

In addition, I wish to extend my deepest apologies and heartfelt gratitude to my friends, relatives, and family members. I must apologize for unintentionally neglecting many of you during periods when I struggled to balance between work and research commitments. This journey has been long and arduous, and I am acutely aware that it would not have been possible without constant support from each and every one of you. Your constant encouragement, thoughtful check-ins, and invitations to spend time together have demonstrated the true essence of camaraderie. To my family members, Tan Lye Hock, Chua Bee Yen, Tan Ying Zi, Ong Yi Hao, and Calista Ong, and my best friends, Mr. Wong Yijie, Mr. Lim Xuan Yin, and Ms. Charlene Tan, I owe an immeasurable debt of gratitude to each of you. Throughout the most challenging periods of this journey, you were the unwavering pillars of strength that sustained me. Your steadfast presence, whether it was offering a listening ear or respectfully giving me space to process my thoughts, has been an invaluable source of support. I am eternally grateful for your companionship and the comfort it has brought me during the toughest times.

Last but not least, I would like to acknowledge AMD (Advanced Micro Devices) and EDB (Economic Development Board) for their joint sponsorship of my Industrial Postgraduate Programme (IPP). Their generous support not only facilitated the pursuit of this research but also enriched my academic journey by providing invaluable resources and opportunities.

"A person who never made a mistake never tried anything new."

—Einstein, Albert

To my dear family

Abstract

3D X-ray Microscopy (XRM) is a key technique used in semiconductor product failure analysis due to its non-invasive nature and high-resolution capabilities. The process uses X-ray to scan the target region and generate 2-D images from different angles. It then combines these images to generate a 3-D cross-section picture of the target region, which is then visually inspected by failure analysis engineers on the possible cause of the failure. Despite the widespread adoption of 3D X-ray microscopy (XRM) as a valuable tool for failure analysis, it faces challenges such as extended scan times and substantial storage requirements. These issues stem from the necessity of acquiring a large number of measurements to achieve high-quality reconstructions. This thesis presents the development of novel techniques that are designed to make the 3D XRM measurement process to be more efficient in terms of time and storage requirements.

The first technique leverages compressive sensing to capture 2-D images, hence reducing the time and storage spaces required when compared to the existing process. A novel measurement process, Sparse Matrix Deep Compressed Sensing (SM-DCS), is developed that is able to generate 2-D image from sparse (15%) version of the scanned image with quality that is comparable to the full-resolution image captured using the same equipment.

The second technique, XRM-NeRF leverages Neural Radiance Field (NeRF) to reduce the number of 2-D projection images (16.67%) required to reconstruct an optimized 3-D model. It uses a deep learning process to obtain a set of fine-tuning parameters to mitigate measurement noise in the pose estimation of semiconductor samples. Results shows that it is able to achieve quality comparable to those using the existing 3D XRM process.

iXRM-NeRF is then proposed to accelerate the learning process used in XRM-NeRF. It leverages meta-learning technique to learn common knowledge shared

across scenes, reducing the number of steps required for optimizing an iXRM-NeRF model for a scene. Additionally, iXRM-NeRF model proposes a visualization technique that queries the model using orthogonal projection to generate the cross-sectional view of a 3D XRM scene, assisting failure analysis engineers to identify defects.

The work presented in this thesis hence introduced several innovative techniques that illustrate the promise of integrating compressive sensing and deep learning methodologies to expedite 3D XRM for semiconductor failure analysis. The integration of compressive sensing techniques, deep learning techniques, and specialized domain knowledge in this thesis highlights the potential to accelerate the product development lifecycle for semiconductor products.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathbf{R}^n$	n-dimensional Euclidean space
$A \odot B$	Element-wise multiplication between elements in matrix A and B
$A \cdot \vec{p}$	Inner product between matrix A and vector $\vec{p}$
$A \times B$	Cartesian product between set A and set B
$\|A\|$	Cardinality of set A

Acronyms

CS	Compressive Sensing
DCS	Deep Compressive Sensing
SM-DCS	Sparse Matrix Deep Compressive Sensing
NeRF	Neural Radiance Field
XRM-NeRF	XRM-Neural Radiance Field
iXRM-NeRF	Improved XRM-Neural Radiance Field
FOV	Field Of View

Chapter 1

Introduction

Integrated circuits serve as the fundamental building blocks for modern technology. The ubiquity of smart devices, laptops, and specialized AI processors has encouraged the semiconductor industry to explore methods for maximizing computational capabilities within increasingly compact die spaces. Semiconductor fabrication process node sizes have been steadily diminishing in alignment with this trajectory. Consequently, as process nodes decrease in size and more components can be squeezed into the same die area, the complexity of the integrated circuit increases. This results in the increasing importance of the failure analysis phase in the semiconductor product lifecycle.

Failure analysis plays a crucial role in the semiconductor product lifecycle, serving as a vital step in pinpointing systematic failures. Additionally, it acts as a valuable feedback mechanism for the design team, enabling the development of robust products that can withstand fabrication imperfections. Early stages of failure analysis often incorporate 3D X-ray Microscopy (XRM), a technique utilized by engineers to detect defects in semiconductor products. This process involves reconstructing the internal structure of the product to obtain a three-dimensional (3D) view, aiding in defect identification. The popularity of 3D XRM in failure analysis is due to its non-invasive nature and its high-resolution capabilities. These capabilities enable the precise detection of defects at sub-micron scales without damaging the product, thereby assisting engineers in identifying the underlying causes of failures.

X-ray Microscopy (XRM) works by directing X-rays from a source through a sample (E.g. Semiconductor product) at different angles and measuring the attenuated

X-ray using a detector on the opposite end of the source to form a 2-dimensional projection image slice. Since different materials have different X-ray attenuation coefficients, the 2-dimensional (2D) projection image slices can inform engineers about the different materials present in the sample. This process is illustrated in Figure 1.1. As the X-ray beams illustrated by the yellow dotted lines pass through the sample, a projection depending on the attenuation of the X-ray measured by the detector is generated. 3D XRM scan is completed by gathering multiple projections of the focus region at different angles through rotating the sample which is illustrated using the green arrow. By collecting multiple 2-dimensional X-ray projection image slices at sufficiently dense angles, the 3D internal structure of the sample can be reconstructed with high fidelity using appropriate software tools [2][3]. **As integrated circuits continue to become denser with more components, the occlusion effect becomes a significant challenge in failure analysis, where overlapping structures can obscure critical defects. Therefore, 3D reconstruction is necessary to fully understand the complex geometry of the circuit, allowing engineers to accurately identify the root cause of failures that might be hidden in a 2D projection view**.

Despite the advantages of 3D XRM, there remain several challenges that need to be addressed for its practical and efficient usage.

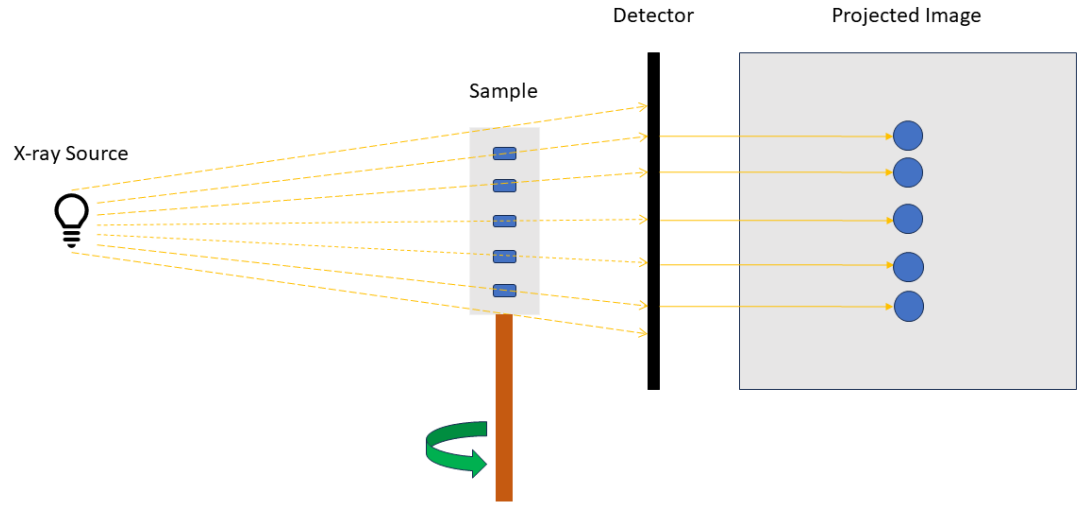

FIGURE 1.1: Overview of the 3D XRM scan process. The different colors (gray and blue) in the sample illustrate different materials.

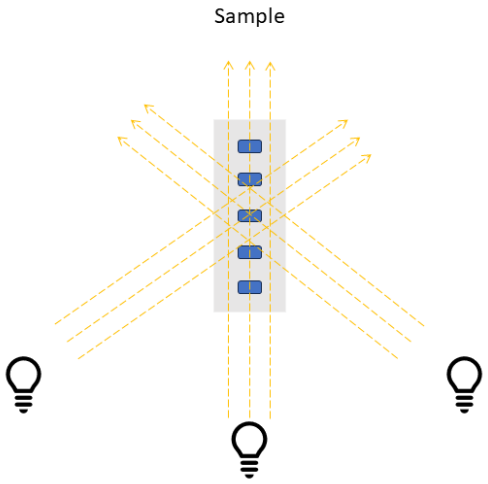

FIGURE 1.2: Illustration of the under-constrained problem of 3D XRM using a top down view of the measurement process.

Slow Scan Times

In 3D XRM, the task of 3D reconstruction from 2D projections is an under-constrained problem as illustrated in Figure 1.2. The light bulb represents the position of the X-ray source as the sample is rotated along the Y-axis. The yellow dotted lines trace the X-ray beams, and the intersection points of those beams represent points in space that can be reconstructed accurately. The reconstruction of the surrounding spaces that do not lie in the intersection paths of the X-ray beams is an under-constrained problem due to a lack of information. The limited range of angles results in incomplete information about the 3D internal structure. This means that there are many possible 3D structures that could generate the same set of 2D projections, leading to ambiguity in the reconstruction process. A practical but inefficient way of reducing such ambiguity is to image the sample at a sufficiently dense set of angles (i.e. higher resolution) to ensure that the generated 3D reconstruction is at an appropriate level of fidelity for its proposed task (E.g. Defect identification in semiconductor products). Selecting a dense set of angles increases the scan time as the sample has to be physically rotated to each selected angle and stabilized before the X-ray imaging can happen.

Large Storage Requirements

For high-resolution imaging tasks, the data volume of each 3D XRM projection scan escalates significantly, comprising at least $K \times K \times M$ data points, where

$K \times K$ represents the pixel count and M denotes the number of angles at which the 2D projections are captured. This substantial increase in data volume can lead to extensive storage demands.

1.1 Approaches

In this thesis, we tackle the challenges associated with 3D XRM for semiconductor products, with a particular focus on integrated circuits, by proposing three complementary approaches. These approaches were developed sequentially, each building upon the limitations of the previous one. To maintain a coherent perspective and logical progression of ideas, the proposed approaches are organized in chronological order, as outlined below.

1.1.1 Reducing Pixel Scan on 2D XRM Projection

There are multiple factors that can affect the scan times and storage requirements for 3D XRM. Our initial approach focuses on leveraging advances in compressive sensing techniques to minimize the number of pixel scans for each 2D XRM projection while maximizing the reconstruction quality.

Recent advances in compressive sensing techniques and deep learning enable the relaxation of requirements for signal recovery by introducing a neural network encoder to non-linearly project a low-dimensional vector to a higher-dimensional signal vector space $\vec{y} = G(\vec{\hat{x}}; \theta)$, where $G(.; \theta)$ is the neural network encoder and the constraints on $\vec{\hat{x}}$ is low dimensionality rather than sparsity [4][5].

Although the ability to minimize measurements through compressive sensing techniques closely aligns with our objective of reducing pixel scans in 2D XRM projection, it poses compatibility challenges with existing XRM scanners, which will be further discussed in Chapter 3. This incompatibility requires a significant hardware overhaul to existing 3D XRM scanners, such as adding a digital micromirror device to the detector [6], to employ existing compressive sensing techniques to XRM.

To avoid expensive hardware changes, we propose a novel approach based on the deep compressive sensing [5] algorithm to simulate sparse signal scans during the training process. This thesis hence proposes a novel two-phased measurement process, Sparse Matrix Deep Compressed Sensing (SM-DCS), to address the incompatibility issue of compressive sensing and existing 3D XRM scanners. The first phase of SM-DCS measures a sparse subset of the pixels using existing 3D XRM scanners, and the second phase of SM-DCS formulates a compressive sensing problem using the reduced measurements from the first phase. The experimental results demonstrate that we are able to achieve reconstruction quality parity with deep compressive sensing while maintaining compatibility with existing 3D XRM scanners. By measuring only 15% of the image pixels, SM-DCS demonstrates the capability to reconstruct a 2D XRM projection image with an average PSNR of 27 dB.

Our major contributions are as follows: (a) Modification of the deep compressive sensing algorithm to utilize existing XRM scanners to reduce pixel scan per projection. (b) Quantitative and qualitative evidence to demonstrate the efficacy of our modified algorithm.

1.1.2 Reducing Projection Scan for 3D XRM Reconstruction

Our second approach addresses the 3D XRM challenges to generate a model of the 3D XRM scene using only a sparse set of the 2D XRM projection images by leveraging the concept of a variant of Neural Radiance Field (NeRF) [7], known as mip-NeRF [1].

Although mip-NERF has demonstrated strong performance in reconstructing 3D scenes from 2D projection data, its algorithm was primarily crafted for RGB images captured using camera devices and presents a disconnect with the physics of XRM [8]. For our 3D XRM-based data, this thesis proposes the XRM-NeRF, an adaptation of the mip-NERF training process, which reformulates the volumetric rendering process of NeRF to cater to the physics of 3D XRM.

In addition to the problem discussed above, the position and rotation of 3D XRM samples, in 3D XRM, are controlled via a mechanical process and the measurements

of such data can be noisy. This noise poses a challenge for leveraging mip-NeRF to perform 3D reconstruction as it requires accurate information about the pose of the projection images. One of the possible manifestations of such noise is in the offset of the measured rotation axis from the actual rotation axis, known as the center shift. To mitigate the effects of noise in the center shift, the proposed XRM-NeRF includes the learning of a fine-tuning parameter to ensure that accurate pose information can be constructed by incorporating the learned fine-tuning parameter.

The experimental results demonstrate that we can achieve reconstruction quality with an average PSNR between 37dB and 50dB, and an average SSIM between 0.9803 and 0.9924 while utilizing only 16.67% of the initial number of projection images.

Our major contributions are as follows: (a) Developed XRM-NeRF to tailor the mip-NERF training process for the 3D XRM scenario by replacing alpha composition with attenuation calculation and incorporating a novel volumetric rendering process to account for the physics of X-ray attenuation. (b) Develop a novel method to automatically learn a fine-tuning parameter to account for measurement errors of the center shift due to mechanical processes.

1.1.3 Improving XRM-NeRF Reconstruction and Cross-Sectional View Generation

1.1.3.1 Meta-Learning

While the above approaches lead to the reduction of scan time for 3D XRM, computational time during the reconstruction process becomes more complicated and time-consuming. A crucial aspect of the proposed reconstruction process involves training the XRM-NeRF model to accurately learn the 3D structure of the scene. The training of XRM-NeRF is a computationally demanding task, requiring the training process to make $K \times K \times M \times D$ minimum number of queries on the XRM-NERF model for each epoch, where $K \times K$ represents the pixel count, M represents the number of imaged angles, and D represents the number of sampled points per cone.

To reduce the computational overhead of the reconstruction process, we proposed iXRM-NeRF that leverages on meta-learning to learn an initial set of weight parameters that reside closer to the optimal solution for multiple 3D XRM datasets. iXRM-NeRF incorporates a meta-learning phase that is conducted before the actual reconstruction process to learn a set of weight parameters that encode macroscopic knowledge, such as common 3D structures found in 3D XRM datasets. The weights learned during the meta-learning phase are then used to initialize the weights of the iXRM-NERF model to perform the reconstruction of an unknown scene. During the reconstruction phase, the iXRM-NERF training process is responsible for learning the microscopic details of the 3D scene and fine-tuning the weight parameters to generate projections that fit the set of projections captured by the 3D XRM scanner. Our experiments show that iXRM-NeRF trained with a meta-learning phase outperforms XRM-NeRF trained without a meta-learning phase.

1.1.3.2 Cross-Sectional View

A common procedure in semiconductor failure analysis using 3D XRM involves examining the cross-sectional view of the reconstructed 3D XRM to detect defects like short circuits or open circuits. This poses a challenge with iXRM-NERF, as generating such a cross-sectional view necessitates the capability to acquire 3D data using orthographic projection. The iXRM-NeRF model performs cone casting which inherently acquires 3D data through perspective projection. We propose to perform cylinder casting, as shown in Figure 1.3, to calculate the mean and variance of each sampled point which will be used as query inputs for the iXRM-NeRF model during inference to generate accurate cross-sectional views. The sampled points have variances independent of the distance along the cylinder, accurately modeling orthographic projection. This enables the iXRM-NeRF model to incorporate perspective projection during training time while generating orthographic projection during inference time, enhancing its applicability in semiconductor failure analysis. **Note that Figure 1.3 uses a simpler 3D construction digger scene instead of a 3D semiconductor scene. This choice is made for illustration purposes, as the audience is likely to be more familiar with the 3D construction digger model compared to a complex 3D semiconductor scene.**

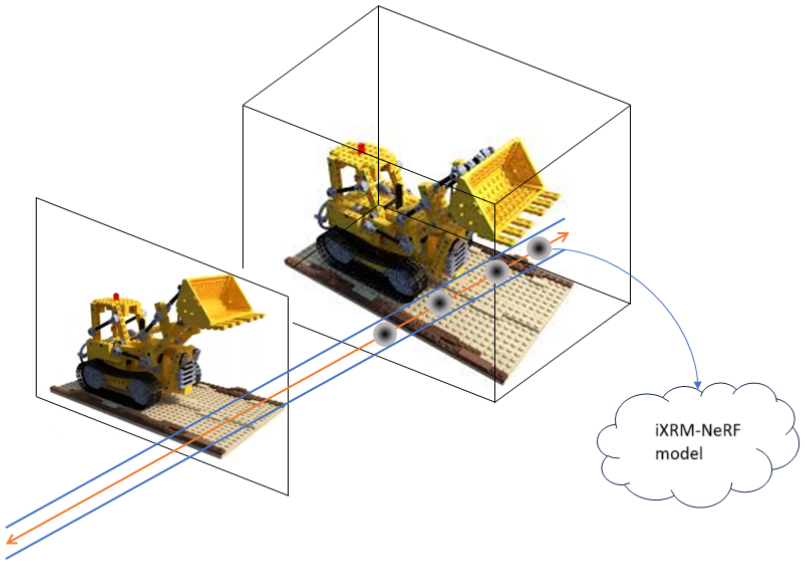

FIGURE 1.3: Overview of the iXRM-NeRF algorithm to cast cylinders instead of cones.

1.2 Major Contributions

In summary, the contributions in the main thesis can be stated as follows:

- Proposal of Sparse Matrix Deep Compressive Sensing (SM-DCS) algorithm that enables the application of deep compressive sensing to 2D XRM projection to reduce pixel scan per projection image.
- Proposal of XRM-NeRF that caters to 3D XRM dataset by predicting the attenuation coefficient rather than the color and density profile of a 3D point and automatically learning the fine-tuning parameter to account for measurement errors of the center shift due to systematic noise that can occur during the mechanical processes of 3D XRM scans.
- Proposal of iXRM-NeRF that learns initial weight parameters for XRM-NeRF model to improve the convergence of the optimization process and an inference time algorithm that enables queries based on orthographic projection on a trained iXRM-NERF model to generate cross-sectional views of a 3D scene.

1.3 Outline of the Thesis

This thesis is organized in the following order:

Chapter 1 introduces the background and scope of the challenges addressed by this thesis.

Chapter 2 provides a comprehensive literature review of the existing works which are related to the approaches described in section 1.1.

Chapter 3 describes the research methodologies, experimental results, and conclusion for our approach to reducing pixel scan on 2D XRM projection.

Chapter 4 describes the research methodologies, experimental results, and conclusion for our approach to reducing projection scan for 3D XRM reconstruction.

Chapter 5 describes the research methodologies, experimental results, and conclusion for our approach to improving and accelerating 3D XRM reconstruction using meta-learning and generation of cross-sectional views for industrial applications.

Chapter 6 concludes the research by summarizing key contributions and findings; This chapter also suggests possible future research directions that may further address the challenges posed by 3D XRM.

Chapter 2

Literature Review

This chapter provides a comprehensive review of existing research work relevant to the approaches discussed in this thesis. It covers a diverse range of topics, such as advances in compressive sensing, 3D reconstruction from sparse views, and meta-learning.

2.1 Compressive Sensing

Compressive sensing is a signal processing technique that exploits the sparsity of a signal set to achieve signal reconstruction by gathering fewer samples than required by the Nyquist-Shannon sampling theorem [9]. In compressive sensing, a known measurement matrix $\Omega \in \mathbf{R}^{M \times N}$ in conjunction with a measured vector $\vec{m} \in \mathbf{R}^{M}$ are used to reconstruct a signal vector, $\vec{y} \in \mathbf{R}^{S}$. In this context, the number of elements in the signal vector is strictly more than the number of elements in the measured vector $S >> M$, leading to the under-constrained problem of $\vec{m} = \Omega\vec{y}$. The theory of compressive sensing states that the signal vector $\vec{y}$ can be uniquely recovered from the measurement vector $\vec{m}$ if there exists a linear projection matrix $\phi \in \mathbf{R}^{S \times N}$ that can form an injective projection of sparse vectors into the signal of interest $\vec{y} = \phi\vec{x} \mid ||\vec{x}||_0 \leq K$. The existence of such a linear projection matrix facilitates the creation of a measurement matrix $\hat{\Omega} = \Omega\phi$, which transforms the reconstruction problem of the signal vector into the reconstruction problem of the sparse vector, $\vec{m} = \hat{\Omega}\vec{x}$. Due to the sparsity constraints of the sparse vector $\vec{x}$, the problem of recovering the sparse vector $\vec{x}$ from the measurement vector $\vec{m}$ is

not under-constrained, and the recovery of the signal vector can be completed by projecting the recovered sparse vector $\vec{x}$ through the linear projection matrix ϕ.

Research in compressive sensing is generally categorized as (i) the search for a sparse basis for a signal set and (ii) the optimization of recovery algorithms to recover the sparse vectors from measurement vectors.

2.1.1 Sparse Basis

Compressive sensing states that the minimum number of measurements to guarantee recovery quality is linearly proportional to the signal sparsity; However, most naturally occurring signal sets, such as images and sound, may not be sparse in their original basis. Nevertheless, it may still be possible to apply compressive sensing to these signals by leveraging on the observation that the entropies of these signals are usually much lower than random signals, and it may be possible to design a sparse basis that injectively projects a sparse auxiliary signal to the signal of interest. By subsequently recovering the sparse auxiliary signal and projecting it through the sparse basis, it is possible to perform signal reconstruction with fewer measurements.

The problem of constructing a sparse basis for a signal set can be formulated as the minimization of L_0 norm of the sparse vector under the constraints that ϕ should injectively project the sparse vectors to the set of signal vectors $\min_{\phi} ||\vec{x}||_0 \mid \forall \vec{y} \in Y, \vec{y} = \phi\vec{x}$.

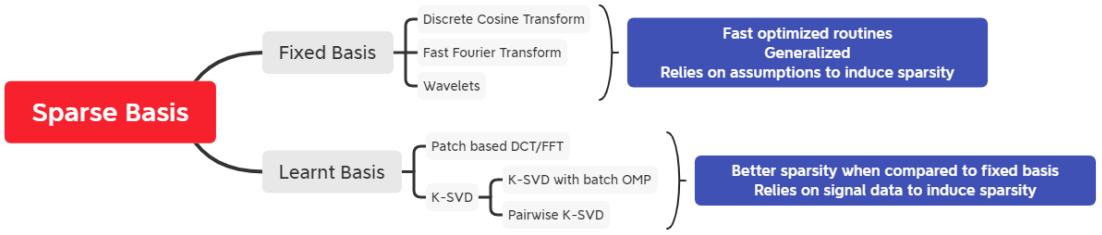

FIGURE 2.1: An overview of techniques used to construct sparse basis as well as their respective advantages and disadvantages

Initial research of sparse basis revolved around fixed transforms such as fast Fourier transform (FFT) or discrete cosine transform (DCT) of the signals. The advantages

of fixed transforms are ease of generalization and computational efficiency. The full-rank nature of FFT and DCT ensures that they guarantee the ability to represent all signals through these transforms, leading to ease of generalization.

Fixed transforms rely on the assumption that the signal of interest is dense so that they are sparse after the transformation due to the uncertainty principle [10], which may not be true. As such, there are various attempts to find a sparse basis, such as by grouping signals into local patches [11], enabling the construction of a sparse basis for locally dense signals.

More recent advancements in finding sparse bases have contributed to the emergence of other prominent techniques such as K-Singular Value Decomposition (K-SVD) and its various iterations [12–15].

K-SVD basically uses an iterative method to find a sparse basis ϕ. It reformulates the problem of finding a sparse basis that induces a minimum L_0 norm on the auxiliary signal vector to the problem of finding a sparse basis that approximates signal recovery for a given sparsity level constrain $\min_{\phi} ||Y - \phi X||_F^2 \mid \forall \vec{x} \in X, ||\vec{x}||_0 \leq K$. Each iteration in K-SVD is a two-step process that first finds a K-sparse matrix X that best approximates the signal set Y given an initial sparse basis ϕ, follows by minimizing the error $||Y - \phi X||_F^2$ through the adjustments of the coefficients in the sparse basis ϕ. The two-step process is repeated until the error term $||Y - \phi X||_F^2$ converges [12]. The advantage of K-SVD over fixed transforms is that K-SVD can find a customized basis ϕ that generally induces sparser auxiliary signals due to its ability to learn from patterns present in a signal set.

Figure 2.1 provides an overview of the techniques used to construct sparse bases discussed in this section.

2.1.2 Recovery Algorithm

This section discusses the various techniques that have been proposed to recover the signal set Y after compressed measurements are gathered. Recovery algorithms in compressive sensing are generally classified into two major classes, matching pursuit and basis pursuit. An overview of the classes of recovery algorithms is shown in Figure 2.2.

FIGURE 2.2: An overview of the classes of algorithms to recover signals from compressed measurements

Matching pursuit seeks to solve the problem of signal recovery by minimizing the signal recovery error term under L_0 norm constrain on the sparse vector, $\min_{\vec{x}} ||\vec{m} - \Omega\phi\vec{x}||_2^2 \mid ||\vec{x}||_0 \leq K$. Due to the non-convex nature of the L_0 norm constraint in matching pursuit, solving for such a problem is NP-hard. Most matching pursuit algorithms such as Orthogonal Matching Pursuit and its variants [13] [16] attempt to find an approximate solution to the problem through greedy methods instead.

Basis pursuit seeks to solve the problem of minimizing the L_1 norm of the sparse vector under signal recovery constraints, $\min_{\vec{x}} ||\vec{x}||_1 \mid \vec{m} = \Omega\phi\vec{x} \leq \epsilon$. While the problem of minimizing the L_1 norm may sound different from finding sparse coding given a measurement vector and a measurement matrix, there is a direct relationship between them and solving for minimum L_1 norm is equivalent to solving for minimum L_0 norm under specific constraints of the measurement matrix [17]. In this context, basis pursuit establishes a convex problem which can be solved in polynomial time using linear programming. To improve the practicability of basis pursuit for industry applications where noise may be present in the measurements, the constrained optimization problem of basis pursuit can be reformulated to include a noise tolerance threshold ϵ for signal recovery, $\min_{\vec{x}} ||\vec{x}||_1 \mid ||\vec{m} - \Omega\phi\vec{x}||_2^2 \leq \epsilon$, and this is commonly known as basis pursuit denoising [18]. The reformulated problem of basis pursuit denoising cannot be solved by linear programming and as such, various algorithms such as iterative shrinkage threshold algorithm [19] and learned iterative shrinkage algorithm [20], are developed to address the problem instead.

Iterative shrinkage thresholding algorithm (ISTA) solves the Lagrangian function of the constrained optimization problem, $\frac{||\vec{m}-\Omega\phi\vec{x}||_2^2}{2} + \lambda||\vec{x}||_1$, by iteratively performing projected gradient descent [19] on the sparse vector $\vec{x}_t$, starting from an initial estimate $\vec{x}_{t_0}$ [21], until convergence of the signal recovery error term. Subsequent literatures develop upon the basic algorithm of ISTA and employ various techniques, such as using Nesterov momentum [22] [23] and search space reduction [20], to increase the convergence speed of ISTA.

A notable literature that improves upon ISTA convergence is learnt iterative soft-thresholding algorithm (LISTA) [20]. LISTA relies on the intuition that the sparse auxiliary signal vectors to be recovered may reside in a strict subset of the K-sparse manifold; Therefore, by learning from the input distribution of the sparse auxiliary signal, the search space of the sparse auxiliary signals can be reduced and the upper bound on the speed of convergence can be improved. LISTA relies upon an autoencoder architecture to learn to predict the sparse auxiliary signal and establishes a relationship between ISTA and time-unfolded recurrent neural network.

2.1.3 Deep Learning Compressive Sensing

Building upon the foundations laid by K-SVD and LISTA, recent advances in the field of compressive sensing are progressing towards a focus on learning inherent patterns from the signal set. In this context, the superior capability of deep learning for pattern acquisition aligns with the trend of compressive sensing research; This convergence has spurred the emergence of literature exploring the adoption of deep learning in the field of compressive sensing [4, 5, 24–26].

The adoption of deep learning in the field of compressive sensing has also led to several interesting developments such as the relaxation of the sparsity constraints [4] and the replacement of the sparse basis ϕ with a non-linear projection function $G(\vec{x};\theta)$ [5]. These relaxations enable compressive sensing to be easily adapted to a new set of signals by retraining a deep neural network to learn the patterns in the signal set.

A novel literature in this direction, deep compressed sensing [5], proposes to adopt the generator component of a generative adversarial network (GAN) as the non-linear projection function $G(\vec{z}; \theta_g)$. The training process of deep compressed sensing iteratively involves a sequence of meta-learning steps to learn a set of optimal weight parameters for the generator and discriminator that is tailored specifically for compressed sensing recovery tasks.

Despite the promising results of deep compressed sensing, the method cannot be directly applied to 3D XRM scanners that we are concerned with due to its requirements for the measurement process to measure a linear combination of the pixels in a full-resolution projection image $\vec{y}$. Chapter 3 of this thesis presents our proposed method to overcome this limitation and facilitate the integration of deep compressed sensing into existing 3D XRM scanners.

2.2 3D Reconstruction from Sparse Views

The long-standing challenge of reconstructing 3D geometry from limited 2D perspectives has significant implications across various industries. One of the main topics in this thesis focuses on such a particular application: the reconstruction of 3D XRM data from sparse 2D XRM projection views.

The synthesizing of 3D geometry from sparse 2D perspectives is a major research topic, as it directly impacts the resolution, speed, and realism of the reconstructed 3D scene. Current research focuses on three major output representations, voxel representation [27–31], point representation [32], and occupancy field [1, 7, 33–36].

2.2.1 Voxel Representation

Voxel representation stands out as one of the most prevalent choices for 3D tasks, primarily because of its simplicity and widespread usability. Similar to the pixel representing the image color of a point on a 2D grid, the voxel represents the 3D scene color of a point on a 3D grid. Due to the similarity between pixel representation and voxel representation, early literature for 3D reconstruction focuses on adapting advances in the image synthesis domain into the domain of 3D reconstruction [30, 37]. The disadvantage of the voxel representation is the inefficient usage

of memory to store 3D representation since voxels allocate memory regardless of the occupancy status of each grid. Research in voxel representations struggles with high-resolution reconstruction and recent works have generated 3D reconstructions for resolutions only up to 128^3 [38, 39]. To address the challenges of memory inefficiency using voxel representations, hierarchical methods [40, 41] have been proposed for 3D reconstruction tasks. Despite the innovative solutions offered by such methods, they remain constrained by the limitations of the underlying 3D grid resolution.

2.2.2 Point Representation

Point representation resolves the memory inefficiency of voxel representation by representing points on the surface of a 3D object with a dense list of vertex, and storage is not allocated for unoccupied regions. Fan et al. [42] proposed developing a generator model utilizing deep convolutional neural networks to recover 3D point cloud sets given an input image and a viewpoint. They highlighted several other key advantages of point clouds for 3D reconstruction, such as rotation, translation, and scale invariance. Despite the advantages of point cloud representation, these representations require complex post-processing steps [43, 44] to generate 3D meshes, a common downstream task in 3D reconstruction.

2.2.3 Occupancy Field

Occupancy field represents a 3D scene by encoding the geometry of the scene as a continuous function that maps coordinates in a 3D scene to an output, representing occupancy [45–47], and/or color [1, 7, 35, 36]. In contrast with the discrete nature of voxel and point representation, a major advantage of occupancy field is the potential for representing a 3D scene with a limited memory footprint [47].

Park et al. [45] describes an approach for learning to map coordinates to signed distances (DeepSDF), $S : \mathbf{R}^3 \rightarrow \mathbf{R}$, where the sign represents whether the point is inside (negative) or outside (positive) the surface of an object. They propose to train a single auto-decoder feed-forward neural network to generate the signed distance for different 3D shapes by concatenating a latent vector z that serves as a sort of shape embedding to the coordinate input vector. The training process

involves regressing the output of the neural network with the signed distance randomly sampled from volumetric data, and the latent vector z is randomly sampled from a normal distribution. The inference involves estimating the latent vector z using Maximum-a-Posterior (MAP) followed by forward propagation to obtain the signed distance.

Genova et al. [47] describes an approach to learning a mapping from coordinate space to occupancy probability, $O : \mathbf{R}^3 \rightarrow [0, 1]$. They propose to leverage existing methods such as 3D-R2N2 [27] and PSGN [42] to sample volumetric data from images and train their occupancy network, a feed-forward neural network with 5 ResNet blocks, and conditional batch normalization. To reduce the computational overhead of 3D reconstruction, the inference process of the occupancy network follows a hierarchical approach that subdivides a bounded volume at an initial resolution and discards regions with occupancy below a pre-determined threshold value; This process of subdivision and discarding of regions is repeated until the desired resolution is achieved.

2.2.4 Neural Radiance Field

Despite having addressed the memory representation problems that plague high-resolution 3D scenes, the technique discussed in section 2.2.3 does not address the challenge of how data can be sampled from sparse views to train an occupancy field model.

Mildenhall et al. [7] proposed the Neural Radiance Field (NeRF) model, shown in Figure 2.3, that demonstrates an end-to-end approach to reconstructing a 3D NeRF model using sparse projection views. The NeRF model is a feed-forward neural network that predicts a color vector and occupancy value given a corresponding coordinate vector and viewing direction, $N : \mathbf{R}^3 \times \mathbf{R}^2 \rightarrow \mathbf{R}^3 \times \mathbf{R}$. The training process involves regressing the prediction of the projection views in the training set with its corresponding ground truth images. The prediction of projection views using NeRF model is a multi-step process that involves: (1) Casting a virtual ray for each pixel in the projection view. (2) Sampling points along each virtual ray. (3) Predicting the corresponding color vector and occupancy value for each sampled point. (4) Performing alpha composition to obtain the final pixel color for the points along each virtual ray.

While the proposed framework is capable of training a NeRF model given sparse projection views, the reconstruction is often blurry; Mildenhall et al. hypothesized in the same literature that the NeRF model is learning the low-frequency details of the 3D scene and proposed to encode the input coordinate to the NeRF model with positional encodings to encourage the learning of high-frequency details [48]. The proposed positional encoding function maps the input coordinates to a representation that promotes the feed-forward neural network to learn higher-frequency functions:

$$\gamma(\vec{p}) = [sin(2^0\pi\vec{p}), cos(2^0\pi\vec{p}), sin(2^1\pi\vec{p}), cos(2^1\pi\vec{p}), ..., sin(2^L\pi\vec{p}), cos(2^L\pi\vec{p})]$$

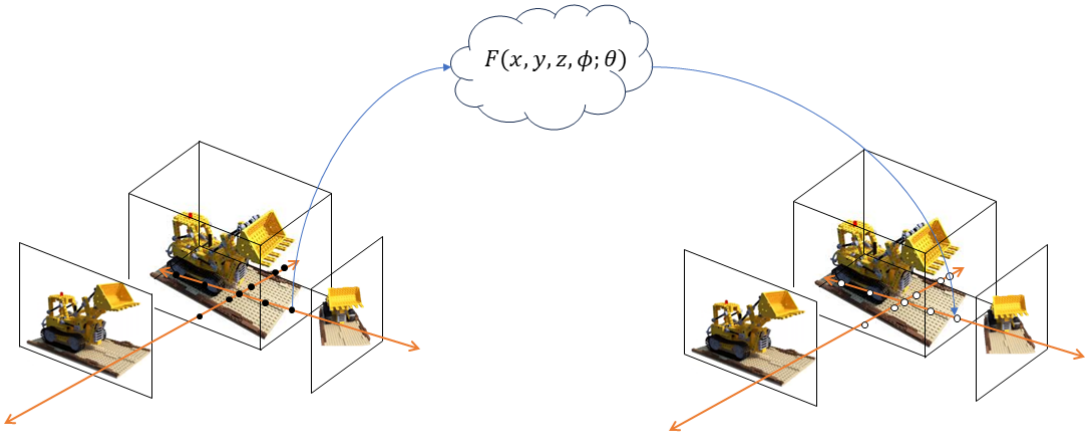

FIGURE 2.3: Overview of the NERF model.

In addition to proposing the positional encoding of input coordinates, Mildenhall et al. also proposed to leverage hierarchical sampling to reduce the computational burden of evaluating the whole bounded volume. Their hierarchical sampling approach involves an initial coarse sampling phase, which uniformly samples points along a virtual ray, succeeded by a refined sampling phase that employs weighted sampling for points along the same virtual ray. The weights for the weighted sampling process are determined by the contribution of each point in the coarse sample to the final pixel intensity of the generated image.

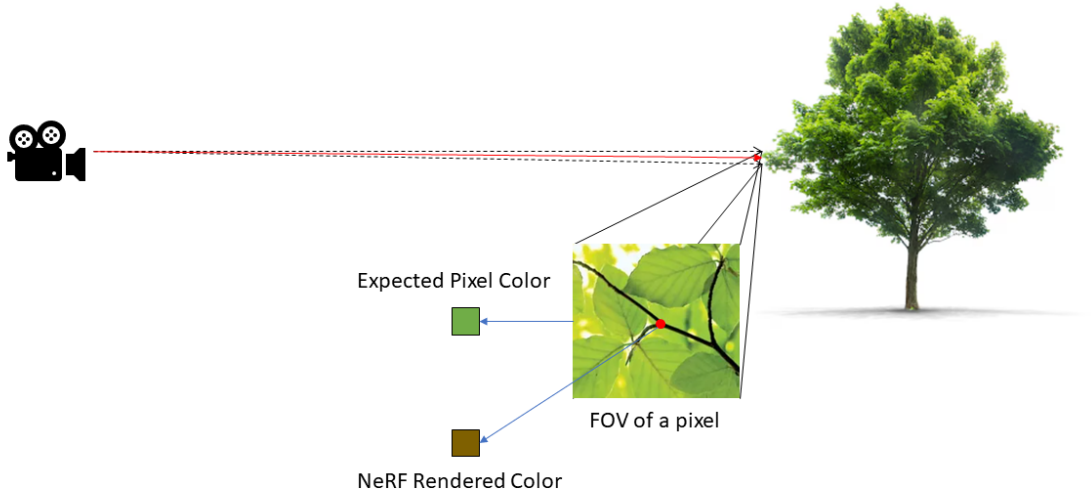

FIGURE 2.4: Illustration of the incompatibility between NeRF framework and perspective projection.

Perspective projection is a camera model that models real-world phenomena where distant objects appear smaller than nearer objects [49]; From the perspective of the imaging device, the field of view of a pixel expands along with the increase of distance from the camera. Figure 2.4 visually demonstrates the incompatibility between the NeRF framework and perspective projection. The two black dashed arrows represent the increasing field of view (FOV) of a pixel as distance increases. The red arrow represents the virtual ray cast by NeRF model and the red point in the FOV image for the pixel represents the infinitesimal point sampled by the NeRF procedure. In the illustration, the expected color of the pixel is green, as the leaves predominantly occupy the pixel's field of view. However, the use of infinitesimal point sampling by the NeRF framework causes rendering artifacts, resulting in the rendered color being brown if the sampled point falls on the tree branch. To address the issue of rendering artifacts inherent in the NeRF framework when rendering at multiple scales, Barron et al. [1] proposed mip-NERF, an improved version of the original NeRF framework. The mip-NeRF framework casts virtual cones to model the field of view changes due to perspective projection. In addition, they replaced the sampling of infinitesimal points with sampled Gaussian parameters (mean, variance) along a cone segment. The mip-NeRF process is illustrated in Figure 2.5. The two black dashed arrows represent the increasing field of view (FOV) of a pixel as distance increases. The red box represents the field of view of the pixel at a shorter distance, while the black box represents the field of view of

the pixel at a further distance. The Gaussian parameters are encoded as input to the mip-NeRF model via integrated positional encoding:

$$\gamma(\vec{\mu}, \vec{\sigma}) = [sin(2^0\pi\vec{\mu})e^{-\frac{4^0}{2}\vec{\sigma}^2}, cos(2^0\pi\vec{\mu})e^{-\frac{4^0}{2}\vec{\sigma}^2}, ..., sin(2^L\pi\vec{\mu})e^{-\frac{4^L}{2}\vec{\sigma}^2}, cos(2^L\pi\vec{\mu})e^{-\frac{4^L}{2}\vec{\sigma}^2}]$$

The integrated positional encoding intuitively reduces the magnitude of the high-frequency components with increasing distance between the camera and sampled Gaussian. An interesting advantage of integrated positional encoding is that it eliminates the need to perform hyper-parameter tuning to find the optimal L value for the encoding function. In contrast with NeRF formulation where L serves as a hard limit on the maximum frequency, in mip-NeRF, the proximity of the sampled Gaussian from the camera automatically modulates the magnitude of high-frequency components; Consequently, the L hyper-parameter can be set to a arbitrarily large value without compromising model performance.

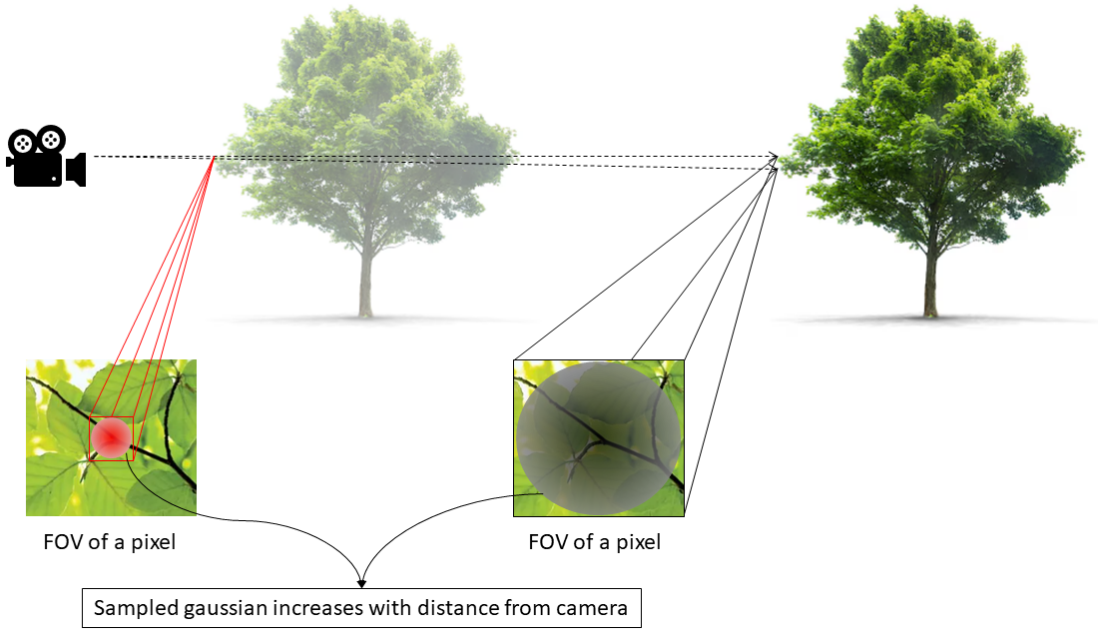

FIGURE 2.5: Illustration of the mip-NeRF framework and perspective projection.

Another major shortcoming of the NeRF framework is the requirement to optimize a separate neural network model for each distinct 3D scene. The optimization of a NeRF model requires a minimum of $K \times K \times M \times D$ evaluation of the model per

epoch. Matthew et al. [50] observed that the function space of a NeRF model is often far larger than the set of NeRF models representing the 3D signal of interest; Given that the optimization of a NeRF model is an iterative refinement process, the speed of convergence can be improved by initializing the NeRF model to a weight space that is closer to the weight spaces for the class of 3D signals that we are interested in reconstructing. They proposed to employ meta-learning methods [51] such as MAML [52] or Reptile [53] to search for an initial set of weights by training a NeRF model on the specific class of 3D signals targeted for reconstruction.

Model-agnostic meta-learning (MAML) [52] and Reptile [53] both aim to approximate a solution to the challenge of discovering an optimal set of initial weights for a distribution of tasks. In addition, both literatures propose algorithms that explicitly encourage fast adaptation of weights when training for a specific task within the same distribution. They proposed to set up an outer training loop to search for an optimal set of weights θ_0^* such that the expected final training loss over the task distribution T, after m inner loop optimization step, is minimized:

$$\theta_0^* = \min_{\theta_0} E_{t\sim T}[L(\theta_m(\theta_0, t))][50] \tag{2.1}$$

The inner loop optimization step $\theta_m(\theta_0, T)$ involves m steps of gradient descent [54] or similar solvers, such as Adam [55], that optimize the weights for the specific task T. The difference between MAML and Reptile for solving equation 2.1 is in the update rule for the outer loop.

MAML [52] proposes to directly backpropagate through the outer loop loss function $L(\theta_m(\theta_0, t))$ and update θ_0 using a gradient descent algorithm, as shown in eq. 2.2.

$$\theta_0 = \theta_0 - \alpha \frac{\partial L(\theta_m(\theta_0, t))}{\partial \theta_0} \tag{2.2}$$

As a result of backpropagating through the inner loop, which contains first-order derivatives, the update rule requires the calculation of second-order derivatives. While the calculation of second derivatives enables the optimization process to account for the curvature of the loss function resulting in faster convergence, it is memory intensive due to the requirement to store the backpropagation results for all m steps of the inner loop.

Reptile [53] proposes an update rule that interpolates between the initial weight $\hat{\theta_0}$ estimate and the final weight θ_m after m inner loop optimization, as shown in eq. 2.3.

$$\theta_0 = \theta_0 + \alpha(\theta_m(\theta_0, t) - \theta_0) \tag{2.3}$$

The advantage of Reptile over MAML is the invariance of its memory requirements with regards to the inner loop optimization step size m. This enables the Reptile algorithm to approximate a good initial set of weights for task distributions which requires many optimization steps, such as NeRF.

The experiment results of Matthew et al. [50] demonstrated that the set of weights learned during the meta-learning phase forms a very strong prior that guides the optimization process to achieve the best view synthesis PSNR metric over the class of 3D signals it was trained on; Furthermore, these experimental findings showcase the capacity of NeRF models when initialized with meta-learned weights to form a strong prior, will enable the recovery of 3D geometry from single input views, a task which the conventional NeRF formulation fails due to the under-constrained nature of such scenarios.

This chapter presents a comprehensive literature review of various topics such as compressive sensing, output representations for 3D geometric data, and the variants of NeRF for synthesizing 3D scenes from sparse views. The literature review of compressive sensing establishes the foundation for our work in Chapter 3, where we propose an extension of deep compressed sensing to perform compressive sensing on existing 3D XRM scanners. The literature review of the output representation for 3D geometric data and NeRF establishes the foundation for our work in Chapter 4. The literature review of the different variants of NeRF inspired the development of iXRM-NeRF in Chapter 5.

Chapter 3

Reducing Pixel Scan on 2D XRM Projections

This chapter[1] presents a Sparse Matrix Deep Compressive Sensing (SM-DCS) technique that minimize pixel scanning for XRM projections using existing 3D XRM scanners.

3.1 Limitation of Compressive Sensing

As mentioned in Chapter 2, Deep Compressive Sensing (DCS) [5] based techniques utilizing a combination of deep learning and compressive sensing have shown to be capable of training an end-to-end system by learning the generator model and measurement matrix [56, 57], hence enabling the application of compressive sensing techniques to many more different domains. In our case, the main challenge of applying Deep Compressive Sensing to 3D XRM is its incompatibility with existing scanners, that otherwise will require expensive hardware modifications.

Existing 3D XRM scanners operate using charged-coupled device (CCD) [58] technology, where each detector is responsible for gathering data for a single pixel in the resulting projection image. The measurement process for 3D XRM scanners

[1]This work in this chapter has been published by Tan, Ying Hao, Nicholas Vun, and Bu-Sung Lee. "Sparse image measurement using deep compressed sensing to accelerate image acquisition in 3D XRM." Sixteenth International Conference on Quality Control by Artificial Vision. Vol. 12749. SPIE, 2023.

can be expressed using the formula $\mathbf{I}\vec{p} = \vec{m}$, where $\vec{p}$ represents the pixel vector and $\vec{m}$ represents the measurement vector.

The measurement process of Deep Compressive Sensing is expressed using the formula $\Omega\vec{p} = \vec{m}$, where Ω represents the measurement matrix, $\vec{p}$ represents the pixel vector, and $\vec{m}$ represents the measurement vector. The unconstrained nature of the measurement matrix necessitates the gathering a linear combination of every pixel in the projection image for each element in the measurement vector. Figure 3.1 provides a visual illustration to demonstrate the incompatibility between compressive sensing and 3D XRM hardware scanners. The diagram on the left shows the measurement process of compressive sensing. Each detector detects a combination of multiple pixels. The shade of the digital micromirror represents the different attenuation coefficients to project a linear combination of the X-ray at different pixel locations into the detector. The diagram on the right shows the measurement process using an existing 3D XRM hardware scanner, each pixel value in the resulting projection is individually measured by a single detector. The incompatibility between the measurement processes presents a major challenge to adopting conventional DCS for existing 3D XRM scanners since each detector in the scanners can only measure the value of a single pixel.

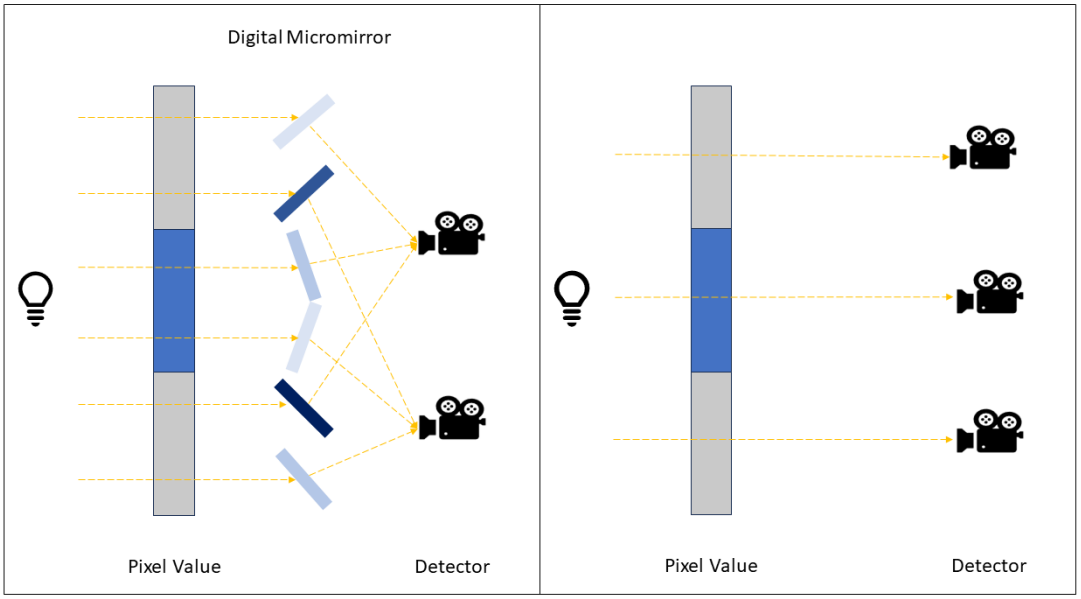

FIGURE 3.1: Illustration of the incompatibility between compressive sensing and 3D XRM hardware scanners.

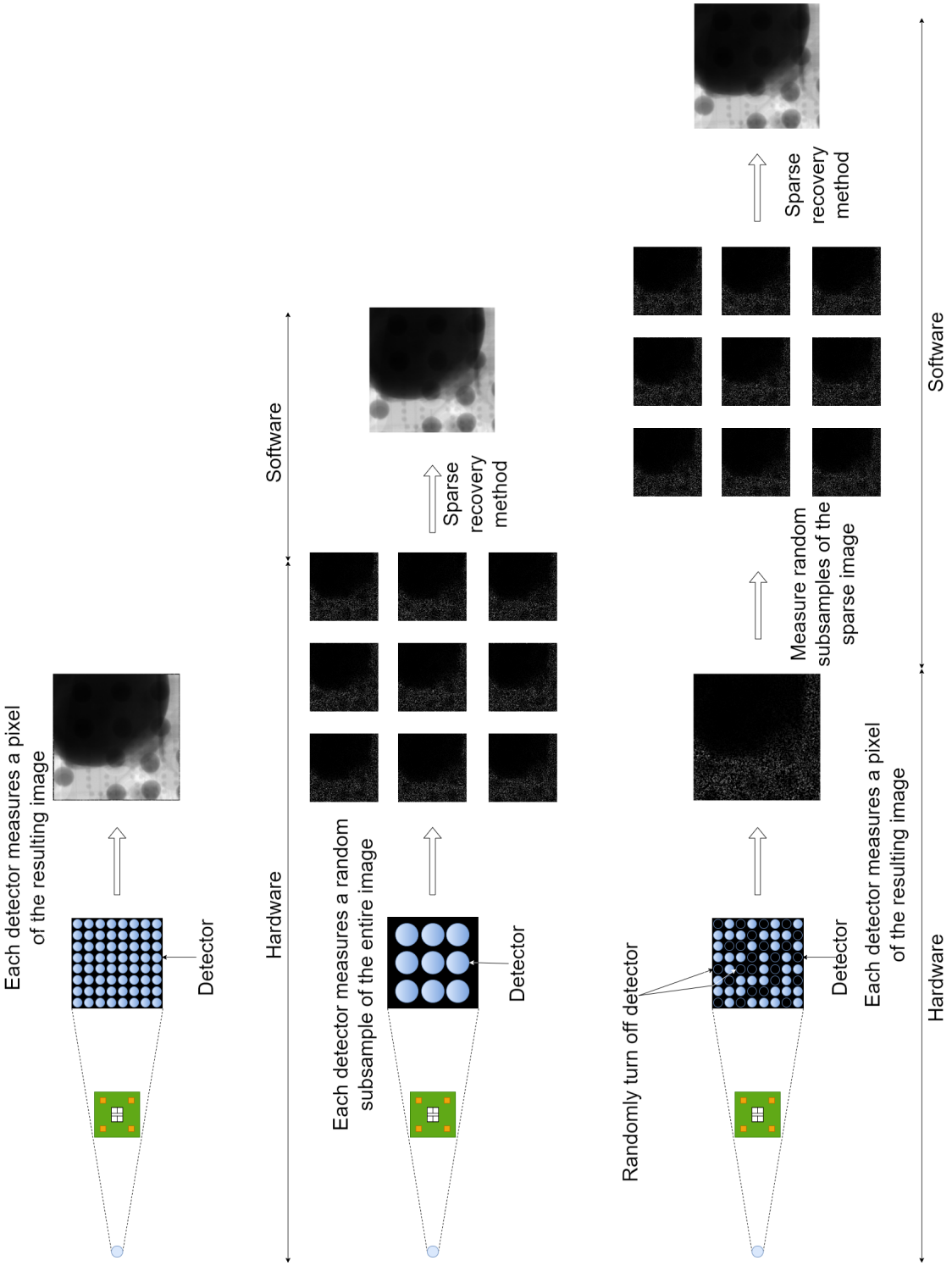

FIGURE 3.2: Comparison of the measurement process used for the various methods for image acquisition. Conventional image measurement process (Top), DCS measurement process (Middle), and SM-DCS measurement process (Bottom).

3.2 Sparse Matrix Deep Compressive Sensing

Our proposed Sparse Matrix Deep Compressive Sensing (SM-DCS) addresses the challenge of adopting Deep Compressive Sensing for existing 3D XRM scanners.

SM-DCS utilizes a two-phase approach for the measurement process. In the initial phase, a sparse subset of pixels in the resulting projection image is measured, expressed as $(\Lambda \odot \mathbf{I}) \cdot \vec{p} = \vec{m}$, where Λ is a sparse binary mask matrix indicating activated detectors in a 3D XRM scanner. This phase is used to significantly reduce the number of pixels to be scanned while ensuring compatibility with existing scanners. The subsequent phase involves measuring a linear combination of the results gathered in the first phase, expressed as $\Omega\vec{m} = \Omega(\Lambda \odot \mathbf{I}) \cdot \vec{p} = \hat{\vec{m}}$. This second phase facilitates the application of Deep Compressive Sensing for reconstructing $\vec{p}$ by formulating the first phase measurements as a compressive sensing problem $\hat{\Omega}\vec{p} = \hat{\vec{m}} \mid \hat{\Omega} = \Omega(\Lambda \odot \mathbf{I})$. Figure 3.2 provides a visual illustration that compares the differences between the measurement process using existing XRM scanners, using DCS, and our proposed SM-DCS.

The reconstruction process consists of two stages. The first stage involves a supervised training process where a generator model, defined by a neural network model, is trained to reduce the Euclidean distance between the predicted measurement vector and the actual measurement vector. The second stage involves a recovery process that performs a search in the latent space to find a latent vector $\vec{z}$ that minimizes the Euclidean distance between the predicted measurement vector and the actual measurement vector given the trained generator model. The signal vector $\vec{y}$ is reconstructed by passing the latent vector found in the second stage through the trained generator model. Due to the reformulation of the measurement matrix $\hat{\Omega}$ proposed in SM-DCS, which is a constrained version of the original measurement matrix Ω, naively applying a DCS generator $G(\vec{z}; \theta)$ trained using the unconstrained measurement matrix results in sub-optimal reconstruction. Therefore, during both the training and recovery processes, SM-DCS encodes the sparsity constraint $\Lambda \odot \mathbf{I}$ in the calculation of the measurement vector, $\vec{m} = \Omega(\Lambda \odot \mathbf{I})Y$, to ensure that the generator model learns to reconstruct the projection data in the constrained setting of sparse measurements. Algorithm 1 in Table 3.1 illustrates

Algorithm 1: SM-DCS recovery algorithm [5]

Input: Generator network $G(z)$, measurement matrix Ω and measured vector $\vec{m}$, number of latent optimization steps K, latent step size α

Output: Reconstructed image $G(\vec{z}_K)$

Function `RECOVER`$(G(z),\ \Omega,\ \vec{m},\ K,\ \alpha)$**:**

- Sample $\vec{z_0} \sim \mathcal{N}(0, 1)$
- **foreach** $i \in \{1, 2 \cdots K\}$ **do**
  - Set $L_M = ||\vec{m} - \Omega G(\vec{z}_{i-1})||_2^2$
  - Update $\vec{z_i} = \vec{z}_{i-1} - \alpha \frac{\partial L_M}{\partial \vec{z}_{i-1}}$
- **end**
- **return** $G(\vec{z}_K)$

Algorithm 2: SM-DCS training algorithm [59]

Input: Dataset of images I, measurement matrix Ω, Sparse binary mask Λ, number of latent optimization steps K, latent step size α, parameter step size β

Output: Trained generator network G_{θ_T}

Function `TRAINING`$(\Omega,\ K,\ \alpha,\ \beta)$**:**

- Initialize G_{θ_0}
- **foreach** $t \in \{1, 2 \cdots T\}$ **do**
  - Sample images $Y \subseteq I$
  - Set $\hat{\Omega} = \Omega(\Lambda \odot \mathbf{I})$
  - Set $M = \hat{\Omega} Y$
  - Set $L_M = \mathbb{E}_{\vec{m}_i \in M} ||\vec{m}_i - \hat{\Omega} RECOVER(G_{\theta_{t-1}}, \hat{\Omega}, \vec{m}_i, K, \alpha)||_2^2$
  - Set $L_N = \mathbb{E}_{\vec{m}_i, \vec{m}_j \in U} \left(||\hat{\Omega}(\vec{m}_i - \vec{m}_j)||_2 - ||\vec{m}_i - \vec{m}_j||_2\right)^2$
  - Set $L = L_N + L_M$
  - Update $\theta_t = \theta_{t-1} - \beta \frac{\partial L}{\partial \theta_{t-1}}$
- **end**
- **return** G_{θ_T}

TABLE 3.1: Recovery and training algorithms utilized by SM-DCS to perform reconstruction of 2D XRM projections

the process of recovering the optimal latent vector which is then used to reconstruct the projection data while Algorithm 2 in Table 3.1 illustrates the process of training the generator model for SM-DCS.

3.3 Dataset Preparation and Experiments Setups

3.3.1 Dataset

The experiments in this chapter compare the quality of the recovered image using DCS and SM-DCS methods and show quantitative and qualitative evidence that SM-DCS retains the recovery quality of DCS methods while enabling compressive sensing methods to be applied to existing hardware detectors. Experiments were performed on both the CelebA dataset [60] and a manually curated set of 3D XRM projection image datasets. The objective of performing the experiments on the CelebA dataset is to show the performance of SM-DCS on a larger scale, well-recognized dataset, while the objective of performing experiments on the 3D XRM dataset is to show that similar results translate well to the domain of semiconductor 3D XRM imaging.

FIGURE 3.3: A subset of preprocessed CelebA images.

The CelebA dataset consists of 202,599 images of celebrity faces, with each image originally having a resolution of 178 × 218 pixels. **For the purpose of the experiments, the images were subjected to a series of preprocessing steps. The first step of preprocessing involved randomly selecting a cropping factor from** 80% **to** 100% **of the image content and cropping the**

images accordingly to introduce variation into the dataset. Following this, the cropped images were resized to a uniform resolution of 128×128 **pixels. Finally, the images were converted to grayscale to simplify the input data and to focus on structural features.** Figure 3.3 shows a subset of preprocessed examples of the CelebA dataset.

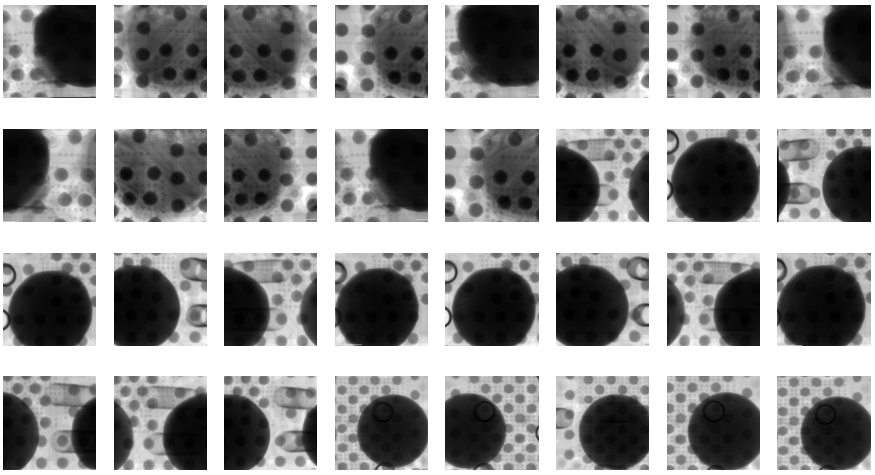

FIGURE 3.4: A subset of preprocessed 2D XRM projection images.

The dataset used in the SM-DCS experiments consists of 149 2D projections from 3D XRM measurements of integrated circuits (ICs), a subset of semiconductor products, with each projection having a resolution of 1013×1013 **pixels. These 3D XRM measurements were manually curated by conducting the measurements on a total of 5 IC samples, including 2 defective and 3 non-defective samples. The curation process involved manually running the 3D XRM scans to ensure the dataset accurately represents both defective and non-defective ICs. Since all measurements were conducted using the same 3D XRM machine, we acknowledge the potential for bias, particularly the possibility that the machine learning model may have overfitted to the characteristics of this specific machine. Although we recognize the limitations of this approach, it was performed to explore the feasibility of using SM-DCS to reduce pixel scans and gain insight into its potential for more efficient scans in future**

experiments. For the purposes of the experiments, the images were preprocessed using a sequence of steps similar to celebA images. The first step of preprocessing involved randomly selecting a cropping factor from 80% **to** 100% **of the image content and cropping the images accordingly to introduce variation into the dataset. Following this, the cropped images were resized to a uniform resolution of** 256×256 **pixels and converted to grayscale.** Figure 3.4 shows a subset of preprocessed examples of the 3D XRM dataset.

The output resolution for both the CelebA and 3D XRM dataset images was selected to balance GPU memory limitations while maintaining as much resolution as possible for the experiments.

After the preprocessing of each dataset as described above, a random selection of 70% of the images as the training set while the remaining images form the test set. The generator model was trained using the images from the training set, following the procedure outlined in algorithm 2. Before the start of the training process, the sparsity mask Λ was randomly generated by sampling from a Bernoulli distribution, such that only 15% of the pixels were activated. The PSNR results between the original and recovered image in the test dataset are reported in section 3.4. The experiments were evaluated five times by training the generators on resampled training and test sets to reduce the variance in the reported results.

3.3.2 Generator Architecture

The experiments utilized a generator model based on the state-of-the-art BigGAN-style architecture [61], which combines linear, residual, and spatial attention layers. The generator model used during training and evaluation remains the same, but the architecture varies depending on the dataset. **Specifically, the generator architecture changes in terms of the number of residual blocks and the number of channels in each layer to accommodate the differences in output size between the datasets.**

Figure 3.5 illustrates the generator architecture used to reconstruct CelebA images. This generator architecture features a linear layer that projects the latent vector into 2048 feature maps, each with a resolution of 4×4, followed by 5 residual blocks

that progressively project and upsample the input feature maps to 128 feature maps with a resolution of 64×64 each. The reconstructed image with a resolution of 128×128 pixels is eventually generated by passing the feature maps through a spatial attention module and a residual block layer.

Figure 3.6 illustrates the generator architecture used to reconstruct 3D XRM images. This generator architecture also utilizes a linear layer that projects the latent vector into 4096 feature maps, each with a resolution of 4×4, followed by 6 residual blocks that progressively project and upsample the input feature maps to 128 feature maps with a resolution of 128×128 each. The reconstructed image with a resolution 256×256 pixels is finally generated by passing the feature maps through a spatial attention module and a residual block layer.

The number of residual blocks used in the generator architecture was determined by evaluating various randomly selected configurations. Specifically, the base set of upsampling residual blocks was fixed according to the output spatial requirements for each dataset. For example, in the CelebA dataset, the output spatial requirement is 128×128 pixels, which requires 5 residual upsampling blocks. Due to limited resource constraints, specifically GPU memory, we could only add one layer of spatial attention. We tested various configurations by randomly inserting the spatial attention layer in different parts of the generator architecture, such as between residual blocks or before the output layer. Additional non-upsampling residual blocks were randomly inserted between the upsampling blocks, added after the input layer, before the output layer, or in any combination of these positions. After testing these different configurations, the best architecture—based on validation scores—was selected as the final model architecture for each dataset.

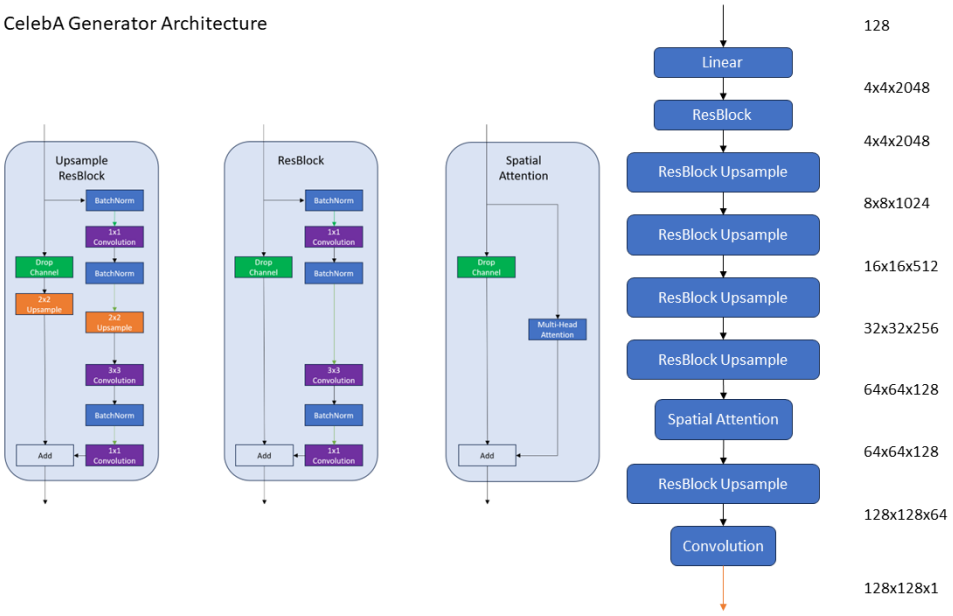

FIGURE 3.5: Generator model architecture for reconstructing CelebA images. The green arrows indicate ReLU activation, the black arrows indicate identity, and the orange arrow indicates tanh activation.

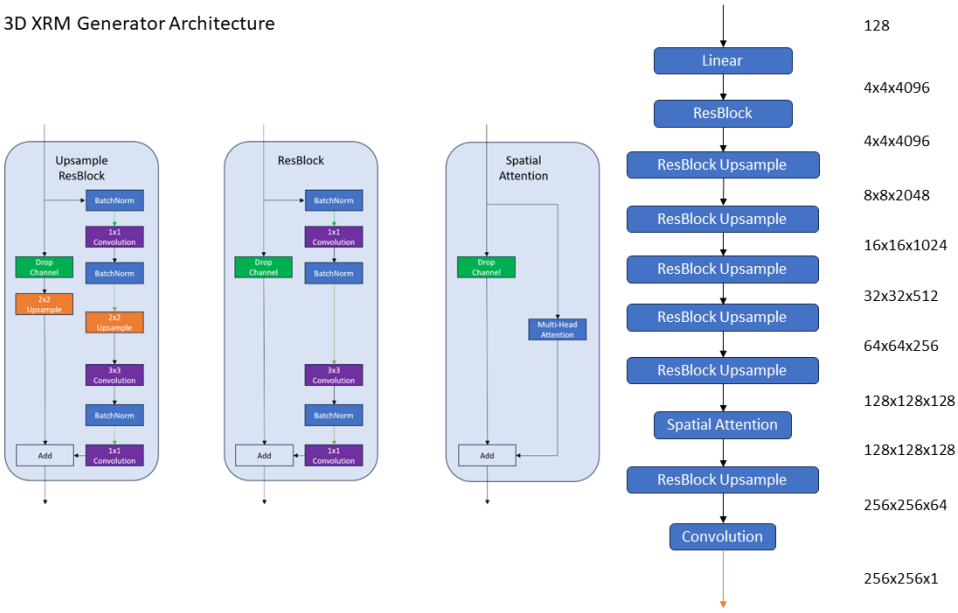

FIGURE 3.6: Generator model architecture for reconstructing 3D XRM images. The green arrows indicate ReLU activation, the black arrows indicate identity, and the orange arrow indicates tanh activation.

3.4 Results

3.4.1 Qualitative Evaluation

Figure 3.7 and 3.8 shows a qualitative comparison based on the visual comparison between DCS and SM-DCS methods to reconstruct the CelebA and 2D XRM projection images. Despite only measuring 15% of the image pixels for the CelebA dataset, as shown in the second column of the figures, SM-DCS can recover similar image quality compared to DCS. The same experiments were conducted on the 3D XRM dataset and similar image recovery quality can be observed when comparing DCS and SM-DCS.

3.4.2 Quantitative Evaluation

Figure 3.9 and Figure 3.10 show the box plots of the PSNR distribution of the recovered images using DCS and SM-DCS methods across the five separate experiment runs for the CelebA dataset and 3D XRM dataset, respectively. For both CelebA and 3D XRM datasets, the box plots show statistically insignificant differences between images recovered using DCS and SM-DCS methods. To further validate that there are insignificant differences in the PSNR distribution between the experimental runs for images recovered using DCS and SM-DCS, silhouette scores were calculated using the Bhattacharyya distance as the metric, the results are shown in the bar graph in Figure 3.11, and Figure 3.12. For all experiment runs, the figures show that the silhouette scores were very close to zero, indicating insignificant differences between the PSNR distribution of the recovered images between DCS and SM-DCS.

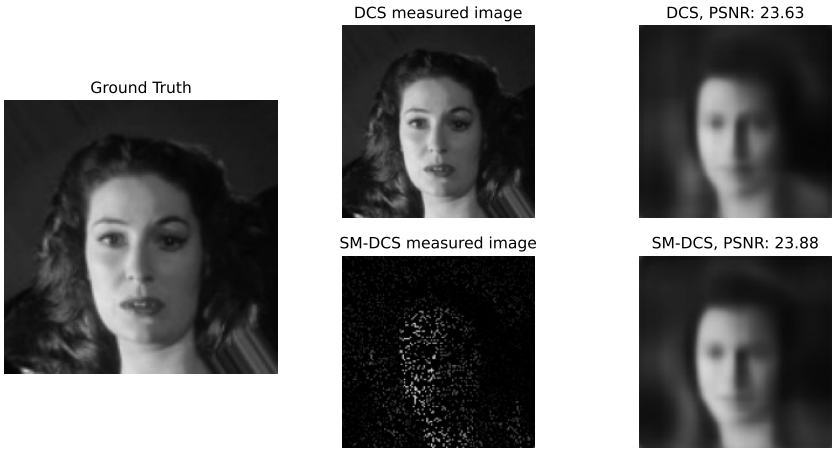

FIGURE 3.7: Qualitative comparison between recovered image quality using DCS and SM-DCS using CelebA dataset

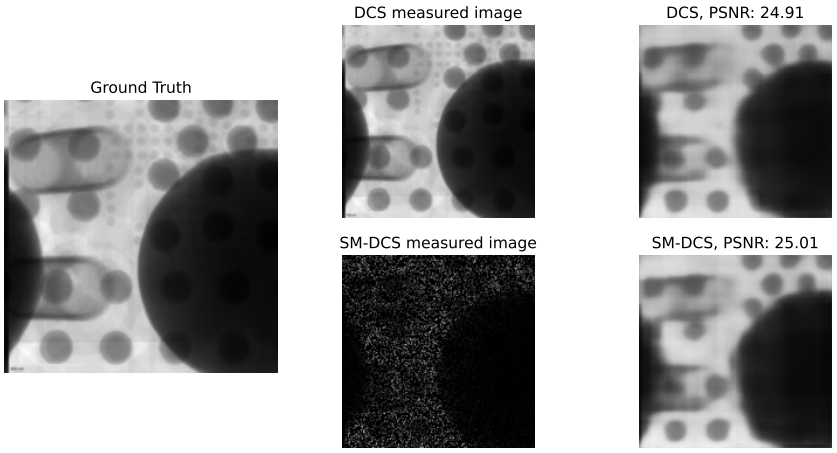

FIGURE 3.8: Qualitative comparison between recovered image quality using DCS and SM-DCS using 3D XRM dataset

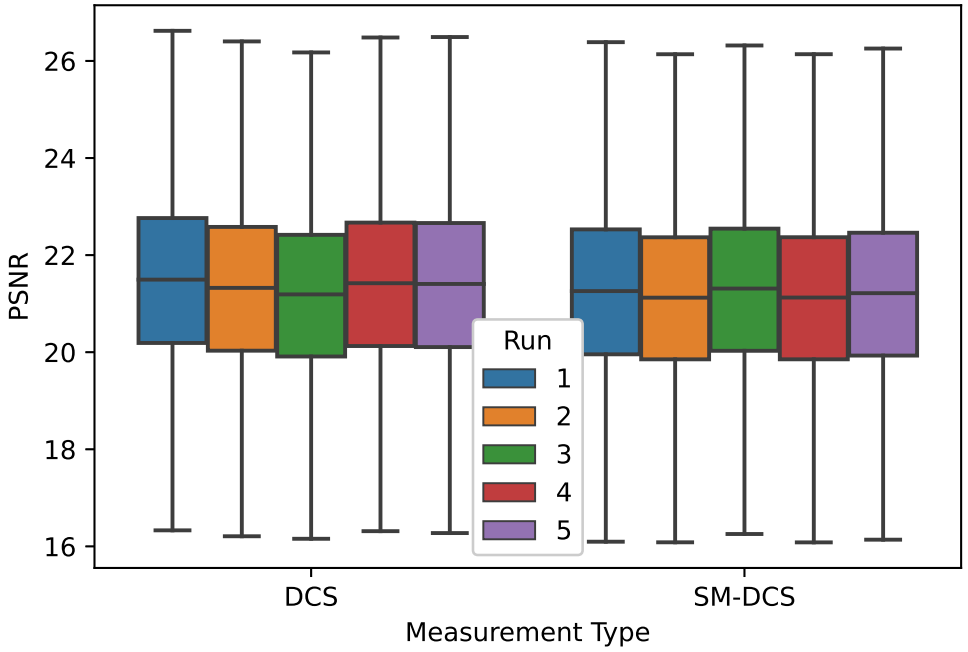

FIGURE 3.9: Comparison box plot of the different PSNR for the different experimental runs for DCS and SM-DCS reconstruction (CelebA)

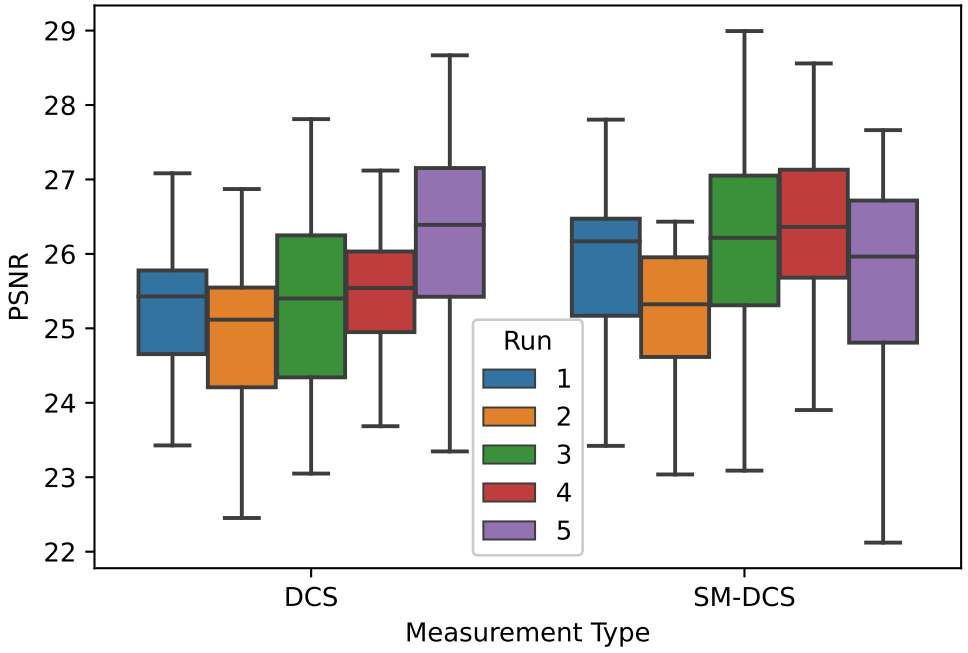

FIGURE 3.10: Comparison box plot of the different PSNR for the different experimental runs for DCS and SM-DCS reconstruction (3D XRM)

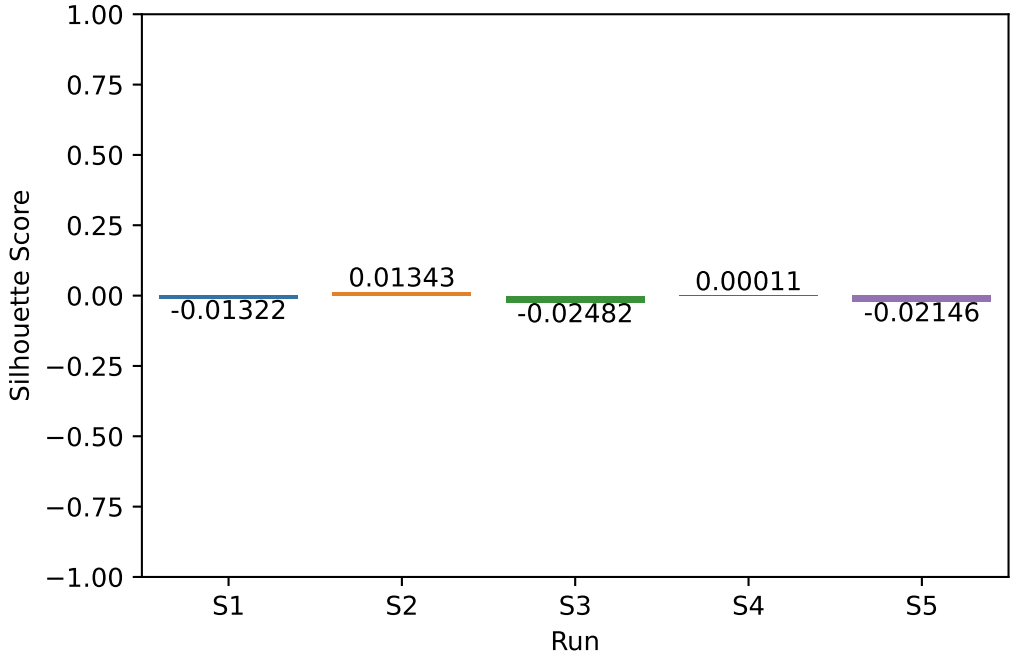

FIGURE 3.11: Bar graph of the silhouette score for the Bhattacharya distance between each SM-DCS run when compared with all other DCS runs (CelebA)

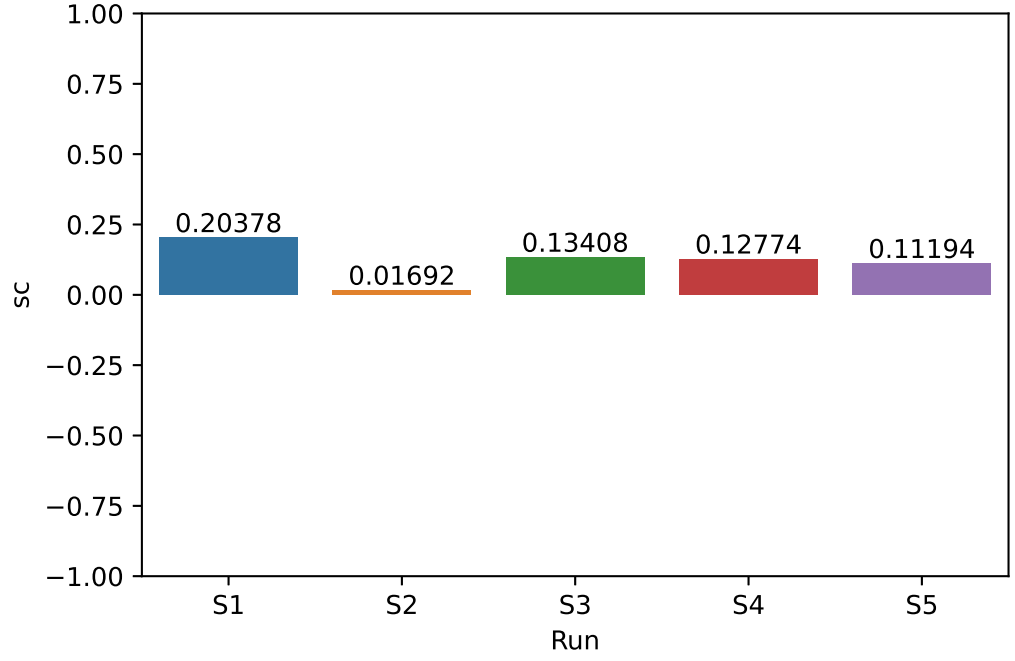

FIGURE 3.12: Bar graph of the silhouette score for the Bhattacharya distance between each SM-DCS run when compared with all other DCS runs (3D XRM)

3.5 Summary

This chapter presents our proposed two-phase measurement approach, SM-DCS, to bridge the incompatibility between DCS and the 3D XRM scanners used in the semiconductor failure analysis process. The first phase of SM-DCS focuses on reducing the number of pixel scans per XRM projection image, while the subsequent phase reformulates the obtained measurements into a compressive sensing problem. Quantitative and qualitative evidences are presented to show that SM-DCS is able to reconstruct projection images from sparse measurements without degradation to the quality of the reconstructed images when compared with the DCS approach. With the average PSNR approaching 27dB for reconstructed projection images that are based on only 15% of the full resolution, SM-DCS demonstrates the potential in expediting the 3D XRM measurement process that is used in the failure analysis in semiconductor products.

However, the study acknowledges the potential for bias, particularly the possibility that the machine learning model may have overfitted to the specific characteristics of the machine used. While recognizing these limitations, the experiments were performed to explore the feasibility of using SM-DCS to reduce pixel scans and to gain insight into its potential for improving scan efficiency in future experiments.

The next chapter presents an alternative approach, XRM-NeRF, to accelerate the 3D XRM measurement process by reducing the number of projection images to reconstruct a 3D XRM scene.

Chapter 4

XRM-NeRF

This chapter presents the XRM-NeRF approach to reconstruct 3D XRM scenes from sparse 2D XRM projection images.

4.1 Limitation of NeRF

Neural Radiance Field (NeRF) is a significant development milestone in the field of computer vision, enabling the reconstruction of 3D scenes from a small number of projection images captured at different viewpoints. The popularity of NeRF stems from its potential to represent an unbounded or high-resolution 3D scene with a bounded memory model, and the incorporation of the physics of differentiable volumetric rendering into its training process. The usage of a differentiable volumetric rendering in its training process constraints a NeRF model to learn the 3D geometry of a scene and enables downstream tasks such as depth map, mesh, point cloud, or novel view synthesis. Subsequent NeRF literature work, mip-NeRF [1], proposed to incorporate the perspective camera model into the training process, significantly enhancing the performance of NeRF models in multi-resolution scenarios. The natural solution offered by mip-NeRF in learning the 3D structure of a scene from limited viewpoints holds significant promise in streamlining the process of 3D XRM reconstruction by minimizing the number of projection scans.

Despite the advantage that mip-NeRF offers to address the scan time complexity of 3D XRM scanners, its volumetric differentiable rendering process was designed

primarily to learn 3D structures from images captured using RGB imaging devices. As a result, the rendering process used by mip-NeRF presents a disconnect with the physics of 3D XRM scanners. 3D XRM projects X-rays from a source through a sample and the different materials in the sample attenuate the X-ray energy differently resulting in an XRM projection image captured by a detector placed at the opposite end of the source. The intensity of X-rays passing through a material can be accurately modeled by the equation $I(t) = I_0 e^{-\int_0^t (u/p)(x)dx}$ [8], where u/p represents the mass attenuation coefficient and t represents the distance the X-ray traveled through the material. This implies that X-ray intensity can only decrease when it passes through a material since mass attenuation coefficient and distance are strictly positive; Therefore, in the 3D XRM scenario, each point is only characterized by a single scalar attenuation coefficient value. The additional assumptions of the degree of freedom in the mip-NeRF volumetric differentiable rendering process can result in artifacts for predicted 3D scenes that cannot occur in XRM scenes. Figure 4.1 and Figure 4.2 illustrate two distinct configurations of density predictions that can be generated by mip-NeRF to accurately reconstruct the projection image. However, as discussed earlier regarding the physics of XRM, only the latter figure accurately represents the 3D XRM scene. In Figure 4.1, NeRF can reconstruct the projection image shown in the X-Y view by predicting the presence of white cylindrical geometrical structures, shown as dark dense structures in the density prediction Y-Z view. This presents a disconnect with the physics of 3D XRM, where X-ray passing through the dense material should result in lower intensity or darker pixel values of the cylindrical structure shown in the X-Y projection image. Figure 4.2 illustrates a density prediction Y-Z view that is compatible with the physics of XRM scanners. In the illustration, a pipe structure can be seen in the density map implying that the cylindrical structure is hollow rather than being a dense material.

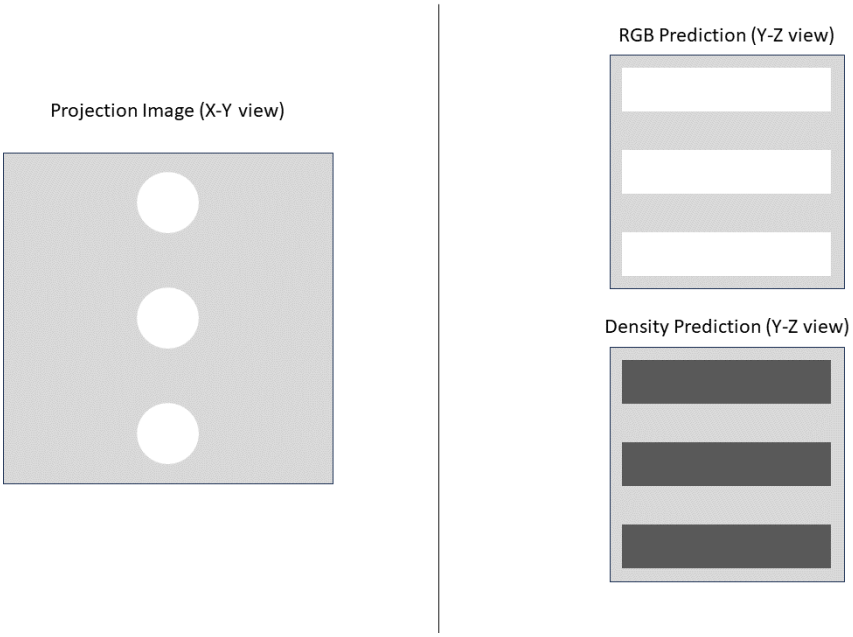

FIGURE 4.1: Illustration of a possible artifact that can be caused by the extra degree of freedom that NeRF predicts. Darker shaded regions in the density prediction view indicate the presence of denser material.

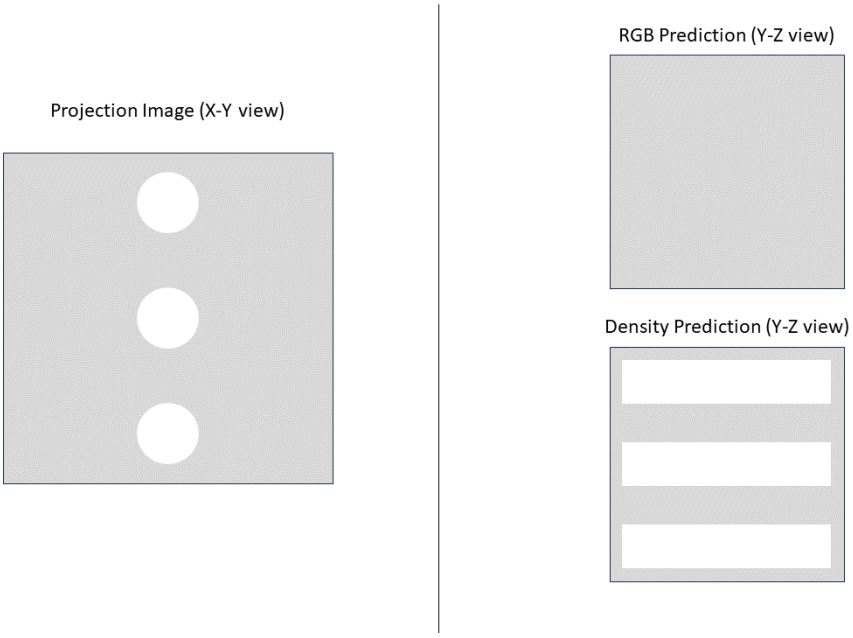

FIGURE 4.2: Illustration of a ground truth density prediction compatible with the physics of XRM scanners. Darker shaded regions in the density prediction view indicate the presence of denser material.

Another challenge for leveraging mip-NeRF to reduce projection scans for 3D XRM scanners is its requirement for complete and accurate information of the pose to project virtual rays to be used for volumetric rendering. During a 3D XRM scan, the sample is adjusted and rotated manually using a motor fitted with sensors to determine its position where the pose for each projection image can be derived. However, the motor sensors merely offer approximations of the position, necessitating engineers to fine-tune the center shift manually to achieve an accurate reconstruction of a 3D XRM scene; center shift refers to the difference in pixels that the axis of rotation is offset from the center column of the projection image [62] as shown in Figure 4.3. Direct usage of the pose information from the 3D XRM scan can therefore offer inferior reconstruction results due to inaccurate center shift.

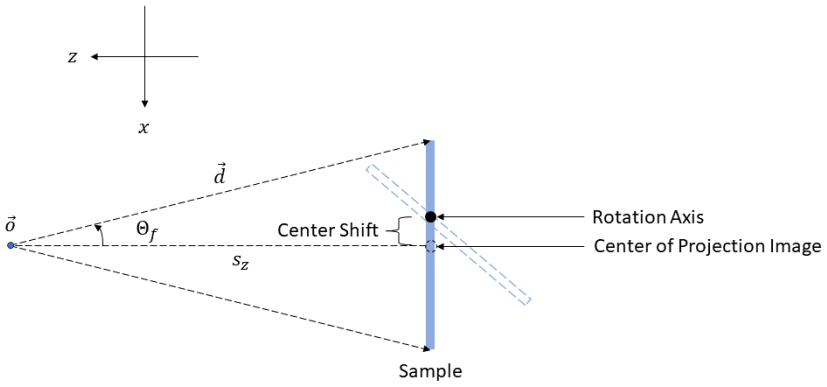

FIGURE 4.3: Illustration of center shift in 3D XRM measurements.

The following section hence presents a novel approach, XRM-NeRF, to address the two challenges of leveraging mip-NeRF for 3D XRM.

4.2 XRM-NeRF

4.2.1 Volumetric Rendering Process

XRM-NeRF characterizes each sampled point within a 3D XRM scene with a single scalar mass attenuation coefficient value. An XRM-NeRF model is designed to predict the mass attenuation coefficient based on both the 3D coordinate position and viewing direction $G(\vec{p}, \vec{d}; \theta) : \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}$.

The volumetric rendering process of XRM-NeRF is specifically tailored to accommodate the measurement of X-ray intensity as it traverses through a material. The X-ray intensity after passing through a material can be modeled by equation 4.1 [8], where I_0 represents the initial X-ray intensity before passing through the material. Consequently, for complex objects with $|\dot{T}|$ number of materials stacked together, the measured X-ray intensity can be expressed as equation 4.2. By replacing the mass attenuation coefficient, u/p, in equation 4.2 with the prediction of the XRM-NeRF model the volumetric rendering equation 4.3 for XRM-NeRF can be derived, where $\vec{o}$ represents the origin of the virtual ray. Based on equation 4.3, a separate model has to be trained for each material in the complex object scene; This is often impractical as the only information from a 3D XRM scan are the projection images. To simplify the equation for practical usage, it is possible to assume that the XRM-NeRF model can be trained to learn the material agnostic mass attenuation coefficient of a 3D XRM scene solely based on the position and viewing direction, resulting in the volumetric rendering equation 4.4.

$$I(\dot{t}) = I_0 e^{-\int_0^{\dot{t}} (u/p)(t)dt} \tag{4.1}$$

$$I(\dot{T}) = I_0 \prod_{(t_0,t_1)\in\dot{T}} e^{-\int_{t_0}^{t_1} (u/p)(t)dt} = I_0 e^{-\sum_{(t_0,t_1)\in\dot{T}} \int_{t_0}^{t_1} (u/p)(t)dt} \tag{4.2}$$

$$V(I_0, \vec{o}) = I_0 e^{-\sum_{(t_0,t_1,\theta_i)\in\dot{T}} \int_{t_0}^{t_1} G(\vec{o}+t\vec{d},\vec{d};\theta_i)dt} \tag{4.3}$$

$$V(I_0, \vec{o}, T) = I_0 e^{-\mathbf{E}_{t\in(0,T)}[G(\vec{o}+t\vec{d},\vec{d};\theta)]} \tag{4.4}$$

While the calculation of the exponent term in equation 4.4 is intractable, the exponent term can be approximated by discretizing the continuous function by sampling discrete points along the virtual ray, simplifying the exponent term calculation in the volumetric rendering equation to 4.5. The set of discrete samples, X, is an ordered set of distances along a virtual ray from the origin between 0 and T.

$$\mathbf{E}_{t\in(0,T)}[G(\vec{o}+t\vec{d},\vec{d};\theta)] \approx \sum_{t_i\in X} G(\vec{o}+\frac{t_i+t_{i-1}}{2}\vec{d},\vec{d};\theta)(t_i-t_{i-1}) \tag{4.5}$$

Following the derivation presented, equation 4.6 summarizes the volumetric rendering process of XRM-NeRF.

$$V(I_0,\vec{o},T) = I_0 e^{-\sum\limits_{t_i\in X} G(\vec{o}+\frac{t_i+t_{i-1}}{2}\vec{d},\vec{d};\theta)(t_i-t_{i-1})} \tag{4.6}$$

4.2.2 Fine-tuning Center Shift

Our proposed XRM-NeRF also extends the mip-NeRF model by including a learnable parameter, θ_x, to learn an offset to fine-tune the center shift during the training process. To understand how θ_x can be learned using gradient optimization methods, it is important to consider the camera model used by XRM-NeRF and subsequently derive the gradient of the mass attenuation prediction model $G(\vec{o}+t\vec{d},\vec{d};\theta)$ with respect to θ_x.

The camera model for a 3D XRM scene can be described by a translation along the x and z axis, followed by a rotation along the y axis [62], as shown in Figure 4.4. The perspective camera model is considered by including Θ_f which represents the camera's field of view.

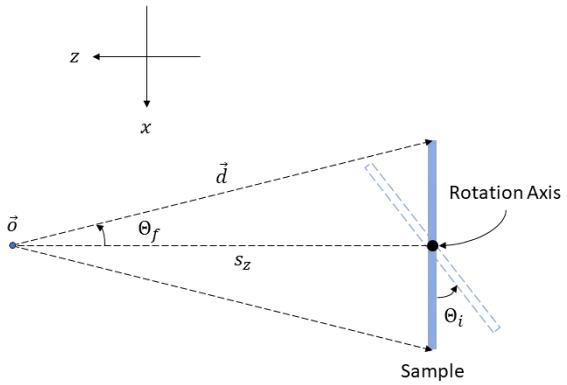

FIGURE 4.4: Illustration of the camera model utilized by XRM-NeRF to calculate the origin vector and direction vector for the XRM-NeRF model.

Equation 4.7 shows the definition of the translation matrix T, the rotation matrix R_i, and the projection matrix H [63] used for XRM-NeRF. The translation matrix T represents the origin vector offset from the rotation axis [64] along x and z axes. The rotation matrix R_i is a unique matrix for each projection image that represents a rotation along the y axis to capture the sample at multiple angles; The subscript i represents the rotation matrix for the i^{th} projection image. The H matrix represents a projection matrix to project from a homogeneous coordinate system to a Cartesian coordinate system. Based on the defined matrices, the origin vector in world coordinates for a 3D XRM scene can be derived by a translation operation followed by a rotation operation on the origin vector in eye coordinates [64], $\vec{o} = HR_iT\begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}^T$. Similarly, the direction vector in world coordinates can be derived from the direction vector in eye coordinates, $\vec{d} = HR_iT\begin{pmatrix}x\tan\Theta_f & y\tan\Theta_f & -1 & 0\end{pmatrix}^T$, where $x \in [-1, 1]$ and $y \in [-1, 1]$ represents the frame coordinates and Θ_f represents the field of view of the camera.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, R_i = \begin{pmatrix} \cos\Theta_i & 0 & -\sin\Theta_i & 0 \\ 0 & 1 & 0 & 0 \\ \sin\Theta_i & 0 & \cos\Theta_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T = \begin{pmatrix} 1 & 0 & 0 & s_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.7}$$

XRM-NeRF also adds a fine-tuning matrix C, as defined in Equation 4.8 that contains a learnable parameter θ_x to perform a fine-tuning translation operation before the translation by T. Based on the above, the XRM-NeRF origin vector and direction vector in world coordinates are derived as shown in Equation 4.9 and 4.10 respectively.

$$C = \begin{pmatrix} 1 & 0 & 0 & \theta_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.8}$$

$$\vec{o} = HR_iTC \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} (s_x + \theta_x)\cos\Theta_i - s_z \sin\Theta_i \\ 0 \\ (s_x + \theta_x)\sin\Theta_i + s_z \cos\Theta_i \end{pmatrix} \tag{4.9}$$

$$\vec{d} = HR_iTC \begin{pmatrix} x\tan\Theta_f \\ y\tan\Theta_f \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} x\tan\Theta_f \cos\Theta_i + \sin\Theta_i \\ y\tan\Theta_f \\ y\tan\Theta_f \sin\Theta_i - \cos\Theta_i \end{pmatrix} \tag{4.10}$$

$$\begin{aligned}\frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \theta_x} &= \frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \vec{o}}^T \begin{pmatrix} \frac{\partial (s_x+\theta_x)\cos\Theta_i}{\partial \theta_x} \\ 0 \\ \frac{\partial (s_x+\theta_x)\sin\Theta_i}{\partial \theta_x} \end{pmatrix} \\ &= \frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \vec{o}}^T \begin{pmatrix} \cos\Theta_i \\ 0 \\ \sin\Theta_i \end{pmatrix}\end{aligned} \tag{4.11}$$

Finally, the gradient of the mass attenuation prediction model with respect to the learnable center shift parameter (Eq. 4.11), $\frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \theta_x}$, can be derived by applying the chain rule using $\frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \vec{o}}$ and $\frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \vec{d}}$ as intermediate derivatives.

Algorithm 3: XRM-NeRF optimization algorithm

Input: Dataset of projection images I, Number of coarse evaluation c, Number of fine evaluation f, Learning rate α

Output: Trained XRM-NeRF network $G(\vec{o}+t\vec{d},\vec{d};\theta)$

Function `OPTIMIZE`(I, c, f, α)**:**

- Initialize θ, C
- Build matrix T (Eq. 4.7) given metadata for I
- **foreach** $i \in I$ **do**
  - Build matrix R_i (Eq. 4.7) given metadata for i
  - Set $\vec{o} = HR_iTC\begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}^T$ (Eq. 4.9)
  - Set $\vec{d} = HR_iTC\begin{pmatrix}x\tan\Theta_f & y\tan\Theta_f & -1 & 0\end{pmatrix}^T$ (Eq. 4.10)
  - Uniform sample ordered set $T_c \sim (t \mid t \in (0,1)) \mid |T_c| = c$
  - Set $G_i^c = \text{EVALUATION}(T_c, \vec{o}, \vec{d}, G(\vec{o}+t\vec{d},\vec{d};\theta))$
  - Perform coarse volumetric rendering $\hat{i_c} = exp(-\sum_{t_i \in T_c} G_i^c(t_i - t_{i-1}))$
  - Calculate sampling weights $w_i = (1-G_i^c)exp(-\sum_{i=0}^{i-1} G_i^c(t_i - t_{i-1}))$ [7]
  - Weighted sample ordered set $T_f \sim (t \mid t \in (0,1)) \mid |T_c| = c$
  - Set $G_i^f = \text{EVALUATION}(T_f, \vec{o}, \vec{d}, G(\vec{o}+t\vec{d},\vec{d};\theta))$
  - Perform fine volumetric rendering $\hat{i_f} = exp(-\sum_{t_i \in T_f} G_f^c(t_i - t_{i-1}))$
  - Optimize θ; $\theta = \theta - \alpha\frac{\partial[(i-\hat{i_f})^2+(i-\hat{i_c})^2]}{\partial\theta}$
- **end**
- **return** $G(\vec{o}+t\vec{d},\vec{d};\theta)$

Algorithm 3 combines all the equation derivations in this section and forms the complete training algorithm for XRM-NeRF. Each optimization step involves an initial uniform sampling procedure to produce a coarse reconstruction image, followed by the use of the predicted mass attenuation coefficient for each sample from the coarse reconstruction to conduct weighted sampling for generating a fine reconstruction image.

4.3 Dataset and Experiment Setup

4.3.1 Dataset

To evaluate the performance of our proposed XRM-NeRF, experiments were performed on seven distinct 3D XRM scenes obtained from different semiconductor product samples. Each scene includes metadata indicating the X-ray source offset from the rotation axis along the x and z axes. These offsets were used to generate the matrix T as outlined in Equation 4.7. In the 3D XRM datasets, the number of projection images ranges from 1401 to 2401, each with a resolution of 1013×1013 pixels, capturing the internal structure of a semiconductor sample at various rotation angles around the rotation axis. Table 4.1 provides a further breakdown of the details for each of the 3D XRM scenes.

The 3D XRM scenes used in this study represent integrated circuits (ICs), a specific type of semiconductor product. Out of the ten scenes, one corresponds to a non-defective IC, while the remaining nine correspond to defective ICs. These scenes were collected by manually running each individual IC through the 3D XRM measurement process. The goal was to ensure that the dataset captures both non-defective and defective characteristics, with a particular focus on defective ICs. Each scene is treated as a separate dataset, meaning that each scene consists of a collection of individual measurements and data points related to the specific IC, enabling independent evaluation of the reconstruction performance for each scene.

The first seven scenes in Table 4.1 were used in experiments to evaluate the performance of 3D reconstruction using XRM-NeRF (as indicated

by the 3D Reconstruction column set to True), while all ten scenes were used in experiments to assess the performance of prediction shift (as indicated by the Prediction Shift column set to True).

$$V(I_0, \vec{o}, T) = e^{-\sum_{t_i \in X} G(\vec{o} + \frac{t_i + t_{i-1}}{2}\vec{d}, \vec{d}; \theta)(t_i - t_{i-1})} \tag{4.12}$$

Dataset	# Projections	FOV (Θ_f)	Shift Actual (s_x^*)	Metadata (s_x)	Task 3D Reconstruction	Prediction Shift
Scene 1	1601	0.527	-0.0180	-0.0220	True	True
Scene 2	1601	0.961	-0.0351	-0.0367	True	True
Scene 3	2401	0.771	0.0168	0.0191	True	True
Scene 4	1401	0.610	-0.0008	0.0004	True	True
Scene 5	1601	0.864	-0.0243	-0.0240	True	True
Scene 6	1601	0.788	-0.0380	-0.0404	True	True
Scene 7	1801	0.981	-0.0037	-0.0050	True	True
Scene 8	1601	1.333	-0.0118	-0.0099	False	True
Scene 9	1601	1.192	-0.0063	-0.0063	False	True
Scene 10	1601	1.398	-0.0197	-0.0197	False	True

TABLE 4.1: Metadata for each TXRM dataset

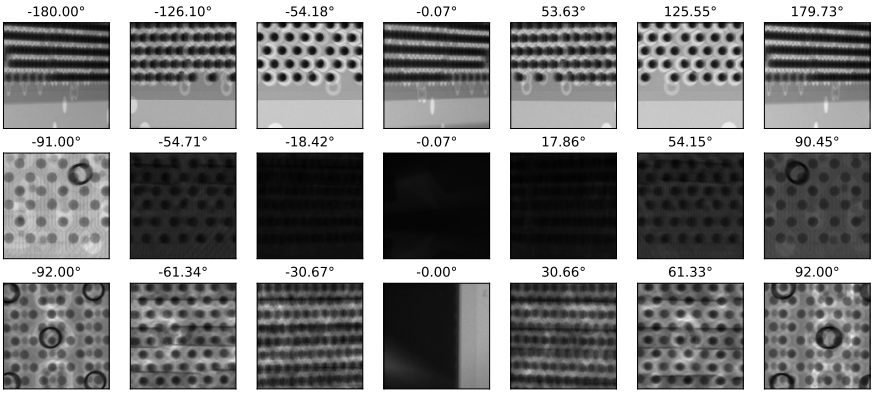

FIGURE 4.5: Subset of normalized projection images from 3D XRM scenes. Each row corresponds to the projection images captured at different angles from the same scene.

FOV (Θ_f) refers to the field of view for the corresponding 3D XRM scene. **Actual Shift** (s_x^*) refers to the optimal center shift, which was determined through visual confirmation by product engineers during data collection for the respective 3D XRM scene. **Center Shift** (s_x) refers to the center shift metadata collected by the 3D XRM scanner sensor during the measurement process. The rotation

angle of each projection image is used to derive its corresponding rotation matrix R_i, as specified by Equation 4.7. Each pixel value in the projection images is a scalar representing the X-ray intensity measured by the detector. To simplify the volumetric rendering process in XRM-NeRF (Eq. 4.4), each projection image was normalized using a reference frame that represents the initial X-ray intensity, absent of the sample. This normalization simplifies the representation of the volumetric rendering process in XRM-NeRF, as shown in Equation 4.12.

4.3.2 XRM-NeRF Architecture

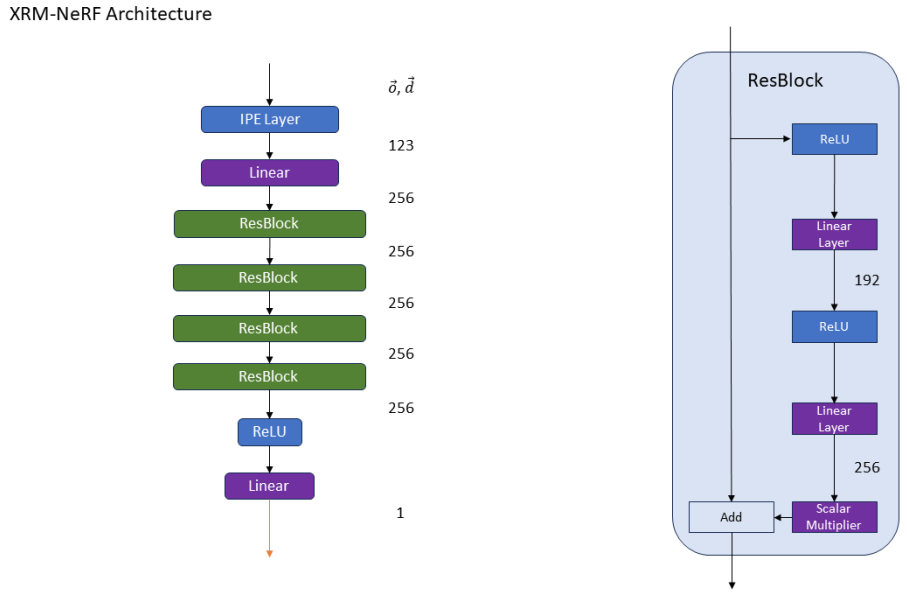

FIGURE 4.6: XRM-NeRF model architecture for reconstructing 3D XRM images. The orange arrow indicates shifted Softplus activation as described by Barron et al. [1]. The numbers to the right of each layer indicate the number of output neurons.

Figure 4.6 shows a detailed illustration of the XRM-NeRF architecture which uses a ResNet-style [65] feed-forward neural network architecture for the XRM-NeRF model $G(\vec{o}+t\vec{d}, \vec{d}; \theta)$ to learn and predict the mass attenuation coefficient for a 3D XRM scene. The rationale for selecting a ResNet-style architecture is to enhance the gradient flow of the XRM-NeRF model with respect to the origin vector [65]. As shown in equation 4.11, the gradient of the learnable center shift parameter is a linear combination $\frac{\partial G(\vec{o}+t\vec{d},\vec{d};\theta)}{\partial \vec{o}}$. Hence, improving the gradient flow can potentially

improve the learning dynamics of the center shift parameter. The XRM-NeRF model sequentially stacks an integrated positional encoding layer [1], a linear layer, four residual block layers, and an output linear layer. The hyperparameter L for the integrated positional encoding layer is set to 16, similar to Barron et al [1], after empirically confirming that larger L values will result in the output values below the machine epsilon for the 3D XRM dataset. There are 256 output neurons for the linear and residual block layers. In the residual path of each block, an encoder linear layer is used to project the input to 192 neurons followed by a decoder linear layer to project the 192 neurons back to 256 neurons. This is similar to the bottleneck layer proposed by He et al. [65] to reduce the computational cost of reconstruction and inference. The output linear layer then transforms the 256 feature vector into a scalar which represents the predicted mass attenuation coefficient.

4.3.3 Experiment Setup

Experiments were performed to assess the effectiveness of XRM-NeRF in generating high-quality images from novel viewing angles for the 3D XRM datasets detailed in section 4.3.1. Projection images from each dataset were organized based on their viewing angles, with every N^{th} image chosen from the sorted dataset for the training set, while the remaining images were designated as the test set. **The selection strategy for the training images simulates the effect of reducing projection scans acquired by 3D XRM scanners by augmenting the rotation angle between each projection scan.**

The training of the XRM-NeRF model involves adjusting the neural network weights to minimize the mean squared error between the reconstructed images and the original projection images.

The XRM-NeRF model weight parameters were initialized by random sampling the weights from a normal distribution with mean and variances that encourage the activation, after a forward pass through the weight layers, to conform to the standard normal distribution. For weight layers that come after a ReLU non-linear activation layer, the weights were sampled from a normal distribution with mean 0. and standard deviation of $\sqrt{\frac{2}{fan\_in}}$ [66]. For other weight layers, the weights were sampled from a normal distribution with mean 0. and standard deviation of $\sqrt{\frac{1}{fan\_in}}$. The fan_in symbol in the standard deviations represents the number of

elements in the input vector for each linear layer. The scalar multipliers in each ResBlock layer were initialized to 1.

An AdamW optimizer [67, 68] was used to optimize the XRM-NeRF model; For all experiments in the datasets, the learning rate was set to 0.005, β_1 was set to 0.9, and β_2 was set to 0.999. A L_2 weight decay value of 0.001 was added to encourage the reduction of the magnitude of the learned weight and reduce model complexity [69]. The learning rate and weight decay hyperparameters were selected through a greedy grid search process as outlined in the ablation study in section 4.4.1.

An ablation study was conducted to explore the correlation between the reconstruction quality of novel views using XRM-NeRF and the number of training images employed during reconstruction; In the ablation study, N was set to values of 4, 6, 8, 12, 16, 32, 64, 128 which corresponds to reconstructing the novel views with 25%, 16.6̇%, 12.5%, 8.3̇%, 6.25%, 3.125%, 1.5625%, 0.78125% of the number of projection images respectively.

All experiments in this chapter were performed using mixed precision (BFloat16) to optimize the XRM-NeRF models.

4.4 Results

XRM-NeRF is evaluated on the Peak Signal To Noise Ratio (PSNR) [70] and Structural Similarity Index Measure (SSIM) [71, 72] by performing a prediction of the projection image for viewing angles in the test set and comparing it to its corresponding ground truth projection image. The SSIM is assessed by employing an 11x11 Gaussian window to aggregate the local neighborhood mean and variance for every pixel. The arithmetic mean of the PSNR and SSIM values of the respective 3D XRM dataset obtained are shown in Table 4.2 and Table 4.3. The average PSNR on the test set for each scene is between 37 dB and 50 dB. The average SSIM on the test set for each scene is between 0.9803 and 0.9924. The comparison plots, illustrating the correlation between the percentage of images utilized for optimizing the XRM-NeRF model and their corresponding PSNR and SSIM values, are shown in Figure 4.7 and 4.8. The comparison plots indicate a pattern of improving XRM-NeRF reconstruction quality, which reaches a plateau when more than 16.67% of projection images were utilized to optimize the XRM-NeRF model.

	Average PSNR							
% Images / **Dataset**	0.78125	1.5625	3.125	6.25	8.3̇	12.5	16.6̇	25
Scene 1	36.475	41.952	43.589	43.432	44.093	45.053	44.715	44.748
Scene 2	22.990	25.821	31.685	35.187	36.688	37.667	37.184	37.364
Scene 3	40.933	44.905	47.893	48.618	49.025	49.728	49.433	49.609
Scene 4	31.746	38.532	41.444	42.680	44.777	44.250	45.495	45.101
Scene 5	31.032	35.405	39.660	41.515	43.479	44.911	44.919	45.068
Scene 6	33.827	36.521	40.874	43.466	45.815	45.004	47.000	46.160
Scene 7	31.114	37.012	41.380	44.460	46.426	47.640	47.744	46.941

TABLE 4.2: A quantitative ablation study result of the PSNR between XRM-NeRF predicted images and test set images.

	Average SSIM							
% Images / **Dataset**	0.78125	1.5625	3.125	6.25	8.3̇	12.5	16.6̇	25
Scene 1	0.9280	0.9687	0.9704	0.9712	0.9697	0.9712	0.9713	0.9701
Scene 2	0.8518	0.8835	0.9086	0.9216	0.9331	0.9325	0.9336	0.9289
Scene 3	0.9777	0.9868	0.9890	0.9909	0.9917	0.9924	0.9920	0.9922
Scene 4	0.9172	0.9758	0.9801	0.9790	0.9831	0.9814	0.9839	0.9809
Scene 5	0.9148	0.9613	0.9784	0.9821	0.9849	0.9867	0.9856	0.9860
Scene 6	0.9132	0.9501	0.9661	0.9741	0.9792	0.9768	0.9803	0.9779
Scene 7	0.8984	0.9534	0.9803	0.9885	0.9903	0.9913	0.9914	0.9915

TABLE 4.3: A quantitative ablation study result of the SSIM between XRM-NeRF predicted images and test set images.

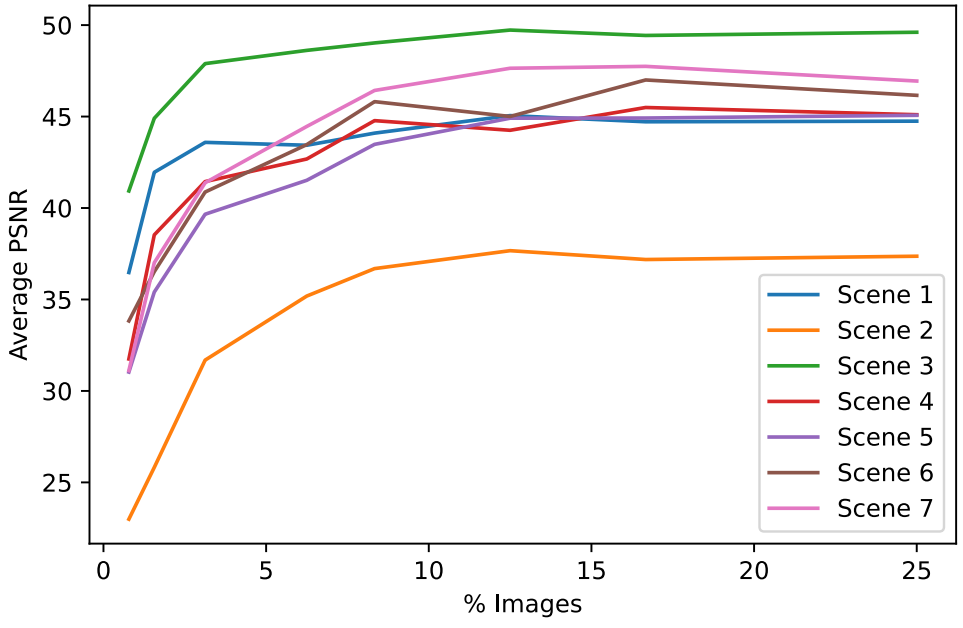

FIGURE 4.7: Plot of the % of training images used for optimizing XRM-NeRF model vs Average PSNR value for each XRM scene.

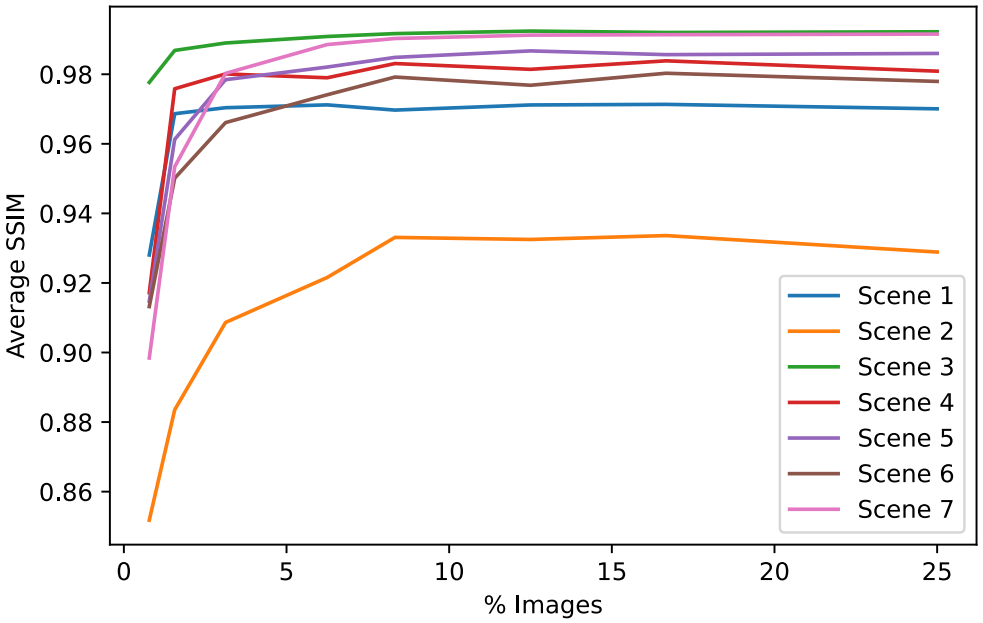

FIGURE 4.8: Plot of the % of training images used for optimizing XRM-NeRF model vs Average SSIM value for each XRM scene.

The qualitative results of XRM-NeRF reconstructions are reported in Figure 4.9, 4.10, 4.11, and 4.12 utilizing 16.67%, 25%, 16.67%, and 16.67% of the number of projection images respectively for optimizing the XRM-NeRF model. The figures demonstrate that XRM-NeRF is capable of faithful reconstruction of novel views for 3D XRM scenes; Despite having lower PSNR values for more complex scenes at the larger viewing angles, the ability of XRM-NeRF to accurately capture scene structure is evident through consistently high SSIM values.

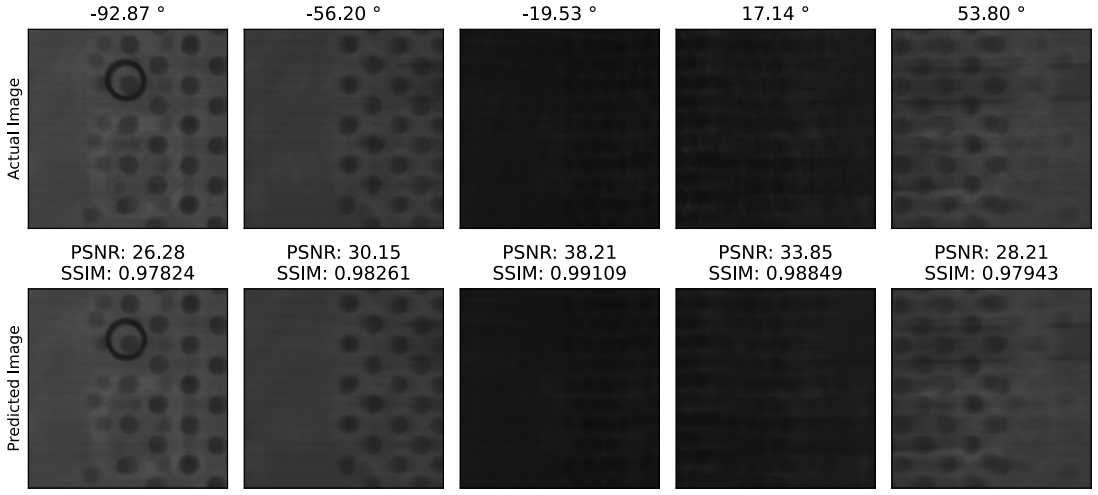

FIGURE 4.9: Comparison between reconstructed projection images and actual images from novel viewing angles in the test set for Scene 4 dataset.

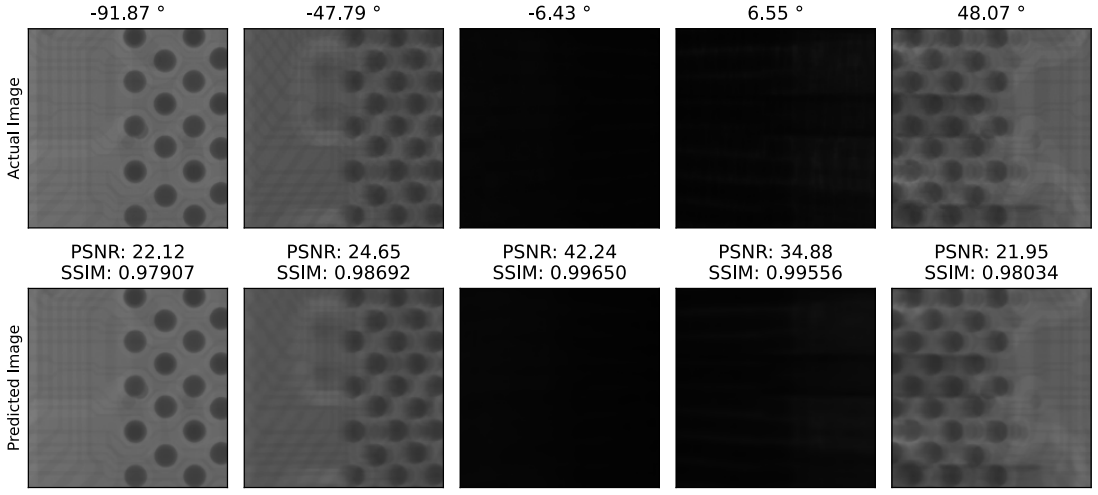

FIGURE 4.10: Comparison between reconstructed projection images and actual images from novel viewing angles in the test set for Scene 5 dataset.

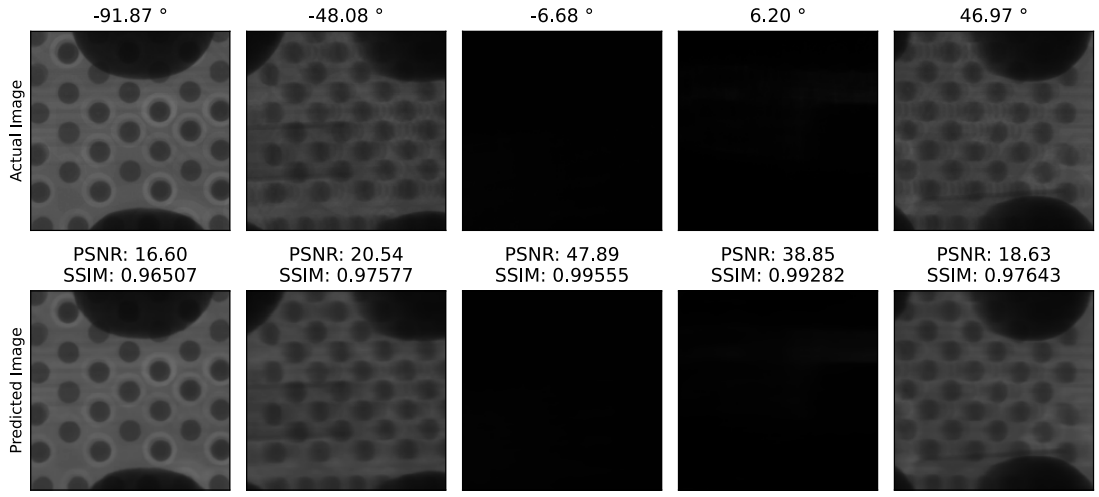

FIGURE 4.11: Comparison between reconstructed projection images and actual images from novel viewing angles in the test set for Scene 6 dataset.

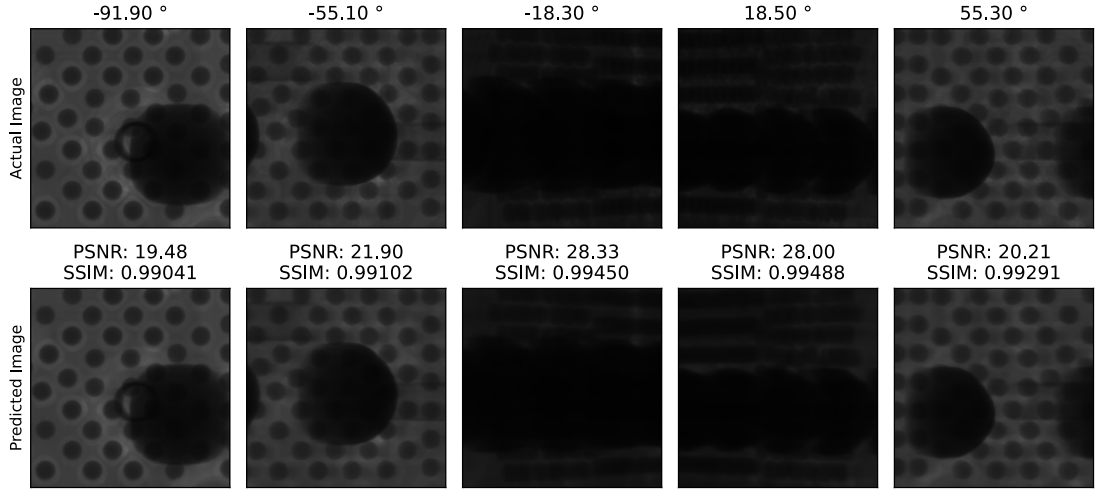

FIGURE 4.12: Comparison between reconstructed projection images and actual images from novel viewing angles in the test set for Scene 7 dataset.

The computational cost of optimizing the XRM-NeRF model for a single epoch in each experiment is reported in Table 4.4. This cost was estimated by benchmarking the Bfloat16 matrix multiplication performance in PyTorch [73] (in TFLOPs) and multiplying it by the time taken for optimization. Figure 4.13 plots the computational cost (in PFLOPs) against the number of training images used to optimize the XRM-NeRF model. Each scatter point represents an experiment instance, and

the regression line, based on least-squares fitting, indicates that the computational cost is approximately 0.84 PFLOPs per projection image. Additionally, Table 4.5 interpolates the computational cost for using XRM-NeRF to reduce the number of projection scans in a 3D XRM dataset with 2401 projection images. **This table also includes the expected optimization time and potential time savings, calculated by extrapolating the computational cost (in PFLOPs) using the MI250's theoretical peak performance of 362.1 TFLOPs. The results demonstrate the potential benefits of using XRM-NeRF to streamline the 3D XRM measurement and reconstruction process.**

Figure 4.14 clarifies the relationship between the 3D XRM measurement and XRM-NeRF optimization processes. The blue bar in the diagram represents the 3D XRM measurement phase. Following this, the red bar indicates the XRM-NeRF optimization stage, which introduces additional computational time compared to the original process. However, the key advantage of our method is that the XRM-NeRF optimization reduces the overall time by shortening the 3D XRM measurement phase. While the optimization step does add extra compute time, the reduction in measurement time leads to an overall decrease in the total time required for the entire process. This figure helps visualize how our approach improves efficiency by streamlining the 3D XRM measurement phase and offsetting the added computational cost of the XRM-NeRF optimization.

% Images **Dataset**	0.78125	1.5625	3.125	6.25	8.3̇	12.5	16.6̇	25
Scene 1	242.52	486.53	953.94	1887.61	2499.68	3758.57	4994.56	7505.99
Scene 2	242.43	487.06	953.30	1886.16	2509.59	3756.77	4990.32	7493.59
Scene 3	355.71	710.45	1421.95	2821.75	3756.87	5626.47	7500.19	11261.10
Scene 4	205.51	411.15	822.19	1643.93	2185.99	3291.63	4373.06	6561.39
Scene 5	243.08	486.12	952.48	1886.56	2504.44	3760.05	4986.66	7509.22
Scene 6	242.73	486.60	953.74	1889.11	2501.86	3752.31	4992.18	7499.10
Scene 7	280.22	542.60	1064.74	2111.33	2827.81	4225.30	5619.86	8440.26

TABLE 4.4: Time (seconds) taken for each epoch of XRM-NeRF optimization for the corresponding dataset and % images.

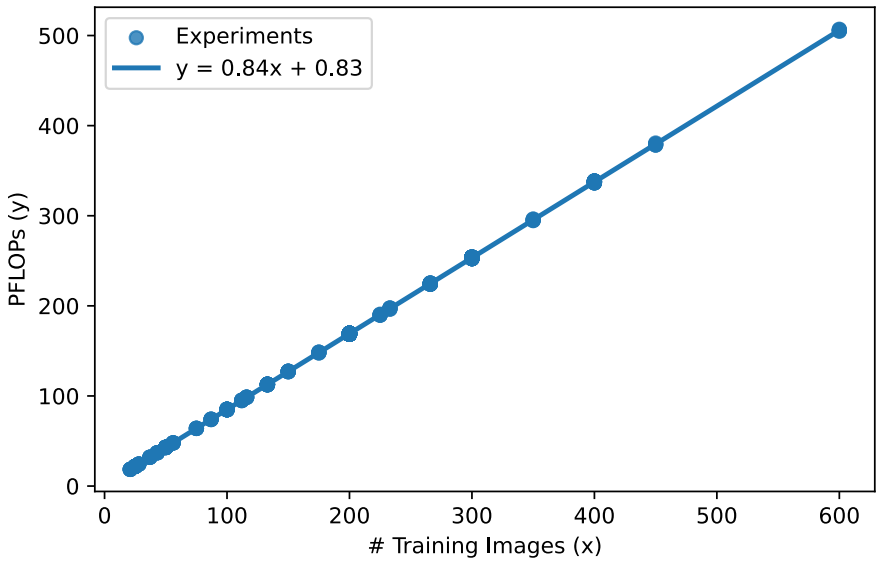

FIGURE 4.13: Plot of the number of images against the computational requirements for a single epoch of XRM-NeRF optimization.

% Images	Scan Time	Computation Cost	Optimization Time	Time Saved
100%	320 min	N/A	N/A	N/A
20%	64 min	403 PFLOPs	19 min	237 min
10%	32 min	202 PFLOPs	9 min	279 min
5%	16 min	101 PFLOPs	5 min	299 min
2.5%	8 min	50 PFLOPs	2 min	310 min

TABLE 4.5: Extrapolation of time saved (minutes) using XRM-NeRF to reconstruct a 2401 projection image 3D XRM dataset.

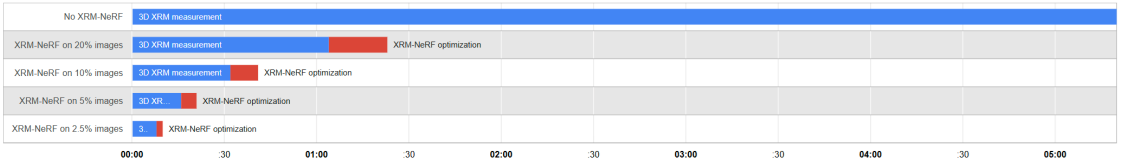

FIGURE 4.14: Comparison using a timeline diagram of the various processes and its corresponding time taken for the proposed XRM-NeRF augmented reconstruction process.

To assess the accuracy of the fine-tuning parameter learned by XRM-NeRF, the predicted shift value is calculated by adding the fine-tuning parameter to the center shift value extracted from the metadata of corresponding 3D XRM scenes, with the results presented in Table 4.6. **The table also includes two columns that**

report the delta between the metadata-reported shift and the actual shift, as well as the delta between the predicted shift and the actual shift. It can be noted that, with the exception of Scene 9 and Scene 10, which have a delta of 0 for the metadata-reported shift, six out of the eight scenes show a better prediction compared to the shift reported by the metadata. With an explained variance [74] value of 0.7453, this analysis underscores the proficiency of XRM-NeRF in approximating the center shift of 3D XRM scenes towards the actual shift values. This ability to refine center shift values can potentially enable XRM-NeRF to learn a better 3D geometrical representation of these XRM scenes.

	Shift			Delta	
	Actual	Metadata	Predicted	Metadata	Predicted
Dataset					
Scene 1	-0.0180	-0.0220	-0.0192	-0.004	0.0012
Scene 2	-0.0351	-0.0367	-0.0356	0.0016	0.0005
Scene 3	0.0168	0.0191	0.0172	-0.0023	-0.0004
Scene 4	-0.0008	0.0004	-0.0005	-0.0012	-0.0003
Scene 5	-0.0243	-0.0240	-0.0247	0.0003	0.0004
Scene 6	-0.0380	-0.0404	-0.0375	0.0024	-0.0005
Scene 7	-0.0037	-0.0050	-0.0052	0.0013	0.0015
Scene 8	-0.0118	-0.0099	-0.0134	-0.0019	0.0016
Scene 9	-0.0063	-0.0063	-0.0059	0	-0.0004
Scene 10	-0.0197	-0.0197	-0.0183	0	-0.0014
Explained Variance					**0.7453**

TABLE 4.6: Table of center shift prediction results for each corresponding 3D XRM scene.

4.4.1 Ablation Study

An ablation study was employed to identify the optimal set of hyperparameters for optimizing the XRM-NeRF model. Each hyperparameter's optimal value was determined through a grid search, and the best combination of hyperparameter values was established via a greedy search process.

The **learning rate** hyperparameter was assessed on values 0.05, 0.005, 0.0005, and 0.00005; For each value, a random 3D XRM dataset was selected and an XRM-NeRF model was optimized to reconstruct the novel views in the validation set.

The experiment was repeated on three different 3D XRM datasets for the same hyperparameter value and the average PSNR validation score is computed, with values as shown in Figure 4.15. The average PSNR to learning rate curve exhibited an inverted U-shape trend and the optimal learning rate of 0.005 was identified.

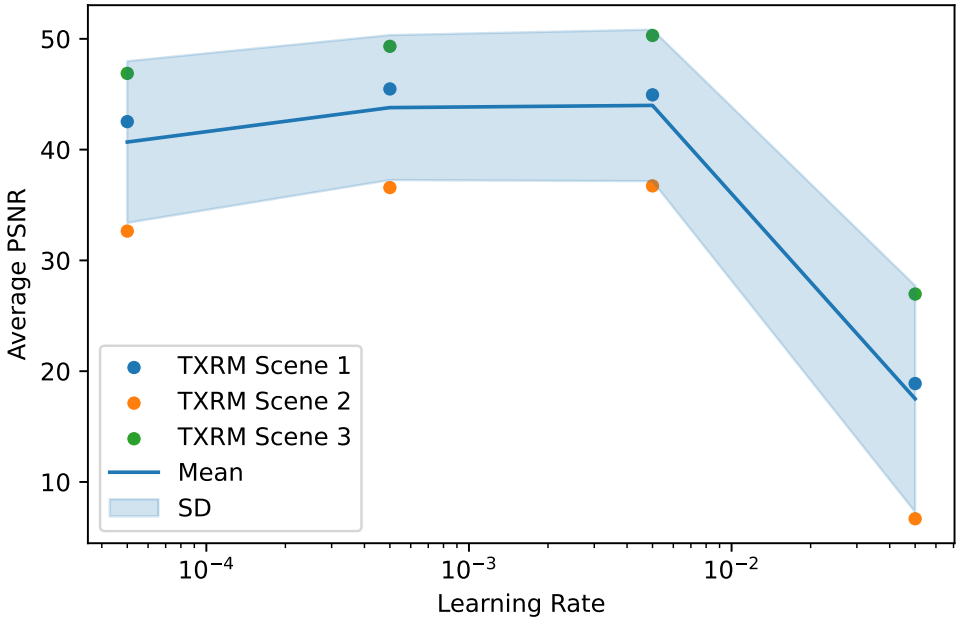

FIGURE 4.15: Learning Rate vs Average PSNR plot.

The **weight decay** hyperparameter was assessed on values 0.01, 0.001, 0.0001, and 0.00001. The XRM-NeRF model for each hyperparameter value was optimized with a learning rate of 0.005 and the average validation PSNR score was calculated. The average PSNR validation scores reported in Figure 4.16 were normalized against the average PSNR validation score of the baseline with a weight decay of 0.0. The optimal weight decay value was established at 0.001.

In addition to the learning rate and weight decay hyperparameters, an ablation study was performed on the regularization hyperparameter described by Barron et al to discourage floaters and background collapse artifacts [35]. The **regularization** hyperparameter was assessed on values 0.01, 0.001, 0.0001, and 0.00001. The XRM-NeRF model for each hyperparameter value was optimized with a learning rate of 0.005 and weight decay of 0.001; The average validation PSNR scores were subsequently calculated and normalized against the average PSNR validation score of the baseline with a regularization value of 0.0. The normalized PSNR scores are

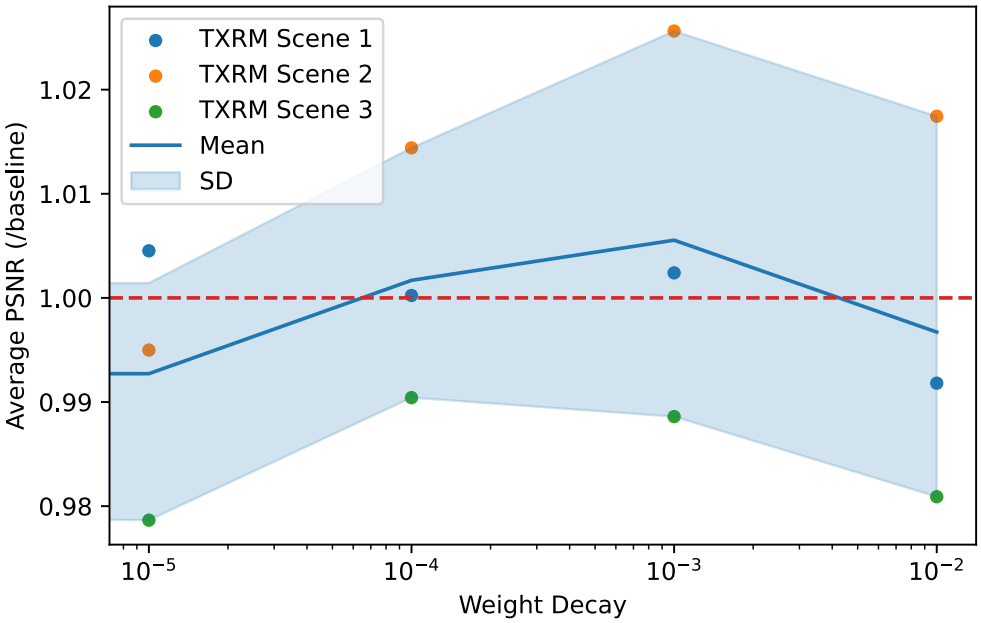

FIGURE 4.16: Weight Decay vs Average PSNR plot.

reported in Figure 4.17. None of the regularization hyperparameter values showed an improved average score against the baseline model, hence the regularization is eventually not used. A possible explanation for the degradation of reconstruction quality could be due to the intrinsic differences between the RGB scenes and the XRM scenes. In RGB scenes, such as those found in the ShapeNet [75] and LLFF [76] datasets, a significant portion of the scene comprises empty spaces with minimal or zero density, whereas, in 3D XRM scenes, the background density is often non-zero. Given that the regularization term [35] aims to promote a compact representation of density, the density distribution in an XRM scene may conflict with such assumptions.

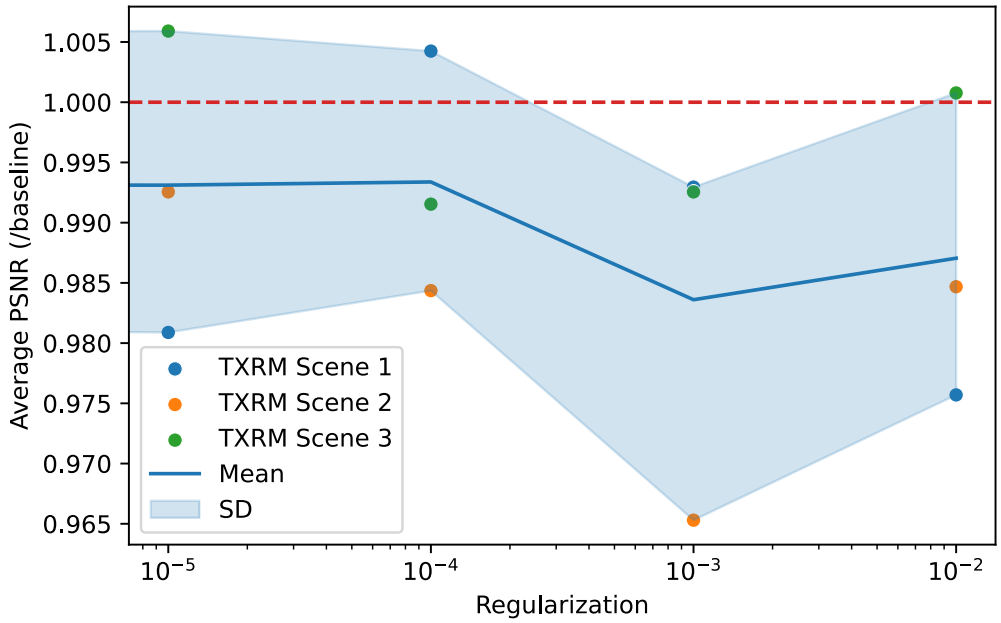

FIGURE 4.17: Regularization vs Average PSNR plot.

4.5 Summary

This chapter presents the XRM-NeRF approach, aimed at overcoming the challenges of implementing NeRF to reduce the number of projection scans for 3D XRM reconstruction while maintaining quality. Both quantitative and qualitative evidences support the ability of XRM-NeRF to reconstruct projection images for new viewing angles using only 16.67% of the full set of projection images. XRM-NeRF effectively reduces the number of projection scans needed by XRM scanners, although it imposes additional computational costs needed to optimize the model required for reconstructing the novel views. Given the high availability of GPU computing in cloud data centers, XRM-NeRF demonstrates potential in accelerating 3D XRM failure analysis by speeding up scan times and subsequently performing XRM-NeRF model optimization for multiple samples in parallel to amortize the extra time required for model optimization.

The next chapter presents iXRM-NeRF, an improved XRM-NeRF, that enhances its applicability for semiconductor failure analysis by reducing the computational

cost associated with optimizing the model and the synthesis of cross-sectional views of reconstructed 3D XRM scenes.

Chapter 5

iXRM-NeRF

This chapter presents the iXRM-NeRF (Improved XRM-NeRF) that is used to improve the convergence rate and reconstruction quality of XRM-NeRF and generate cross-sectional views of 3D XRM samples obtained using XRM-NeRF.

5.1 Limitation of XRM-NeRF

Despite XRM-NeRF demonstrating that it can reduce the number of projections required for 3D XRM reconstruction, it comes at the cost of additional computation requirements for optimizing the XRM-NeRF model. The XRM-NeRF optimization process involves iterative gradient-based weight parameter updates for an XRM-NeRF model that has been randomly initialized. An expanding body of research, as cited in works like those by Huisman et al. [77] and Dauphin et al. [78], hypothesized that gradient-based optimization is highly sensitive to initial parameters; Therefore, the random initialization in of the XRM-NeRF model may result in slower convergence rates. Thus, exploring alternative strategies for initializing weight parameters is needed to enhance the convergence efficiency of XRM-NeRF.

Another challenge for XRM-NeRF in semiconductor failure analysis is its dependence on the perspective projection camera model for synthesizing novel views. In semiconductor failure analysis, a common downstream task involves the generation of cross-sectional views for defect analysis. This process necessitates Euclidean

isometry to preserve the size and shape of the defect, which is incompatible with the perspective camera model utilized by mip-NeRF for novel view synthesis.

The iXRM-NeRF, proposed here will address these two challenges: sub-optimal initialization of the XRM-NeRF model and the application of XRM-NeRF to generate cross-sectional views.

5.2 iXRM-NeRF

5.2.1 Meta-Learning

iXRM-NeRF extends XRM-NeRF by initializing it with weight parameters learned using Reptile [53]. The learning objective of Reptile is to find a set of initial weight parameters that reduces the final objective loss by updating the initial weight parameters closer to the optimal weights of each task. It is characterized by Equation 5.1, where θ represents the weight parameters, T represents a set of 3D XRM scenes, and M represents the XRM-NeRF optimization process (Alg. 4) that returns an optimized set of weights.

$$L_T(\theta) = \sum_{t \in T} ||\theta - M(t;\theta)||_2^2 \tag{5.1}$$

To reduce computational requirements in practice, instead of fully optimizing an XRM-NeRF model for a 3D XRM scene to calculate $M(t;\theta)$, a reduced subset of K number of optimization steps are performed [53], and the initial parameters θ are iteratively updated using gradient descent. The complete algorithm outlining the learning process for the initial weight parameters of iXRM-NeRF can be found in algorithm 4. iXRM-NeRF introduces an extension of prior research [50] by adapting the concept to XRM-NeRF models, with the objective of enhancing convergence through the learning of a refined set of initial weights. This novel approach aims to contribute significant improvements to the optimization process of iXRM-NeRF models.

Algorithm 4: iXRM-NeRF training algorithm

Input: Dataset of 3D XRM scenes D, Number of optimization steps k, Number of coarse evaluation c, Number of fine evaluation f, Inner learning rate α_i, Outer learning rate α_o

Output: Weight parameters to initialize XRM-NeRF network θ

Function `METATRAIN`(D, k, c, f, α_i, α_o)**:**

Initialize θ

while $L_T(\theta) > \epsilon$ **do**

Sample 3D XRM scene, $I \sim D$

Sample subset of 3D XRM scene, $\hat{I} \sim I \mid \hat{I} \subset I \wedge |\hat{I}| = k$

Optimize iXRM-Model, $\hat{\theta}$ =OPTIMIZE($\hat{I}$, c, f, α_i; θ)

Set $\theta = \theta - \alpha_o(\hat{\theta} - \theta)$

end

return θ

5.2.2 Synthesizing Cross-Sectional Views

To more accurately generate the cross-sectional views of a 3D XRM scene, iXRM-NeRF replaces the perspective camera model of XRM-NeRF with an orthographic camera model during the synthesis process. In the orthographic camera model, each pixel is synthesized by casting a virtual ray from a different origin and sampling Gaussian along a cylinder. **Figure 5.2 provides a visual illustration comparing the XRM-NeRF perspective camera model and the proposed iXRM-NeRF orthographic camera model. The orthographic projection model in iXRM-NeRF avoids this distortion by using parallel rays and a cylindrical sampling region, making it more suitable for generating precise, undistorted 3D slices and cross-sections in XRM applications.**

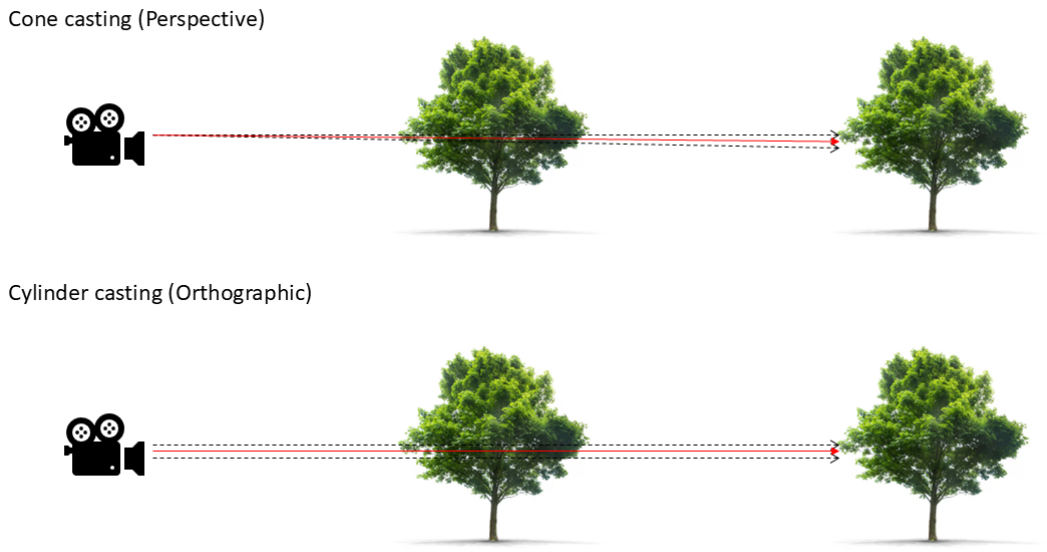

FIGURE 5.1: Comparison of the perspective projection model in XRM-NeRF and the orthographic projection model in iXRM-NeRF. The black dotted lines represent the camera's field of view. The XRM-NeRF perspective camera casts a cone to account for the expanding field of view with increasing distance from the camera, while the iXRM-NeRF orthographic camera casts a cylinder to preserve a consistent field of view and eliminate distortion.

The origin for each pixel is calculated by offsetting the origin used by the perspective camera model with the (x, y) coordinate of each pixel, while an axis-aligned cylinder can be parameterized as $(x, y, z) = \varphi(r, t, \theta) = (rcos\theta, rsin\theta, t) \mid \theta \in [0, 2\pi), t \geq 0$.

To ensure compatibility with the training process of XRM-NeRF, the mean and variance of a randomly selected Gaussian along the virtual cylinder are calculated and encoded as part of the integrated positional encoding. This is then fed to the XRM-NeRF model as input to generate the corresponding mass attenuation coefficient prediction.

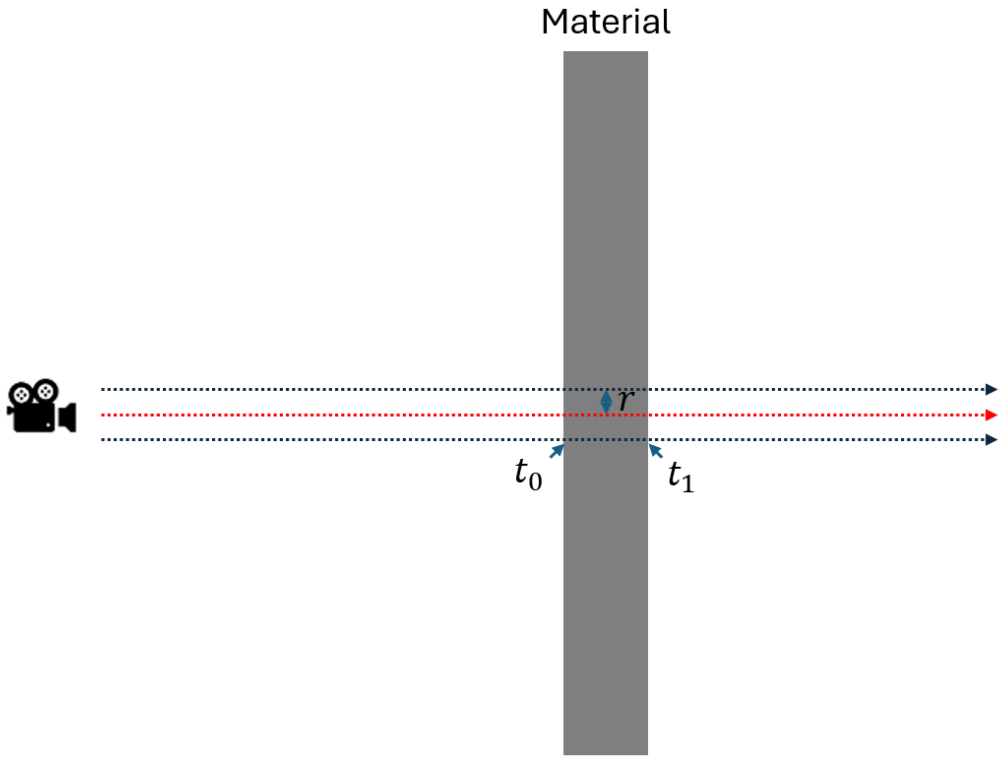

FIGURE 5.2: An illustration of the iXRM orthographic camera model. t_0 represents the start of the material in the x coordinate and t_1 represents the end of the material in the x coordinate. r represents the radius of the cylinder cast by the iXRM-NeRF model.

The probability density function of uniform selection of points in a cylinder can be expressed as $\frac{1}{V}$, where V is the volume of the cylinder. Through a change of variable in Equation 5.2 the volume of the cylinder can be expressed as shown in Equation 5.3. Figure 5.2 provides an illustration of the iXRM orthographic camera model, with the variables used in the equations clearly denoted in the diagram.

$$\begin{aligned} \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z &= \mid det(D\varphi)(r,t,\theta) \mid \mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}t \\ &= \begin{vmatrix} cos\theta & sin\theta & 0 \\ -rsin\theta & rcos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix} \mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}t \\ &= rcos^2\theta + rsin^2\theta\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}t \\ &= r\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}t \end{aligned} \tag{5.2}$$

$$
\begin{aligned}
V &= \int_{t_0}^{t_1} \int_{0}^{2\pi} \int_{0}^{\dot{r}} r \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}t \\
&= \frac{\dot{r}^2}{2} \cdot 2\pi \cdot (t_1 - t_0) \\
&= \pi \dot{r}^2 (t_1 - t_0)
\end{aligned}
\tag{5.3}
$$

Combining Equation 5.2 and 5.3, the mean $\mathbf{E}[t]$ and variance $\mathrm{Var}[t]$ of the parametric variable t are derived in Equation 5.4 and 5.5 respectively.

$$
\begin{aligned}
\mathbf{E}[t] &= \frac{1}{V} \int_{t_0}^{t_1} \int_{0}^{2\pi} \int_{0}^{\dot{r}} rt \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}t \\
&= \frac{1}{V} \cdot \frac{\dot{r}^2}{2} \cdot 2\pi \cdot (\frac{t_1^2}{2} - \frac{t_0^2}{2}) \\
&= \frac{t_1^2 - t_0^2}{2 \cdot (t_1 - t_0)} \\
&= \frac{t_1 + t_0}{2}
\end{aligned}
\tag{5.4}
$$

$$
\begin{aligned}
\mathrm{Var}[t] &= \mathbf{E}[t^2] - \mathbf{E}[t]^2 \\
&= \frac{1}{V} \int_{t_0}^{t_1} \int_{0}^{2\pi} \int_{0}^{\dot{r}} rt^2 \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}t - \mathbf{E}[t]^2 \\
&= \frac{1}{V} \frac{\dot{r}^2}{2} \cdot 2\pi \cdot (\frac{t_1^3}{3} - \frac{t_0^3}{3}) - \mathbf{E}[t]^2 \\
&= \frac{t_1^3 - t_0^3}{3 \cdot (t_1 - t_0)} - \mathbf{E}[t]^2 \\
&= \frac{t_1^2 + t_1 t_0 + t_0^2}{3} - \frac{(t_1 + t_0)^2}{4} \\
&= \frac{t_1^2 - 2t_1 t_0 + t_0^2}{12} \\
&= \frac{(t_1 - t_0)^2}{12}
\end{aligned}
\tag{5.5}
$$

Similarly, the mean and variance of the variables x and y are derived in equations 5.6, 5.7, 5.8, and 5.9 respectively.

$$\begin{aligned}\mathbf{E}[x] &= \frac{1}{V}\int_{t_0}^{t_1}\int_{0}^{2\pi}\int_{0}^{\dot{r}} r \cdot r cos\theta \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}t \\ &= \frac{1}{V}\cdot\frac{\dot{r}^3}{3}\cdot 0\cdot(t_1 - t_0) \\ &= 0\end{aligned} \tag{5.6}$$

$$\begin{aligned}\mathrm{Var}[x] &= \mathbf{E}[x^2] - E[x]^2 \\ &= \frac{1}{V}\int_{t_0}^{t_1}\int_{0}^{2\pi}\int_{0}^{\dot{r}} r \cdot (r cos\theta)^2 \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}t \\ &= \frac{1}{V}\int_{t_0}^{t_1}\int_{0}^{2\pi}\int_{0}^{\dot{r}} r^3 cos^2\theta \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}t \\ &= \frac{1}{V}\frac{\dot{r}^4}{4}\cdot\pi\cdot(t_1 - t_0) \\ &= \frac{\dot{r}^2}{4}\end{aligned} \tag{5.7}$$

$$\begin{aligned}\mathbf{E}[y] &= \frac{1}{V}\int_{t_0}^{t_1}\int_{0}^{2\pi}\int_{0}^{\dot{r}} r \cdot r sin\theta \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}t \\ &= \frac{1}{V}\cdot\frac{\dot{r}^3}{3}\cdot 0\cdot(t_1 - t_0) \\ &= 0\end{aligned} \tag{5.8}$$

$$\begin{aligned}\mathrm{Var}[y] &= \mathbf{E}[x^2] - E[x]^2 \\ &= \frac{1}{V}\int_{t_0}^{t_1}\int_{0}^{2\pi}\int_{0}^{\dot{r}} r \cdot (r sin\theta)^2 \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}t \\ &= \frac{1}{V}\int_{t_0}^{t_1}\int_{0}^{2\pi}\int_{0}^{\dot{r}} r^3 sin^2\theta \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}t \\ &= \frac{1}{V}\frac{\dot{r}^4}{4}\cdot\pi\cdot(t_1 - t_0) \\ &= \frac{\dot{r}^2}{4}\end{aligned} \tag{5.9}$$

During the cross-sectional view synthesis process, the mean and variance of the

variables derived in the above equations are used to replace the mean and variance calculation used in the integrated positional encoding during the optimization process of XRM-NeRF. This ensures that the change from cone to cylinder projection, due to the utilization of an orthographic camera model, is conveyed to the XRM-NeRF model when predicting the mass attenuation coefficient.

5.3 Dataset and Experiment Setup

The experiments were performed with the ten distinct 3D XRM scenes detailed in section 4.3.1 that were used in Chapter 4. The iXRM-NeRF model is also based on the same neural network architecture of Chapter 4.

In the context of meta-learning, each 3D XRM scene is regarded as a **task**, with the seven scenes collectively forming the task distribution. The objective is to discover an optimal initialization of the weight parameters to expedite convergence across this task distribution. Due to the limited number of tasks in the task distribution, a leave-one-out cross-validation strategy [79] was implemented to assess the generalization capability of iXRM-NeRF. A total of seven experiments were conducted. In each experiment run, five tasks were designated for the training set, one for validation, and one for testing. Algorithm 4 was employed to derive a set of initialization weights from the training set, while the validation set was utilized to monitor the generalization capability of the learned initialization weights throughout the training phase. The outer learning rate was fixed at 1.0. Within the inner loop, the weight parameters were optimized over 50 steps, with a batch size of 4096, an inner learning rate of 0.0005, and a weight decay of 0.01 for each task. The weight decay hyperparameter was determined through a grid search process in Section 5.4.1.4. An early stopping rule was employed to end the training process when the validation loss stopped improving. The learned initialization weights were utilized to initialize an iXRM-NeRF model, which was subsequently optimized to reconstruct projection images within the testing dataset.

5.4 Results

5.4.1 iXRM-NeRF Meta-Learning

5.4.1.1 Learning Priors

Qualitative comparisons by visual inspection between the projection reconstruction of various scenes using the iXRM-NeRF model and the XRM-NeRF model are shown in Figure 5.3 and Figure 5.4. In the qualitative comparisons, **Meta Init.** represents generated projection views of the unoptimized iXRM-NeRF model and **Random Init.** represents generated projection views of the unoptimized XRM-NeRF model. **Meta Opt.** and **Random Opt.** represent generated projection views of the corresponding model after optimization.

In both figures, the projection images generated by the unoptimized XRM-NeRF model show no visible structures, indicating that randomly initialized weights provide no information about the general structure of 3D XRM scenes. In contrast, the presence of circular blobs in the projection images generated by the unoptimized iXRM-NeRF model shows that the meta-learning phase of the iXRM-NeRF model learns a set of weight parameters that can comprehend the general structure of 3D XRM scenes for semiconductor products, which consist of bumps between layers [80]. The results show that the unoptimized iXRM-NeRF model provides valuable insights into the structure of the semiconductors that can serve as useful priors that guides the optimization process.

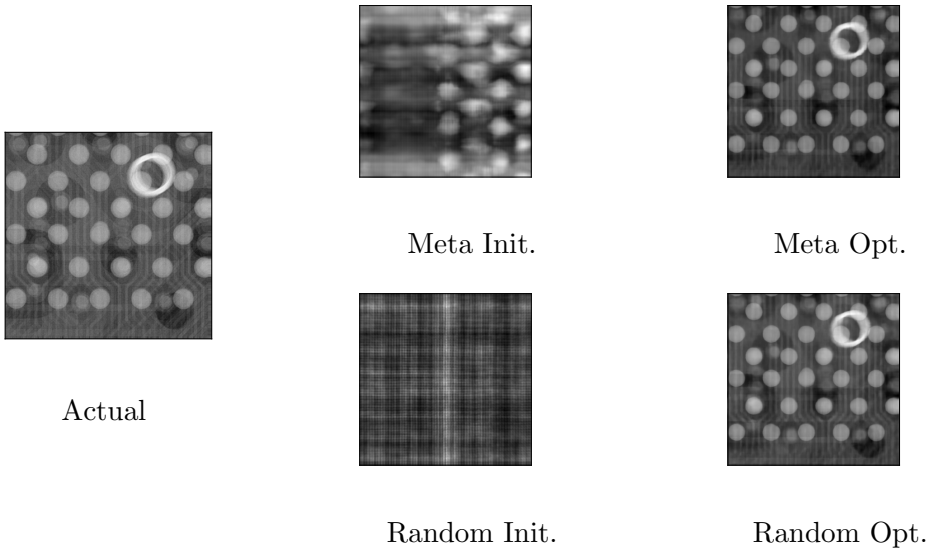

FIGURE 5.3: Visualization of the projections generated by different initialization schemes for Scene 1.

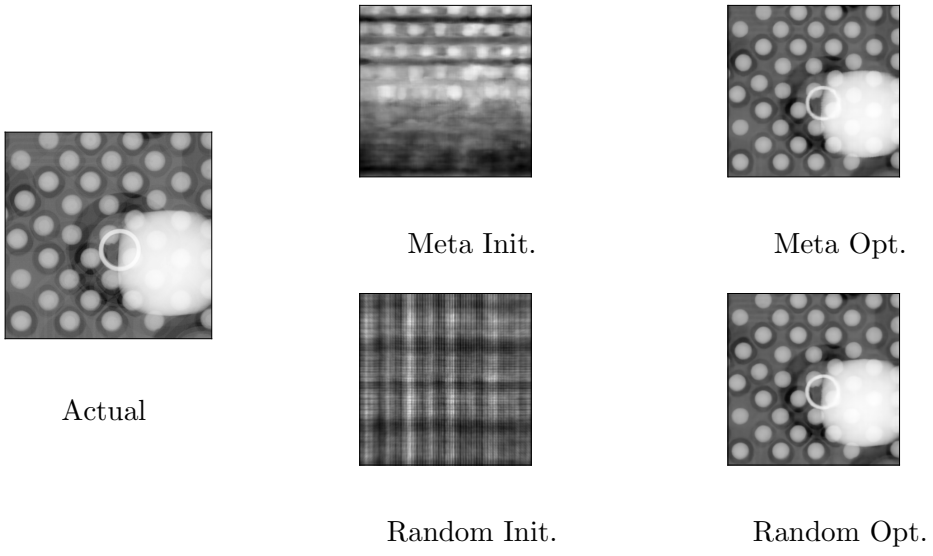

FIGURE 5.4: Visualization of the projections generated by different initialization schemes for Scene 7.

5.4.1.2 Reduced Optimization Steps

The comparison between optimizing an XRM-NeRF model, initialized with random weights, and an iXRM-NeRF model, initialized with meta-learned weights, across various 3D XRM scenes is illustrated in Figures 5.5 and 5.6. A total of four experiment runs were conducted, following the setup detailed in Section 5.3, to

account for variability in both the XRM-NeRF and iXRM-NeRF models. Figure 5.5 compares the number of optimization steps taken for the XRM-NeRF model versus the iXRM-NeRF model, while Figure 5.6 compares the PSNR of images reconstructed by each model. The results show that, in most cases, the meta-learning approach reduces the number of optimization steps, with the median number of steps being lower in the meta-learning model compared to the baseline. However, Scenes 03 and 04 did not follow this trend, as meta-learning did not always result in fewer steps. For Scene 03, the distribution of optimization steps for both models was similar, indicating that meta-learning neither improved nor hindered the process. Importantly, the meta-learned model did not negatively affect the PSNR of the reconstructed images, indicating that it can reduce computation time without sacrificing reconstruction quality. While these results are promising, they are not conclusive due to the limited dataset (only 7 scenes) and the contradictory findings in the exceptions. To strengthen these conclusions and confirm the trend across a broader range of data, further experiments with more scenes are needed.

$$MSE_{delta}(I) = \frac{MSE(I;\theta_{xrm}) - MSE(I;\theta_{ixrm})}{MSE(I;\theta_{xrm})} \tag{5.10}$$

$$Step_{delta}(I) = \frac{NumStep(I;\theta_{xrm}) - NumStep(I;\theta_{ixrm})}{NumStep(I;\theta_{xrm})} \tag{5.11}$$

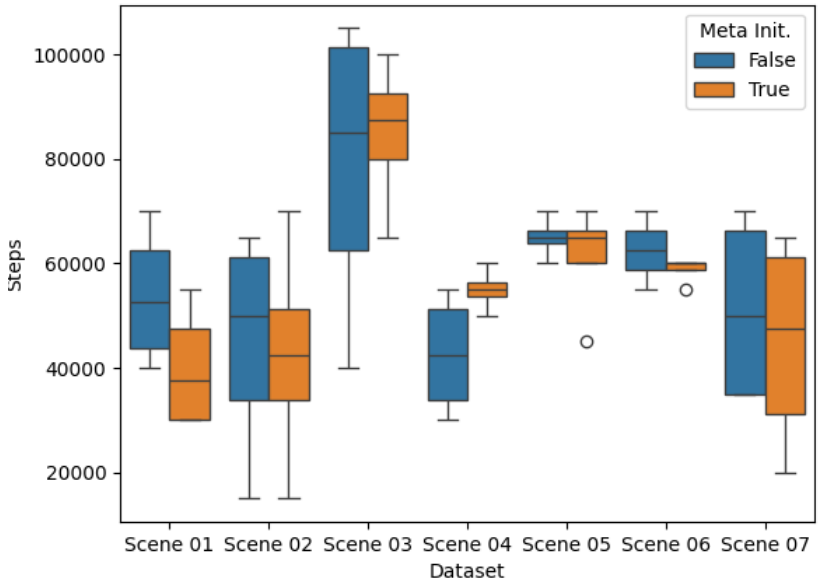

FIGURE 5.5: Comparison of the number of steps taken to optimize an XRM-NeRF model, which is randomly initialized, versus an iXRM-NeRF model, which is initialized by meta-learned weight parameters.

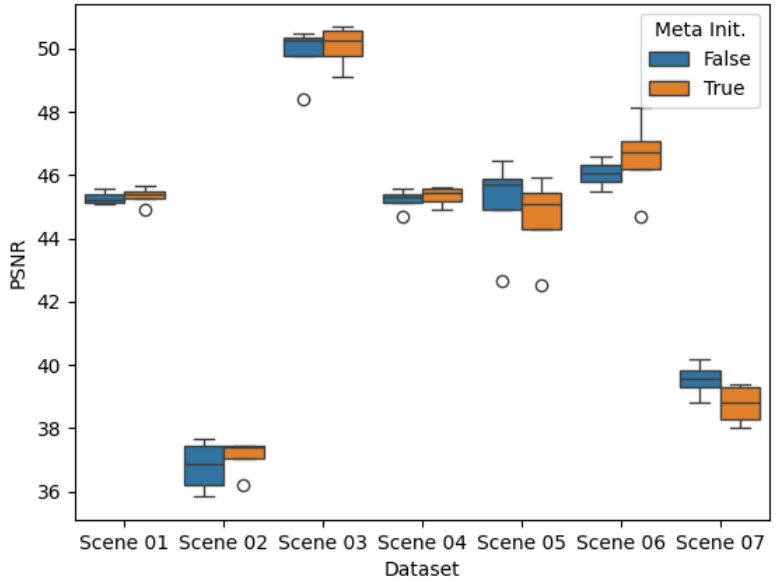

FIGURE 5.6: Comparison of the PSNR of the images reconstructed by a trained XRM-NeRF model, which is randomly initialized, versus a trained iXRM-NeRF model, which is initialized by meta-learned weight parameters.

5.4.1.3 Increased Convergence

The convergence graphs for the various XRM Scenes are displayed in Figure 5.7 which compare the number of optimization steps against the average mean square error on the test dataset. The blue lines represent the optimization graph for the iXRM-NeRF model, while the orange lines represent the optimization graph for the XRM-NeRF model. Scenes 1, 3, 4, and 7 exhibit accelerated convergence compared to the optimization process of the XRM-NeRF model. While scenes 2, 5, and 6 begin with a lower test score upon initialization with meta-learned weights, they do not demonstrate characteristics of accelerated convergence.

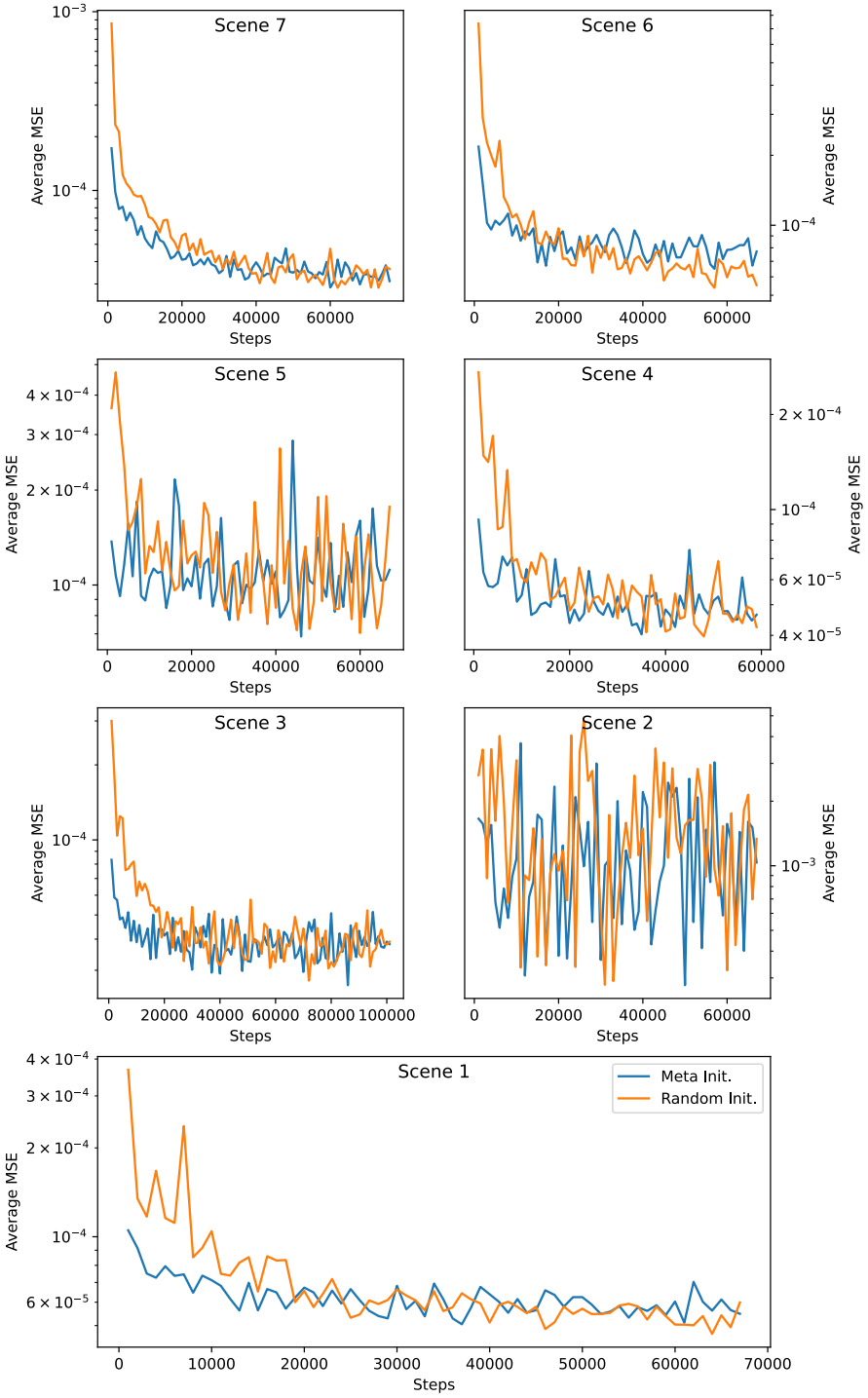

FIGURE 5.7: Number of iteration steps vs Average MSE plot comparing the convergence between iXRM-NeRF model initialized with meta-learned weight parameters and XRM-NeRF model initialized with random weight parameters.

In particular, the scene 6 test score stagnates for the iXRM-NeRF model while the test score for the XRM-NeRF model continues to decrease. According to the convergence graphs, the iXRM-NeRF model initialized with meta-learned weights showcases accelerated convergence in some XRM scenes and highlights the potential of meta-learned weights to expedite the optimization process.

5.4.1.4 Ablation Study

The initial phases for the experiments in this chapter were conducted using the weight decay hyperparameter of 0.001 as prescribed in chapter 4. However, the training process for learning the initial weight parameters for iXRM-NeRF models showed significant instabilities. Analysis of the mean and variance of activations in the last layer of ResBlock showed high variance values, which have been hypothesized to impede the learning process of a neural network [81]. Given that the magnitude of neural network weights directly impacts the variance of the activations, an ablation study was conducted to identify the optimal value for the weight decay hyperparameter. The weight decay hyperparameter was determined through a grid search process.

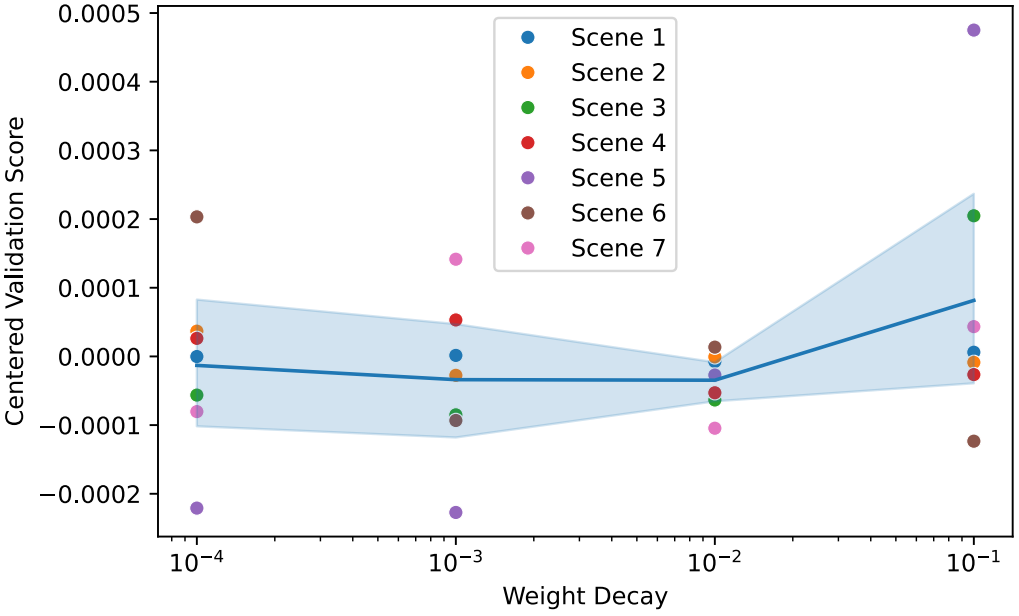

FIGURE 5.8: Weight Decay vs Centered Validation Score plot.

The **weight decay** hyperparameter was assessed on values 0.1, 0.01, 0.001, and 0.0001; For each weight decay hyperparameter value, a set of initial weight parameters for the iXRM-NeRF models was learned through the meta-learning process described in section 5.3. The centered validation scores for the iXRM-NeRF models are reported in Figure 5.8. The centered validation score was derived by subtracting the average of validation mean square errors across various iXRM-NeRF models trained with distinct weight decay hyperparameters from the validation mean square error associated with a particular weight decay hyperparameter for the respective dataset.

Figure 5.8 illustrates the significant influence of the validation dataset on determining the optimal set of weight decay hyperparameters, the optimal weight decay varies across different datasets. The value of 0.01 for the weight decay hyperparameter was selected as the optimal weight decay value as it demonstrated the lowest arithmetic average and variance of the centered validation score across the range of weight decay hyperparameters.

5.4.2 iXRM-NeRF Cross-Sectional View

Figure 5.9 visually illustrates the relationship between the projection image frontal view that is generated by the volumetric rendering process and the cross-section view image that is generated by iXRM-NeRF synthesis using orthogonal projection. The cross-sectional views of 3D XRM scenes are displayed in figures 5.10, 5.11, 5.12, 5.13, and 5.14. Each figure showcases three images. The first image presents projection images of the semiconductor product sample. The red line in the projection image indicates the area from which the full cross-sectional view is generated. The area between the green lines in the full cross-sectional view indicates the focus region for the zoomed cross-sectional view.

Figures 5.10, 5.11, and 5.12 serve as a reference for cross-sectional views of semiconductor product samples **without a defect**. In each of these figures, distinct boundaries separate each bump, depicted by the white cylindrical blobs in the middle of the cross-sectional views.

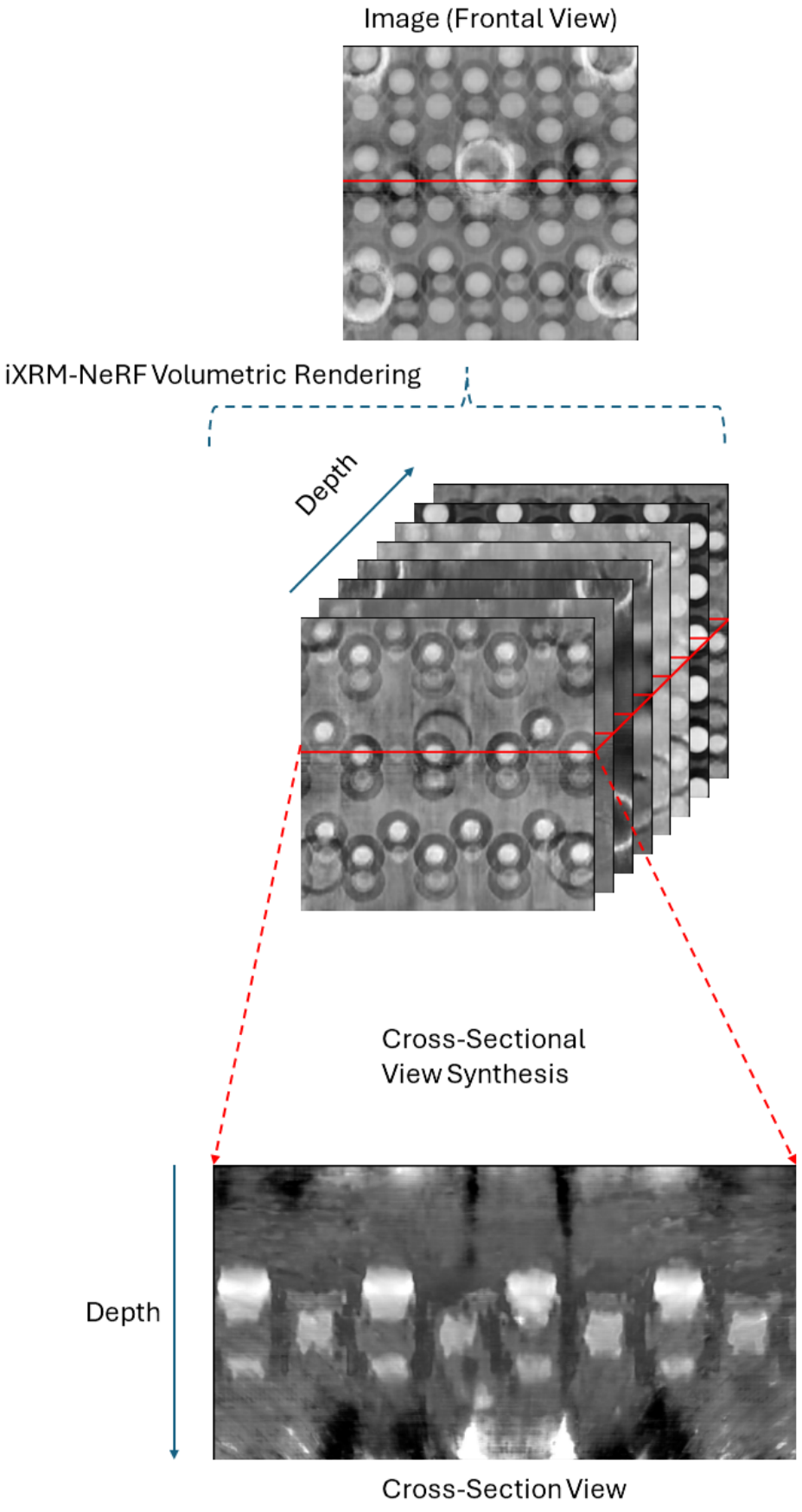

FIGURE 5.9: An illustration of the synthesis process for the image and cross-sectional views.

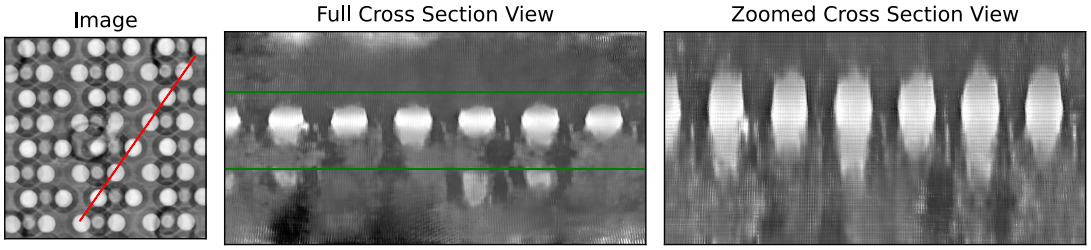

FIGURE 5.10: The cross-sectional view of a non-defective array of bumps.

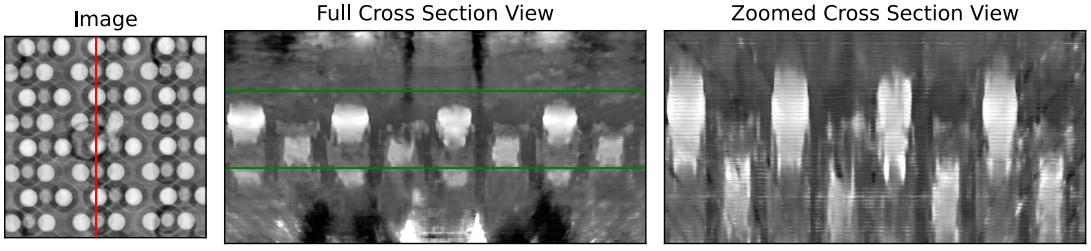

FIGURE 5.11: The cross-sectional view of a non-defective array of bumps.

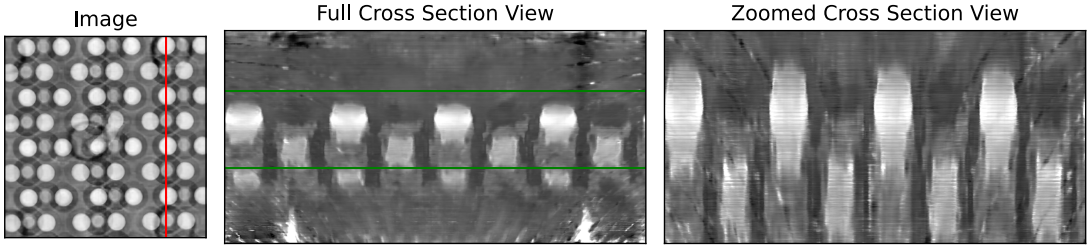

FIGURE 5.12: The cross-sectional view of a non-defective array of bumps.

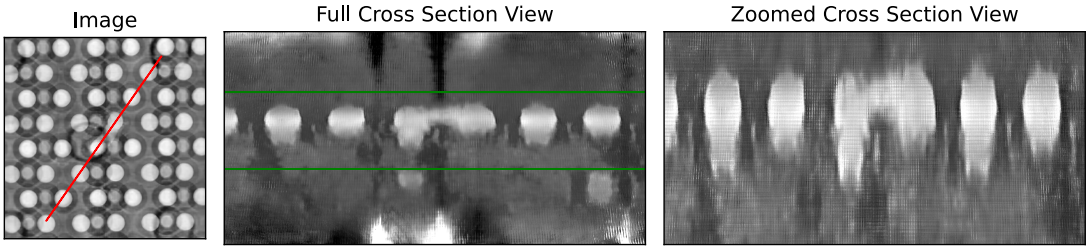

FIGURE 5.13: The cross-sectional view of a solder melt defect that is causing a short circuit between the 3rd bump and 4th bump from the right.

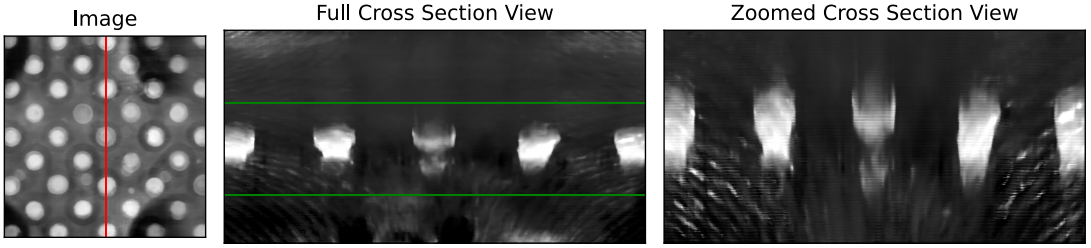

FIGURE 5.14: The cross-sectional view of a solder melt defect that is causing an open circuit for the bump in the middle of the cross-section view.

Figure 5.13 and 5.14 demonstrate the capability of iXRM-NeRF to generate cross-sectional views for visualizing defects. In Figure 5.13, the defective cross-sectional view, materials can be observed connecting bumps 3 and 4, resulting in a short circuit path that indicates a defect in the semiconductor sample. In Figure 5.14, the material for the middle bump can be observed to be melted, visualized as a lack of consistency in the color of the bump, and there is a gap between the top and bottom of the bump, causing an open circuit.

5.4.3 Survey Results

A survey is performed to understand the efficacy of the cross-sectional views for failure analysis product engineers to identify defects in a semiconductor product

sample. Each survey participant is presented with 30 sets of images, of which 3 are cross-sectional views of defective samples and 27 are cross-sectional views of non-defective samples. Sixteen product engineers participated in the survey, including seven domain experts from our industrial partner. All survey participant received fundamental domain-specific information outlining the cross-sectional scene appearance of a semiconductor sample and potential defect types. They are then asked to classify whether each image contained a defect. More information about the survey can be found in Appendix A.

A confusion matrix [82] for the survey result is presented in Table 5.1. The confusion matrix of the survey shows that the sensitivity [83] of the cross-sectional view survey results is relatively high at 93.75%, indicating that survey participants can effectively classify defective samples based on the provided cross-sectional views. The lower specificity [83] of 65.19% indicates participants are less capable of identifying non-defective samples based on the cross-sectional views. Feedback gathered from survey participants highlights the lack of domain knowledge or familiarity with 3D XRM cross-sectional view samples. This is consistent with the confusion matrix reported in Table 5.2, focusing solely on the survey responses from 7 domain experts, familiar with 3D XRM failure analysis cross-sectional view samples, out of the total 16 participants. The confusion matrix with survey results from domain experts shows significantly higher sensitivity at 100% and specificity at 84.13%. **While the results of this survey provide valuable insights, it is important to acknowledge that the relatively small number of participants and evaluation images may limit the generalizability of the findings. Additionally, the set of images used in this study may not encompass all possible variations of defects and non-defective samples that could arise in semiconductor failure analysis. These factors suggest that a larger-scale study, involving a broader range of participants and a more comprehensive image dataset, would be necessary to validate and expand upon the current findings. Future research should aim to include a more diverse pool of participants and a wider variety of images to further assess the effectiveness of cross-sectional views in defect identification.**

	Actual		
Survey		Defect	No Defect
	Defect	93.75% (Sensitivity)	34.81% (Specificity)
	No Defect	6.25%	65.19%

TABLE 5.1: Confusion matrix for survey results asking participants to classify whether each image has a defect.

	Actual		
Survey		Defect	No Defect
	Defect	100% (Sensitivity)	15.87% (Specificity)
	No Defect	0%	84.13%

TABLE 5.2: Confusion matrix for survey results asking domain experts to classify whether each image has a defect.

5.5 Summary

This chapter presents the iXRM-NeRF approach, designed to enhance the convergence of XRM-NeRF, and introduces a technique for producing cross-sectional views of 3D XRM samples. iXRM-NeRF proposes to learn an initial set of weight parameters using meta-learning to accelerate the iXRM-NeRF optimization process for a 3D XRM scene. Additionally, iXRM-NeRF proposes to modify the integrated positional encoding of XRM-NeRF to perform orthogonal projection instead of perspective projection to ensure isomorphism between the predicted 3D XRM scene and the generated cross-sectional views.

The experiments conducted in this chapter showcased the enhanced convergence of iXRM-NeRF, potentially expediting the 3D XRM reconstruction process by 17.81%. Additionally, the qualitative analysis of the cross-sectional views demonstrated the capability of iXRM-NeRF to meet practical demands in industrial domains such as semiconductor failure analysis, by producing visuals that can indicate physical defects on semiconductor chips.

While the findings of this survey offer valuable insights, the limited number of participants and evaluation images suggest that the results

may not be fully generalizable. A larger-scale study with a more diverse participant pool and a broader range of images is needed to validate and extend these findings. Future research should focus on expanding the dataset and participant diversity to further evaluate the effectiveness of cross-sectional views in defect identification.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

Through the exploration of three novel techniques, this thesis has made significant strides in overcoming key challenges within the realm of XRM imaging for semiconductor defect analysis. The integration of these innovative methodologies for 3D X-ray Microscopy (XRM) imaging has demonstrated promising avenues for enhancing reconstruction efficiency.

SM-DCS (Sparse Matrix Deep Compressive Sensing) is proposed to address the incompatibility between Deep Compressive Sensing and existing 3D XRM hardware scanners. It uses a novel two-phase approach to streamline the measurement process within a 3D XRM scanner. In the first phase, the system reduces the number of pixels scanned during each projection. In the second phase, the reduced measurements are then reformulated into a compressive sensing problem, effectively leveraging the capabilities of the Deep Compressive Sensing approach for reconstruction. This innovative approach optimizes the efficiency of compressive sensing while ensuring compatibility with existing 3D XRM scanners, offering a promising solution to the challenges inherent in XRM imaging. By demonstrating comparable reconstruction results to Deep Compressive Sensing while utilizing only 15% of image pixels, SM-DCS showcases its efficacy and presents a practical solution to streamline XRM imaging processes.

The proposed XRM-NeRF (XRM-Neural Radiance Field) introduces a novel volumetric rendering framework tailored to the physics of XRM scanners. XRM-NeRF establishes its volumetric rendering framework based on the fundamental principles of X-ray attenuation, ensuring precise prediction of 3D geometrical structures from XRM projection images. In addition, XRM-NeRF uses a fine-tuning parameter to mitigate systematic noise that can be present in XRM scanners during the XRM imaging process. XRM-NeRF is demonstrated to be able to achieve good reconstruction quality with a significantly reduced number of projection images. While XRM-NeRF introduces extra computational time for optimizing the model as a trade-off for reduced scan time, the extra computational time can be amortized by performing multiple XRM-NeRF optimization in parallel using the high-availability of GPU computing in cloud data centers.

The proposed iXRM-NeRF harnesses the capability of meta-learning to expedite the 3D XRM reconstruction process and also introduces an innovative cross-sectional view generation process. Meta-learning is used to learn an initial set of weight parameters for a class of 3D XRM scenes to enhance convergence and quality across most 3D XRM scenes. The orthogonal projection-inspired based positional encoding used during the cross-sectional view generation process also demonstrated the ability of iXRM-NeRF to enable faithful visualization of the internal structure of a semiconductor sample.

The novel approaches proposed in this thesis hence illustrate the possible avenues to enhance the 3D XRM imaging process and accelerate the semiconductor defect analysis process. The expedited defect analysis process has the potential to accelerate the development lifecycle of semiconductor products, by providing semiconductor design engineers with timely feedback, resulting in cost reduction and enhanced innovation.

6.2 Future Directions

There are a number of areas that are possible to be extended from the work presented in this thesis.

A notable gap in iXRM-NeRF is the high computational cost of iXRM-NeRF models to faithfully represent high-fidelity 3D XRM scenes. While NeRF-related

models are touted for their potential to represent infinite-resolution 3D scenes with a fixed-size parameter model, they usually require optimization of large multilayer-perceptron (MLP) [84] models. Recent studies [85–87] have explored a middle ground between voxel-based representation and NeRF representation using learnable hash-based encoding as a replacement for the positional encoding used in NeRF models. The advantage of utilizing hash tables rather than fixed-size NeRF models enables the neural network to learn to compress scene details from multiple locations in a 3D scene to a much larger hash memory. The exploration of such hash-based encoding can be extended to iXRM-NeRF to improve the quality of the reconstructions and reduce the computational requirements for the optimization process due to smaller model sizes.

Another possible direction for future work is the exploration of automatic localization of physical defects in semiconductor 3D XRM samples. A recent study [88] explored the utilization of NeRF for localization based on RGB images, demonstrating significant potential. This avenue of research holds promise in alleviating the responsibility of manual defect localization by engineers and enhancing the efficiency of the failure analysis process.

Appendix A

Survey for iXRM-NeRF cross-sectional view

This section presents an overview of the survey questions designed to assess how effective iXRM-NeRF cross-sectional views are in detecting defects.

The survey questions are shown in figure A.1. Each participant is provided with basic domain knowledge, such as the expected geometrical structure of each cross-sectional view and what constitutes a defect. They are subsequently tasked to identify whether each image in Figure A.2 is a defect. Cross-section views of defective samples in Figure A.2 are highlighted in red.

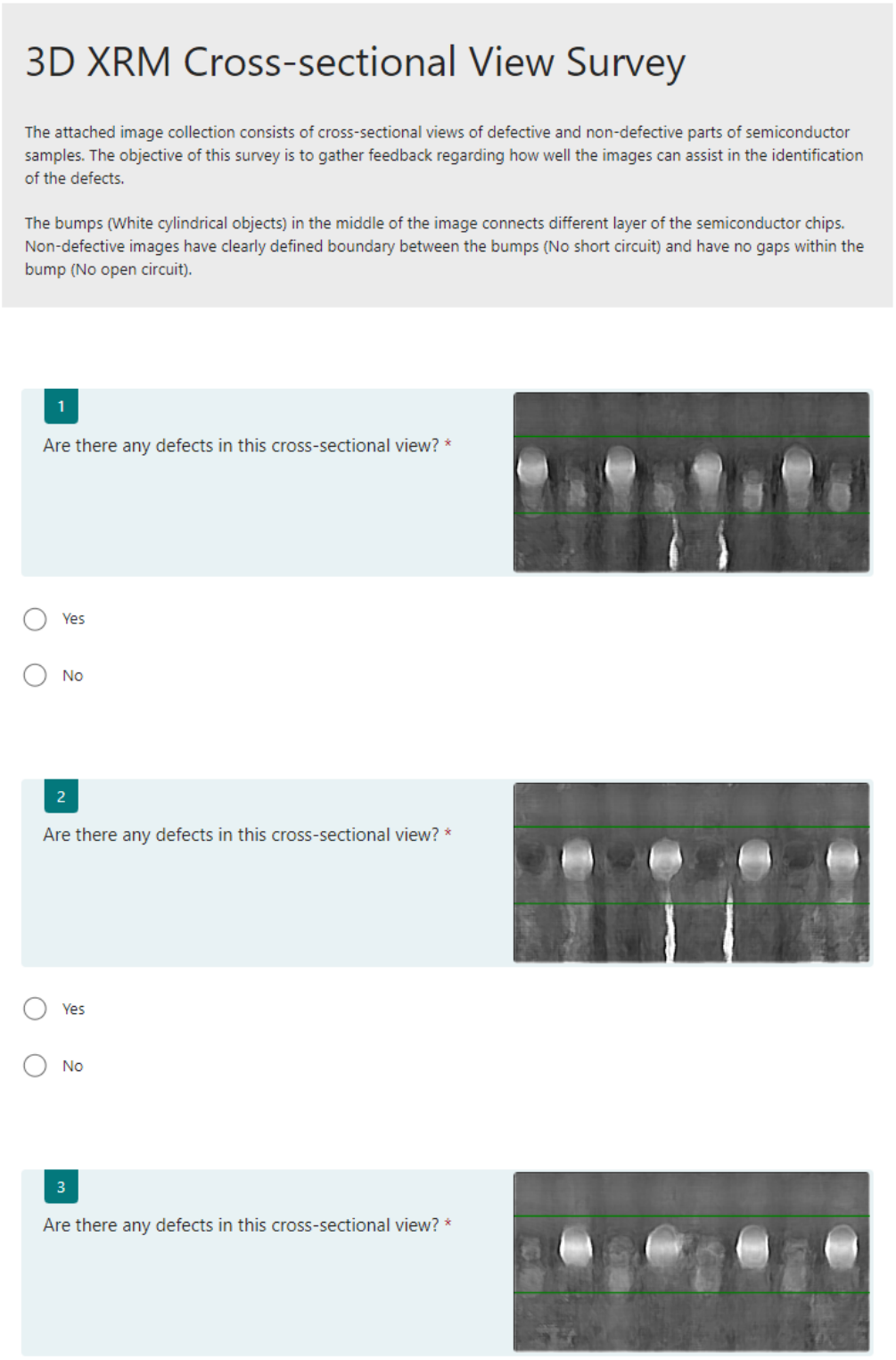

FIGURE A.1: Screenshot of survey questions sent to participants.

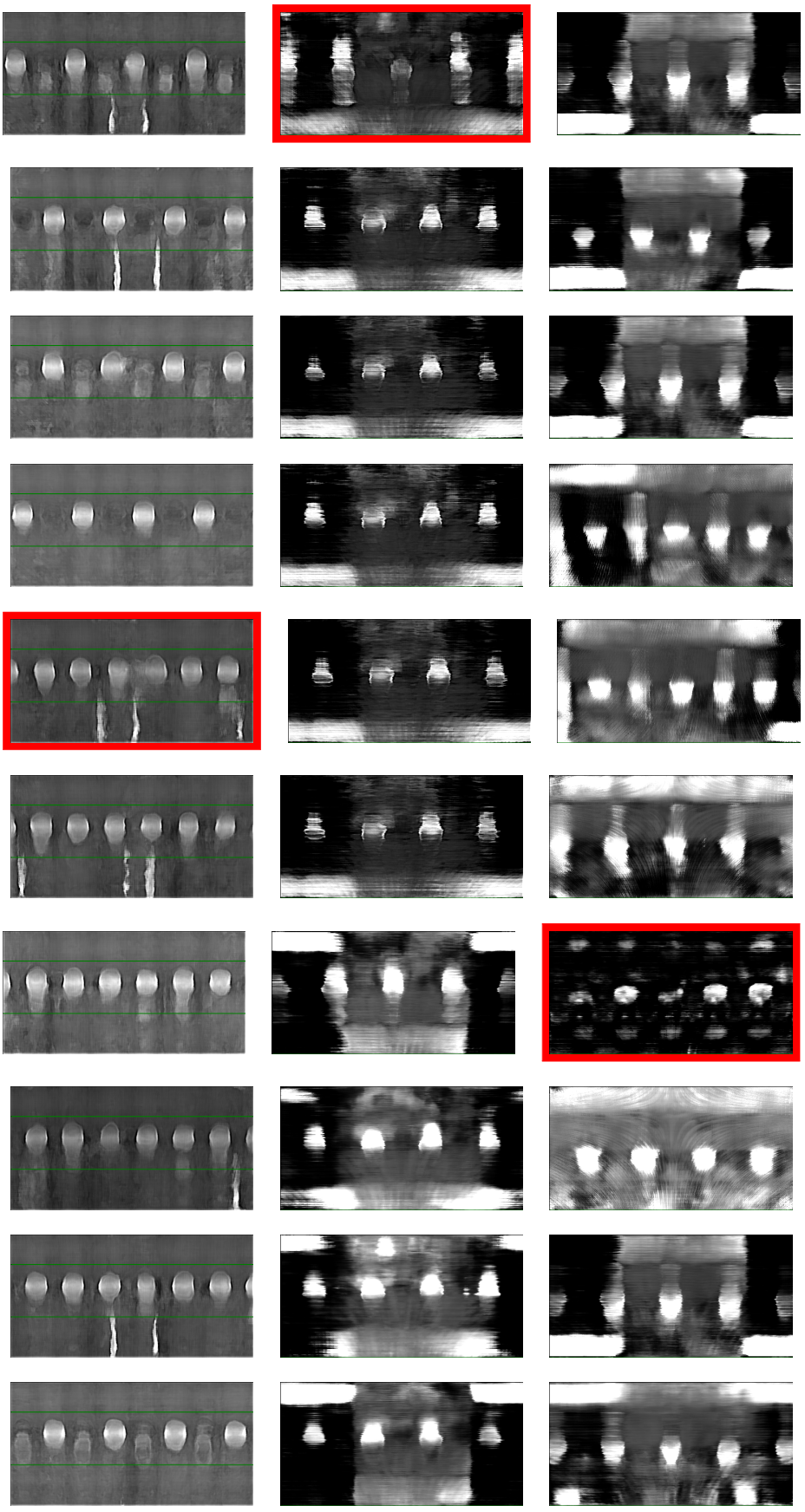

FIGURE A.2: Gallery of the cross-section views of defective and non-defective sample locations sent to survey participants for classification.

List of Publication[1]

Conference Proceeding

- **Tan Ying Hao**, Nicholas Vun, and Bu-Sung Lee, "Sparse image measurement using deep compressed sensing to accelerate image acquisition in 3D XRM" in *International Conference on Quality Control by Artificial Vision. Vol. 12749. SPIE, 2023.*.

[1]The superscript * indicates joint first authors

Bibliography

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. xvi, 5, 16, 17, 20, 41, 52, 53

[2] MultiMedia LLC. Zeiss xradia versa user's guide, 2016. URL `https://www.physics.purdue.edu/xrm/new-users/training/Xradia_Versa_Users_Guide_v11x_510.pdf`. 2

[3] Arno P. Merkle and Jeff Gelb. The ascent of 3d x-ray microscopy in the laboratory. *Microscopy Today*, 21(2):10–15, 2013. doi: 10.1017/S1551929513000060. 2

[4] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017. 4, 15

[5] Yan Wu, Mihaela Rosca, and Timothy Lillicrap. Deep compressed sensing. In *International Conference on Machine Learning*, pages 6850–6860. PMLR, 2019. 4, 5, 15, 16, 25, 29

[6] David J Starling, Ian Storer, and Gregory A Howland. Compressive sensing spectroscopy with a single pixel camera. *Applied optics*, 55(19):5198–5202, 2016. 4

[7] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng Nerf. Representing scenes as neural radiance fields for view synthesis., 2021, 65. *DOI: https://doi. org/10.1145/3503250*, pages 99–106. 5, 16, 17, 18, 49

[8] JH Hubbell. Review of photon interaction cross section data in the medical and biological context. *Physics in Medicine & Biology*, 44(1):R1, 1999. 5, 42, 45

[9] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. 11

[10] KABIR DUBEY. The fourier uncertainty principles. 2021. 13

[11] Gabriel Peyre. Best basis compressed sensing. *IEEE Transactions on Signal Processing*, 58(5):2613–2622, 2010. 13

[12] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006. 13

[13] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007. 14

[14] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. Technical report, Computer Science Department, Technion, 2008.

[15] Paul Irofti and Bogdan Dumitrescu. Pairwise approximate k-svd. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3677–3681. IEEE, 2019. 13

[16] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009. 14

[17] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006. 14

[18] Shaobing Chen and David Donoho. Basis pursuit. In *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44. IEEE, 1994. 14

[19] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014. 14, 15

[20] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010. 14, 15

[21] Kristian Bredies and Dirk A Lorenz. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14(5-6):813–837, 2008. 15

[22] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1): 183–202, 2009. 15

[23] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady an ussr*, volume 269, pages 543–547, 1983. 15

[24] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Admm-csnet: A deep learning approach for image compressive sensing. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):521–538, 2018. 15

[25] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated mri data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

[26] Puyang Wang, Eric Z Chen, Terrence Chen, Vishal M Patel, and Shanhui Sun. Pyramid convolutional rnn for mri reconstruction. *arXiv preprint arXiv:1912.00543*, 2(3), 2019. 15

[27] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016. 16, 18

[28] Danilo Jimenez Rezende, SM Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. *Advances in neural information processing systems*, 29, 2016.

[29] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1955–1964, 2018.

[30] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 16

[31] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 16

[32] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 16

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 16

[34] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 4857–4866, 2020.

[35] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 17, 62, 63

[36] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 16, 17

[37] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 16

[38] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–662, 2018. 17

[39] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. *Advances in neural information processing systems*, 31, 2018. 17

[40] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 412–420. IEEE, 2017. 17

[41] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 17

[42] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 17, 18

[43] Fatih Calakli and Gabriel Taubin. Ssd: Smooth signed distance surface reconstruction. In *Computer Graphics Forum*, volume 30, pages 1993–2002. Wiley Online Library, 2011. 17

[44] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 17

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 17

[46] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[47] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 17, 18

[48] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33: 7537–7547, 2020. 19

[49] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, 1996. doi: 10.1109/34.544079. 20

[50] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *CVPR*, 2021. 22, 23, 68

[51] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021. 22

[52] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 22

[53] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018. 22, 23, 68

[54] Shun ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4):185–196, 1993. ISSN 0925-2312. doi: https://doi.org/10.1016/0925-2312(93)90006-O. URL `https://www.sciencedirect.com/science/article/pii/092523129390006O`. 22

[55] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 22

[56] Amir Adler, David Boublil, Michael Elad, and Michael Zibulevsky. A deep learning approach to block-based compressed sensing of images. *arXiv preprint arXiv:1606.01519*, 2016. 25

[57] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deepbinarymask: Learning a binary mask for video compressive sensing. *Digital Signal Processing*, 96:102591, 2020. 25

[58] Arno P Merkle and Jeff Gelb. The Ascent of 3D X-ray Microscopy in the Laboratory. *Microscopy Today*, 21(2):10–15, 03 2013. ISSN 1551-9295. doi: 10.1017/S1551929513000060. URL `https://doi.org/10.1017/S1551929513000060`. 25

[59] Ying Hao Tan, Nicholas Vun, and Bu-Sung Lee. Sparse image measurement using deep compressed sensing to accelerate image acquisition in 3d xrm. In *Sixteenth International Conference on Quality Control by Artificial Vision*, volume 12749, pages 155–161. SPIE, 2023. 29

[60] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 30

[61] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 32

[62] Zeiss xradia versa user's guide. `ZeissXradiaVersausersguide.ZEISSXradiaVersaUsersGuide.(n.d.).https://www.physics.purdue.edu/xrm/new-users/training/Xradia_Versa_Users_Guide_v11x_510.pdf`. [Accessed 10-05-2024]. 44, 46

[63] Sébastien Briot, Wisama Khalil, Sébastien Briot, and Wisama Khalil. Homogeneous transformation matrix. *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements*, pages 19–32, 2015. 47

[64] R Stuart Ferguson. *Practical algorithms for 3D computer graphics*. CRC Press, 2013. 47

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016. 52, 53

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 53

[67] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 54

[68] Johan Bjorck, Kilian Q Weinberger, and Carla Gomes. Understanding decoupled and early weight decay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6777–6785, 2021. 54

[69] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991. 54

[70] Jari Korhonen and Junyong You. Peak signal-to-noise ratio revisited: Is simple beautiful? In *2012 Fourth International Workshop on Quality of Multimedia Experience*, pages 37–38. IEEE, 2012. 54

[71] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2011. 54

[72] Richard Dosselmann and Xue Dong Yang. A comprehensive assessment of the structural similarity index. *Signal, Image and Video Processing*, 5:81–91, 2011. 54

[73] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 58

[74] Kevin E O'Grady. Measures of explained variance: Cautions and limitations. *Psychological Bulletin*, 92(3):766, 1982. 61

[75] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 63

[76] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 63

[77] Mike Huisman, Jan N Van Rijn, and Aske Plaat. A survey of deep meta-learning. *Artificial Intelligence Review*, 54(6):4483–4541, 2021. 67

[78] Yann N Dauphin and Samuel Schoenholz. Metainit: Initializing learning by learning to initialize. *Advances in Neural Information Processing Systems*, 32, 2019. 67

[79] Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 152–162, 1997. 74

[80] Thomas Oppert, Thorsten Teutsch, Elke Zakel, and David Tovar. A bumping process for 12" wafers. In *Twenty Fourth IEEE/CPMT International Electronics Manufacturing Technology Symposium (Cat. No. 99CH36330)*, pages 328–333. IEEE, 1999. 75

[81] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/5d44ee6f2c3f71b73125876103c8f6c4-Paper.pdf`. 81

[82] Olivier Caelen. A bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, 81(3):429–450, 2017. 86

[83] Abdul Ghaaliq Lalkhen and Anthony McCluskey. Clinical tests: sensitivity and specificity. *Continuing education in anaesthesia, critical care & pain*, 8 (6):221–223, 2008. 86

[84] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009. 91

[85] Bo Peng, Jun Hu, Jingtao Zhou, Xuan Gao, and Juyong Zhang. Intrinsicngp: Intrinsic coordinate based hash encoding for human nerf. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 91

[86] Xiufeng Xie, Riccardo Gherardi, Zhihong Pan, and Stephen Huang. Hollownerf: Pruning hashgrid-based nerfs with trainable collision mitigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3480–3490, 2023.

[87] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 91

[88] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022. 91