POLITECNICO DI TORINO

MASTER's Degree in Biomedical Engineering

Politecnico di Torino

MASTER's Degree Thesis

Cross Attentive PET Image Reconstruction Methods

The Role of the Cross-Attention in the LPD Reconstruction Algorithm

Supervisors

Prof. Filippo MOLINARI
Prof. Massimo SALVI
Prof. Hamidreza Rashidy KANAN

Candidate

Simone BONINO

March 2025

This page is intentionally left blank.

Abstract

Positron Emission Tomography (PET) is a key diagnostic tool in oncology, cardiology and neurology. However, the inherent noise and sparsity of the acquisition process pose significant challenges for image reconstruction. State-of-the-art reconstruction methods often produce low-resolution images and struggle to preserve small details. This thesis studies and develops new reconstruction methods based on the Learned Primal-Dual (LPD) reconstruction and enhanced by the Cross-Attention mechanism. A new synthetic generator capable of producing a wide variety of shapes was implemented, along with a new loss function, leading to improvements in both metrics and generalisation power. Using these new elements, four different LPD architectures incorporating Cross-Attention were tested, achieving comparable performance to previous implementations. Although the Cross-Attention mechanism did not significantly improve the LPD reconstruction algorithm, the results suggest its potential in effectively integrating different information and are promising for future applications.

Keywords
Positron Emission Tomography, Deep Learning, Reconstruction, Generalisation, Learned Primal-Dual Reconstruction, Vision Transformer, Cross-Attention

This page is intentionally left blank.

Acknowledgments

First of all, I would like to express my genuine gratitude to Massimiliano Colarieti Tosti and Hamidreza Rashidy Kanan whose invaluable guidance and constant feedback have been decisive in the success of my work. Their unwavering faith in my abilities has inspired me to overcome each challenge with renewed determination and enthusiasm.

I would also like to extend my sincere thanks to Massimo Salvi and Filippo Molinari for agreeing to remotely oversee my thesis work on behalf of the Polytechnic of Turin.

A heartfelt acknowledgement also goes to KTH Royal Institute of Technology and Polytechnic of Turin for allowing me to participate in a double degree programme. Spending half my master's degree at the Polytechnic of Turin and half at KTH was extremely valuable, broadened my horizons and contributed immensely to my growth as a future engineer and as an individual.

On a personal level, I am deeply grateful to my family. To my parents, Patrizia and Bruno, thank you for your unconditional love, patience and constant support. Finally, to my friends and all those who have played a role in this academic journey, know that your contributions, big or small, have left an indelible mark on my heart.

Simone Bonino
19-01-2025, Stockholm

This page is intentionally left blank.

Table of Contents

List of Tables

This page is intentionally left blank.

List of Figures

Acronyms

AI Artificial Intelligence.

AMP Automatic Mixed Precision.

CAB Cross-Attention Block.

CNN Convolutional Neural Network.

CT Computed Tomography.

DNN Deep Neural Network.

EM Expectation Maximisation.

FBP Filtered Back Projection.

LOR Line of Response.

LPD Learned Primal-Dual.

MAE Mean Absolute Error.

ML Maximum Likelihood.

MLEM Maximum Likelihood Expectation Maximisation.

MLP Multi Layer Perceptron.

MRI Magnetic Resonance Imaging.

MSE Mean Squared Error.

NLP Natural Language Processing.

NM Nuclear Medicine.

OSEM Ordered Subset Expectation Maximisation.

PET Positron Emission Tomography.

PMT PhotoMultiplier Tube.

PSNR Peak Signal-to-Noise Ratio.

SPECT Single-Photon Emission Computed Tomography.

SSIM Structural Similarity Index Measure.

ViT Vision Transformer.

This page is intentionally left blank.

1 Introduction

Detecting early-stage cancers or accurately determining the extent of cardiovascular disease, with the precision required to plan targeted and effective treatments, relies on the use of Positron Emission Tomography (PET), a crucial modality in medical diagnostics, particularly in oncology, cardiology and neurology. As reported by Eurostat [1], the number of PET scanners and scans has increased across the European Union between 2012 and 2022, although without reaching the volumes of other more established imaging techniques, such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI).

Indeed, despite its clinical importance, PET imaging faces inherent challenges that prevent it from becoming widely available. Firstly, the imaging process uses gamma rays emitted by radioactive tracers introduced into the body, which are difficult to produce and pose a danger to the patient's health. Secondly, the acquired data are often noisy and sparse due to the strong presence of noise and the low number of emitted rays, making the image reconstruction problem particularly complex. It is precisely this second problem that this project is interested in.

Analytical reconstruction methods, such as the Filtered Back Projection (FBP) algorithm, and iterative methods, like the Maximum Likelihood Expectation Maximisation (MLEM) algorithm and its variants, have been widely used in PET imaging [2], but the images they produce are often still affected by noise and artefacts. In recent years, the advent of deep learning has opened up new possibilities and made it possible to go beyond numerical algorithms alone, with encouraging results [3].

One of these deep learning approaches is the Learned Primal-Dual (LPD) reconstruction algorithm, which iteratively employs Deep Neural Networks (DNNs) on both the data and image domains to refine the reconstruction [4]. This architecture was originally proposed for CT, but in 2021 was also adapted for PET by A. Guazzo and M. Colarieti-Tosti [5]. Recently, this algorithm was updated and extended by A. Adelöw, who introduced Vision Transformers into the architecture for the first time [6].

Further improving the LPD algorithm and achieving better reconstructions could have a major impact not only on diagnostic, where it is important for disease detection, staging and monitoring [7], [8], but also in drug discovery and development [9]. An enhanced image quality, with higher spatial resolution and reduced noise, could allow for more precise identification of small lesions, such as early-stage tumours, and also support the ongoing efforts to optimise the radiation dose a patient receives [10].

This project aims to introduce a new training strategy to increase the generalisation capabilities of deep learning models and to propose new variants of LPD architecture that employ Cross-Attention Blocks (CABs) to better integrate

information between different steps. CABs are particular Vision Transformers that allow one set of data to learn from another set of data, finding important connections between them.

In particular, the project's tasks are the following:

- implement a new synthetic image generator to produce diverse and realistic images for the training data set;
- define a new loss function;
- evaluate the effectiveness of the previous strategies in the training process;
- propose one or more modified LPD architectures that feature CABs to enhance information integration;
- assess the importance of CABs' presence in the proposed architectures.

2 Theoretical Background

This chapter provides a concise and comprehensive overview of the study's theory, outlining historical developments and key concepts. In particular, the first two sections present an overview of PET imaging and established approaches to the reconstruction problem, while the last three sections illustrate the Learned Primal-Dual (LPD) architecture and the relevant concepts for its improvement.

2.1 Nuclear medicine

Nuclear Medicine (NM) refers to a group of clinical practices involving the use of radioactive substances for diagnosis and treatment. Its conceptualisation dates back to the 1930s, but it was not until the 1990s, with the development of the first tomographic devices and commercial radiopharmaceuticals, that it gained recognition in the medical field [11].

The images produced by NM imaging are called functional, meaning that the focus is on the body's function rather than its anatomy. This makes Nuclear Medicine particularly interesting because it allows to study internal physiological processes in vivo.

The most important components of any NM imaging device are the radioactive tracers and the detectors. Radioactive tracers are made of a radioactive isotope and a biological molecule which targets a specific type of tissue or process. Radioactive isotopes are artificially created from stable atoms by unbalancing the ratio of protons to neutrons in the nucleus. This imbalance causes a series of nuclear decays, which follow the law in Equation 2.1.

$$N(t) = N_0 e^{-\lambda t} \tag{2.1}$$

where λ is the decay constant of the specific atom. Decay occurs according to several nuclear processes, but only those that emit γ rays can be used in medical imaging because only γ photons have enough energy to penetrate outside the body.

To capture these γ photons, special detectors, called gamma cameras, are needed. A gamma camera or Anger camera consists of a scintillator crystal, such as NaI(Tl), behind which one or more PhotoMultiplier Tubes (PMTs) are placed. When a region of scintillator crystal is hit by a γ radiation, it produces a faint light that is detected by the PMTs and converted into current. The current is then read and quantified by a computer. This information constitutes the "acquired data" that is used for image reconstruction.

The most established NM imaging techniques are Positron Emission Tomography (PET), which is addressed more in detail in the next section, and Single-Photon Emission Computed Tomography (SPECT). The latter differs from the

former mainly in the type of emission process used, which is based on single-photon emissions. This requires the use of different radioactive tracers and collimators in front of the gamma cameras [12].

2.1.1 Positron Emission Tomography

Positron Emission Tomography (PET) devices have a detector system composed of many small detector blocks, similar to small gamma cameras, distributed in one or multiple rings as shown in Figure 2.1. The more rings, the greater the volume that can be imaged simultaneously.

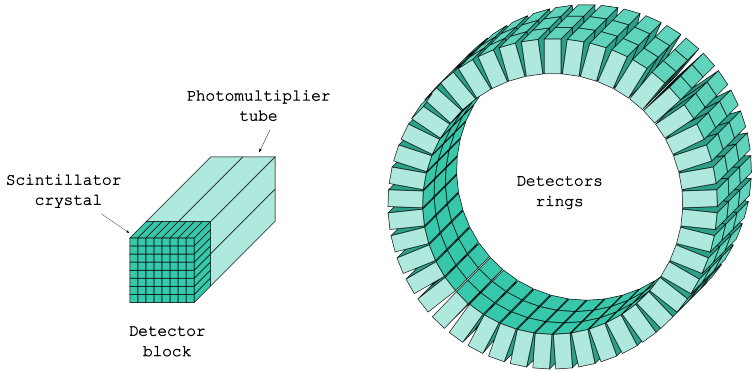

Figure 2.1. Geometry of a PET scanner
Source: Figure created by the author

PET uses radioactive tracers with short lifetimes (from a few minutes to some hours), which must be produced just before imaging. For this reason, a particle accelerator called a cyclotron must be no more than few hours away from a PET scanner room [13].

The decay process used in PET is the beta plus decay or positron emission which happens when a proton in the nucleus decays into a neutron causing the emission of a neutrino and a positron β^+. After travelling a small distance (emission range <1 mm), the positron β^+ annihilates with an electron e^- and a pair of γ photons are emitted in opposite directions, each with an energy of about 511 keV.

If two photons are detected within a given time interval, known as the coincidence window, they are assumed to be from the same emission event and their Line of Response (LOR), that is the line that connects the two affected detectors, is recorded. In contrast, photons arriving at the detectors at times separated by more than the duration of the coincidence window are ignored.

However, not all recorded LORs correspond to real emission lines. There are particular cases where photons are detected in the same emission window, but do not follow a straight line (scatter coincidence) or belong to two different emission events (random coincidence) [14]. This leads to the definition of an erroneous LOR, described by dotted lines in Figure 2.2. Even true coincidences are not without inaccuracies, as the two γ rays do not always have perfect collinearity and, as mentioned above, the emission only occurs after the positron has travelled a certain distance from the position of the tracer [15].

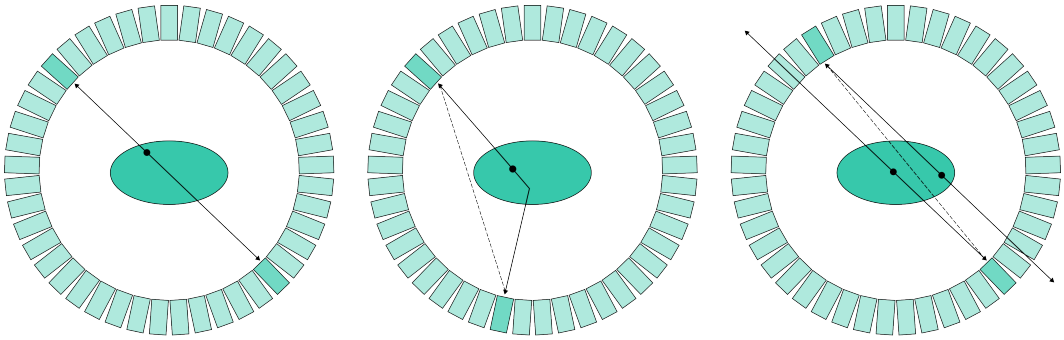

Figure 2.2. Different types of coincidences
Source: Figure created by the author

These issues, combined with the noise, the small number of emission events and the low spatial resolution of the crystals, render the reconstruction of the original image from the detected coincidences particularly complicated.

2.2 The Reconstruction Problem

To understand the reconstruction problem, it is necessary to model the emission process. Since this process is not deterministic, but stochastic, as presented in Section 2.1 of this Chapter, only the expected value of coincidence events detected along a given LOR can be obtained [16]. Equation 2.2 describes the expected value as the integral of the activity distribution along a LOR.

$$E\left[N_{l_{t,\theta}}\right] = \int_{l_{t,\theta}} f(x,y)dl \tag{2.2}$$

where $E[\cdot]$ stands for expected value, $N_{l_{t,\theta}}$ is the number of the events detected along a LOR, denoted with l, and f is the function that describes the activity distribution at point (x, y). The LOR l is parametrised with only two parameters: t, which represents the minimum distance from the centre of the field of view, and θ, which is the angle between the x axis and the segment identified by t [17]. Mathematically, the parameterisation obeys Equation 2.3.

$$l_{t,\theta} = \{(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta) : -\infty < s < \infty\} \tag{2.3}$$

where l is a generic line and s describes all the points belonging to the line.

Equation 2.2 defines the simplest form of the forward model, showing how the activity distribution is converted into the acquired data. The reconstruction task is to invert this relationship and find the activity distribution $\hat{f}$ from the acquired data that best approximates the true distribution f. However, due to the strong presence of noise, this problem is ill-posed and difficult to solve.

Acquired data can be organised in a sinogram. This is achieved by calculating the line integral along all the parallel LORs for a fixed angle θ (projection) and then repeating the process for each angle θ. An example of a sinogram, derived from a simple test object and plotted on a (t, θ) plane, is shown in Figure 2.3.

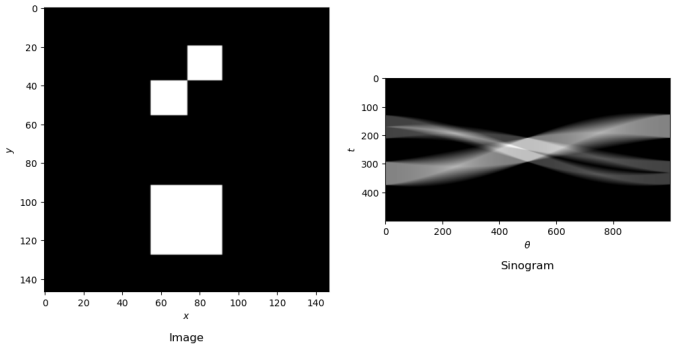

Figure 2.3. A cuboids Phantom (left) and the respective sinogram (right)

These projections are mathematically described for each line by the Radon transform in Equation 2.4, which is often referred to as the forward operator.

$$Rf(t,\theta) = \int_{l_{t,\theta}} f(x,y)dl \tag{2.4}$$

The pseudo inverse of the previous equation, namely the back projection, produces a blurred version of the original image as displayed in Figure 2.4. This happens because the back projection doesn't know the exact locations of the activities and sets a constant value in all pixels along the LOR, causing a linear overlap of the back projections and the consequent visual blurring of the image [18].

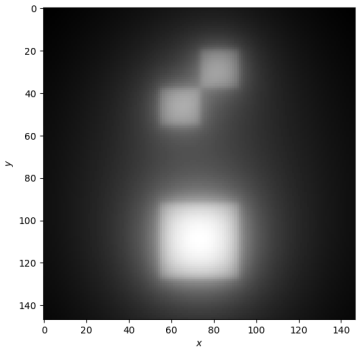

Figure 2.4. Back projection reconstruction

In presence of noise, as in a real PET sinogram, the result is even worse than the one presented in Figure 2.4. This further demonstrates that the inversion of the acquisition process is a problem that cannot be solved by simply inverting the model but requires more sophisticated strategies.

2.2.1 Analytic Algorithms

Analytical reconstruction algorithms attempt to mathematically solve the inversion problem, assuming ideal conditions and neglecting the presence of noise in PET data. For these reasons, while these methods are fast and analytically tractable, they often produce low-quality reconstructions that are still

affected by noise, requiring further denoising.

The most popular analytical algorithm is the Filtered Back Projection (FBP) [19], which attempts to solve the blurring shown above by filtering the projections in the Fourier domain before applying the back projection. Equation 2.5 presents this strategy:

$$f(x,y) = \frac{1}{2} B \left\{ F^{-1}[|\omega| F(Rf)(\omega,\theta)] \right\}(x,y) \tag{2.5}$$

where F and F^{-1} are respectively the Fourier transform and its inverse, ω is the 2D Fourier filter and B is the back projection. There are various types of filters, each offering a different balance between image sharpness and noise reduction. The reconstructed image in Figure 2.5 is the result of a FBP utilising a Ram-Lak filter on a sinogram affected by Poisson noise.

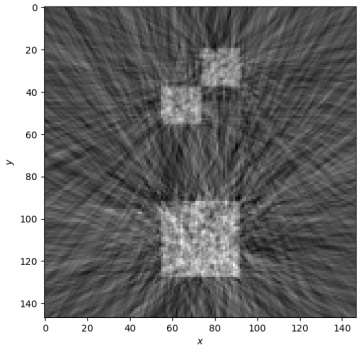

Figure 2.5. Filtered Back Projection reconstruction from a noisy sinogram

As can be observed in Figure 2.5, the FBP algorithm generates several artefacts, mainly in the background.

2.2.2 Iterative Algorithms

Iterative methods try to solve the problem by modelling it statistically. This approach has the benefit of improving reconstruction accuracy, as it aligns better with the inherently statistical nature of the emission process. However, it also introduces complexity, which renders the analytical treatment of these algorithms impracticable. This is the reason why the solution has to be found iteratively.

The most widely used iterative algorithm is the Maximum Likelihood Expectation Maximisation (MLEM) algorithm, in which Maximum Likelihood (ML) is the objective function used to calculate the similarity between the measured data g and the current estimate f_k and Expectation Maximisation (EM) is the algorithm used to correct the current estimate [20]. Equation 2.6 describes one iteration of the MLEM algorithm:

$$f_{(k+1)} = \frac{f_k}{A^* \mathbb{I}} A^* \left(\frac{g}{A f_k} \right) \tag{2.6}$$

where f_k is the current estimate, $f_{(k+1)}$ is the updated estimate, $\mathbb{I}$ is a all-ones matrix, A represents the forward projection operator and A^* its adjoint (back

projection). At each iteration, the measured data g is divided by the forward projection of the current estimate f_k. This ratio is then back projected and divided by the sensitivity matrix, defined as $A^*\mathbb{I}$.

Figure 2.6 shows the reconstructed image obtained using the MLEM algorithm for 50 iterations on the same noisy sinogram utilised before. The reconstruction is not perfect, but there are fewer artefacts and a better spatial resolution than the image produced by the FBP.

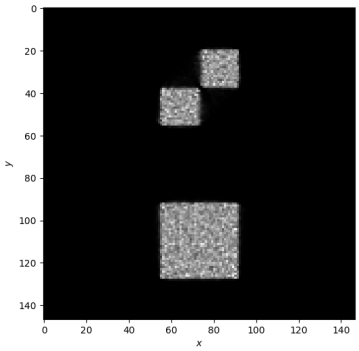

Figure 2.6. MLEM algorithm reconstruction from a noisy sinogram

Given that the convergence time of MLEM is relatively long and requires many iterations, an accelerated variant was proposed in 1994: the Ordered Subset Expectation Maximisation (OSEM) algorithm [21]. This algorithm follows the same principle as MLEM, but divides the problem into subsets, reaching execution speeds approximately n_{subsets} times faster than MLEM. The main disadvantage of this variant is that it fails to converge in some cases, especially when the number of subsets is too large.

Nowadays iterative algorithms like OSEM are common clinical practice and they represent the state-of-the-art of image reconstruction for PET [22].

2.3 Learned Primal-Dual Reconstruction

Recent advances in reconstruction involve the use of Artificial Intelligence (AI), in particular Deep Neural Networks (DNNs). These AI-based methods differ in how DNNs are utilised: some model the reconstruction problem directly, while others focus on denoising the sinogram data, the image data, or both, either directly or iteratively [23].

The Learned Primal-Dual (LPD) architecture, as described by J. Adler and O. Öktem in 2018 [4], employs a cross-domain approach utilising DNNs to enhance both sinogram and image data iteratively. The architecture comprises two interconnected branches: the Dual branch, which processes data within the sinogram domain, and the Primal branch, which operates within the image domain. In each iteration, the sinogram data is processed by a DNN, back-projected into the image domain and subsequently processed by another DNN to be later forward projected again into the sinogram domain. To improve the training process, each DNN also takes as input not only the current data, but also the outputs from all previous iterations, utilising concatenation. The full

process is described in detail in Algorithm 1.

Algorithm 1: LPD Algorithm

Given: Initial sinogram g^0, Forward Projection $\mathcal{A}$, Number of iterations N
$g^1 \leftarrow \Xi^0_{\phi_0}(g^0)$
$f^1 \leftarrow \Lambda^0_{\gamma_0}(\mathcal{A}^*(g^1))/||\mathcal{A}||^2$
for $i = 1, ..., N$ do
  $g^{i+1} \leftarrow \Xi^i_{\phi_i}(\mathcal{A}(f^i), g^{i-1}, ..., g^0)$
  $f^{i+1} \leftarrow \Lambda^i_{\gamma_i}(\mathcal{A}^*(g^{i+1}), f^i, ..., f^0))/||\mathcal{A}||^2$
end for
return f^{i+1}

where g is the acquired data (sinogram) and g^i and f^i are the updated sinogram and image data respectively at iteration i. Ξ^i_ϕ and Λ^i_θ are two families of operators, with parameters ϕ and θ, that work in the sinogram and image domains respectively and are generally represented by DNNs.

Figure 2.7 provides a visual representation of Algorithm 1 with $N_{iterations}$ set to 3 for simplicity.

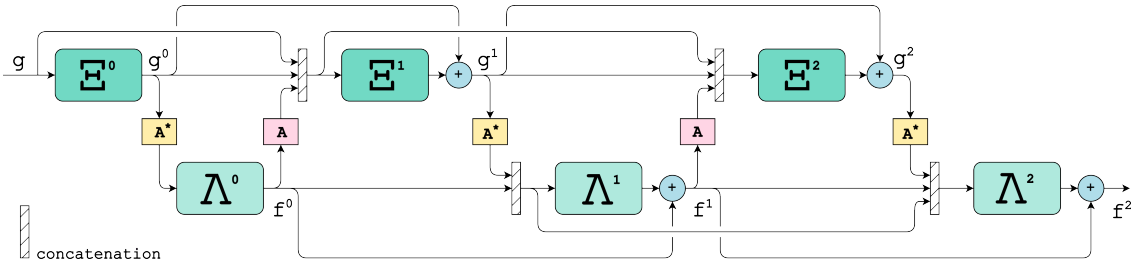

Figure 2.7. The LPD architecture featuring 3 iterations
Source: Figure created by the author

Various types of DNN can be utilised as the operators Ξ^i_ϕ and Λ^i_θ and the choice depends on which one best performs a specific task.

2.3.1 Convolutional Neural Networks

Both in the original implementation of the LPD architecture and in A. Guazzo and M. Colarieti-Tosti's adaptation for PET [5], Convolutional Neural Networks (CNNs) are used as operators. CNNs are a type of DNNs that excels at processing data with a grid topology, such as digital images. The most important components of a CNN are convolutional layers. Convolutional layers use learnable filters, or kernels, to automatically detect and capture spatial hierarchies and patterns in the input data, such as edges, textures and shapes. Figure 2.8 illustrates a simple convolution between a kernel and an input image.

CNNs are widely used in the field of computer vision, with numerous architectures incorporating convolutional layers. However, in recent years, their dominance has been challenged by a novel approach based on deep learning, namely the Vision Transformers.

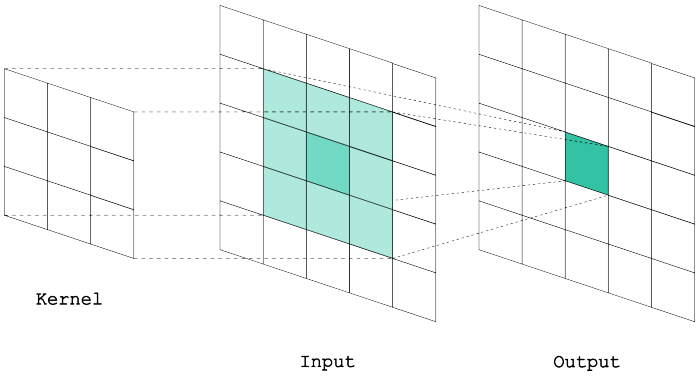

Figure 2.8. Example of convolution in a CNN
Source: Figure created by the author

2.3.2 Vision Transformers

In the most recent research about LPD reconstruction for PET, A. Adelöw implemented Vision Transformers into the architecture with superior results compared to the previous CNNs based version [6].

Transformers appeared in 2017 for Natural Language Processing (NLP) [24] and in 2021 they were proposed also for image processing under the name of Vision Transformers (ViTs) [25]. ViTs divide an input image into patches, which are flattened and embedded into vectors of a chosen length using a learnable linear transformation. These vectors are added to a Position Embedding, which provides information on the relative position of the image patches, and then given as input to an encoder. In the encoder, the Queries, Keys and Values vectors are calculated by multiplying the provided vectors with a set of trainable weights. These vectors are used to compute a new object, called Self-Attention, as described by Equation 2.7.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \tag{2.7}$$

where Q, K, V are the Queries, Keys and Values vectors respectively and d_k represents the Keys vector dimension.

The Self-Attention is typically calculated multiple times in parallel to allow the network to learn different aspects of the input data (Multi-Head Attention mechanism). As Equation 2.8 shows, multiple attentions are computed independently and then concatenated together.

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_\text{h}\right) W^O \tag{2.8}$$

where each attention head is defined as $\text{head}_\text{i} = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$.

The output of the Multi-Head Attention encoder is finally sent to a simple neural network, e.g. a Multi Layer Perceptron (MLP), from which the final output is obtained.

ViTs have the great advantage of capturing the global context of an image, unlike CNNs which are constrained by the nature of convolution to local contexts. They outperformed CNNs in many applications and proved more robust

for computer vision. However, the computational cost of the attention mechanism is typically higher and they don't generalise well when trained with small datasets [26].

2.3.3 Residual Networks

The LPD reconstruction algorithm can be defined as a "residual" network due to the presence of connections that propagate the intermediate outputs to non-consecutive blocks.

Residual networks were introduced by He et al. in the ResNet architecture [27] and allow deeper neural networks to be trained without running into the vanishing gradient issue. This problem occurs in deep networks during back-propagation when repeated multiplications make the gradients too small to update the first layers. By allowing inputs to bypass certain layers through skip connections, residual networks allow the gradient to flow through the entire network and reduce the training complexity. In fact, instead of approximating a complex function (the output), a simpler function (Equation 2.9) is learnt.

$$\mathcal{H}(x) := \mathcal{F}(x) + x \tag{2.9}$$

where x is the input.

Figure 2.9 displays the typical structure of a residual block.

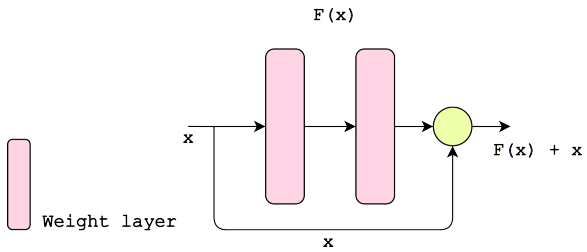

Figure 2.9. A residual block
Source: Figure created by the author

In the Figure above, the yellow circle typically represents an addition between the blocks' output and their input, but it can also be a concatenation or a product. The weight layers can be any type and in any number within a residual block.

In the specific case of the LPD architecture, residual connections are employed in two ways: concatenating the outputs of all previous blocks to form the inputs of the current block and adding the output of the current block with the output of the previous block within the same domain.

2.4 Cross-Attention

Cross-Attention has emerged as an extension of the Attention mechanism by allowing the interaction between different sets of embedded vectors, rather than focusing on a single set as in Self-Attention. In Cross-Attention, the Queries, Keys and Values vectors are derived from different sets of vectors:

typically the Queries come from one set, while the Keys and Values from another, as shown in Equation 2.10.

$$\text{CrossAttention} = \text{Attention}(Q_i, K_j, V_j) \tag{2.10}$$

where Q_i is the Queries of the vectors sequence i and K_j, V_j are the Keys and Values of the vectors sequence j. This is not the only possible combination, since (Q_j, K_i, V_j) was also successfully implemented [28].

By integrating and aligning information across multiple inputs, Cross-Attention improves the model's contextual understanding of the given task, potentially leading to better results. In the LPD architecture, integrating Cross-Attention Blocks (CABs) would facilitate the fusion of information between two different steps, guiding the network towards a more refined comprehension of the reconstruction problem.

A notable example of CABs used between two different branches is CrossViT [29], a multi-scale Vision Transformer architecture for image classification that uses Cross-Attention to facilitate interaction between branches processing image patches of different resolutions. Other examples are related to improving skip connections in U-Net based architectures. In "U-Net Transformer" [30], a Self-Attention mechanism is used to improve the bottleneck and a Cross-Attention mechanism to address the semantic gaps between the contracting and expanding paths.

2.5 Training without acquired images

In 2021 M. Hoffmann et al. presented a contrast agnostic method to train CNNs for multi-modal registration, called "Synthmorph" [31]. This method uses synthetic images, with various types of contrast, as a training dataset to ensure optimal network generalisation and remove any dependence on image acquisition characteristics.

To generate the moving m and the fixed f synthetic images, two paired 3D label maps $\{s_m, s_f\}$ are generated from a function $g(z) = \{s_m, s_f\}$ using a random seed z and then synthesised into intensity volumes with another function $g_i(s_m, s_f, z) = \{m, f\}$. An example of (noisy) fixed images generated using this strategy is shown in Figure 2.10.

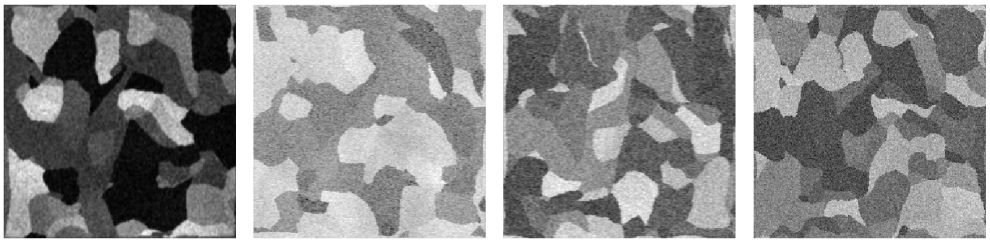

Figure 2.10. Fixed images generated using the Synthmorph approach

The presence of 3D labels overcomes the dependence of the loss function on image contrast by using a similarity function that measures the overlap between

labels and not between images, such as the Dice score described in Equation 2.11.

$$\mathcal{L}'_{dis}\left(\phi, s_m, s_f\right) = -\frac{2}{J}\sum_{j=1}^{J}\frac{\left|\left(s_m^j \circ \phi\right) \circ s_f^j\right|}{\left|\left(s_m^j \circ \phi\right) + s_f^j\right|} \tag{2.11}$$

where ϕ is the deformation field computed by the network.

This training strategy makes it possible to train networks capable of performing multi-modal registration tasks on unseen images with higher accuracy and robustness than best classical and AI-based methods [32]. Although only applied to image registration, this approach could also be advantageous for reconstruction tasks due to its ability to generalise the networks from synthetic images alone. In fact, a challenge in PET image reconstruction is the lack of large datasets of real clinical sinograms, necessitating synthetic data to train DNNs.

This page is intentionally left blank.

3 Methodology

This chapter outlines the methodology and workflow employed to achieve the project objectives. The first sections explain the data generation process and how DNN architectures were built and trained. The last part presents the results evaluation methods and the experimental setup used.

3.1 Data Generation

To give models a good generalisation, training was carried out exclusively with sinograms generated from synthetic 3D phantoms that expose the model to a wide variety of shapes and structures. These images were randomly created on the fly, ensuring diversity and novelty with each iteration. This made the use of a separate validation set for tuning the hyperparameters unnecessary. The trained models were then validated on both synthetic and real data to evaluate their performance and test their generalisation power.

Synthetic data were generated to follow the same characteristics as the real data that was collected using a small-animal PET system, the MiniPET-3 [33]. This scanner is equipped with 35 detection rings, each of which consists of 12 detection modules with 35×35 mm Lutetium-yttrium oxyorthosilicate crystals. The inner diameter of a detection ring is 211 mm. Figure 3.1 depicts a 3D representation of the MiniPET-3 scanner geometry.

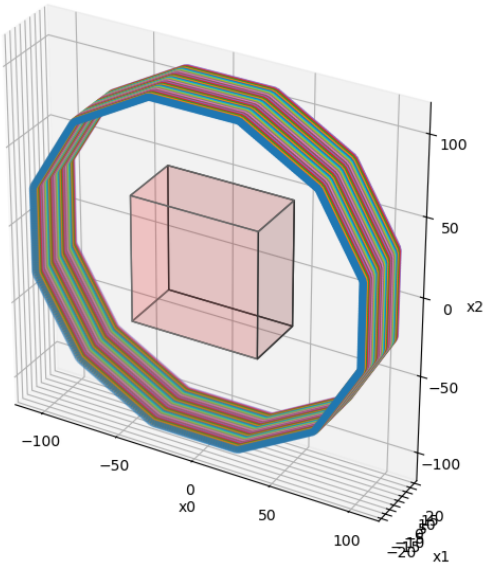

Figure 3.1. MiniPET-3 scanner geometry with the acquisition volume

The above geometry was used to create a projector that was capable of generating synthetic sinograms from images geometrically similar to those obtained in reality. Specifically, 3D blurred phantoms of size $147 \times 147 \times N$, where N is the

depth of the volume that can be chosen between 1 and 35 (the total number of rings), were used to create the synthetic sinograms. The blurring of the original volume before the forward projection was necessary to better simulate the spatial resolution of the emission process. It was applied through a Gaussian filter with a standard deviation of 2 and a kernel size of 5. The choice of these parameters depends on the positron range of the Fludeoxyglucose used to obtain the experimental images, which is between 0.6 mm and 2.4 mm [34], corresponding to maximum 5 pixels in the image. Lastly, to account for the Poissonian nature of the radioactive emissions and detection [35], a Poissonian-like noise was added to the sinograms with an intensity varying between 0.1 and 1.2 to cover a wide range of noise scenarios.

3.1.1 Training Data

The images in the training dataset were randomly generated on the fly using two approaches: one based on ellipsoidal labels and the other on randomly shaped labels. Each approach had a 50% chance of being selected to generate an image of the training set.

3.1.1.1 Random Ellipsoid Labels

Random ellipsoid labels were created by generating multiple ellipsoids in a 3D space and associating each one with a unique integer number (from zero to 6, the set maximum number of ellipsoids in the same label map).

For the creation of each ellipsoid, the equation of an ellipsoid in three-dimensional space was used, as described by Equation 3.1:

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}+\frac{(z-z_0)^2}{c^2}=1 \tag{3.1}$$

where a, b and c are the semi-axes of the ellipsoid along the x, y and z axes, and x_0, y_0 and z_0 are the coordinates of the ellipsoid's center in the 3D space. These parameters were randomised to produce different ellipsoids at each iteration. A random rotation was also performed to vary the spatial alignment.

Since this generation strategy does not produce particularly complex structures, each label map was summed with a similarly generated label map to create overlaps and then subtracted with another label map to form holes. A visualisation of this process is shown in Figure 3.2.

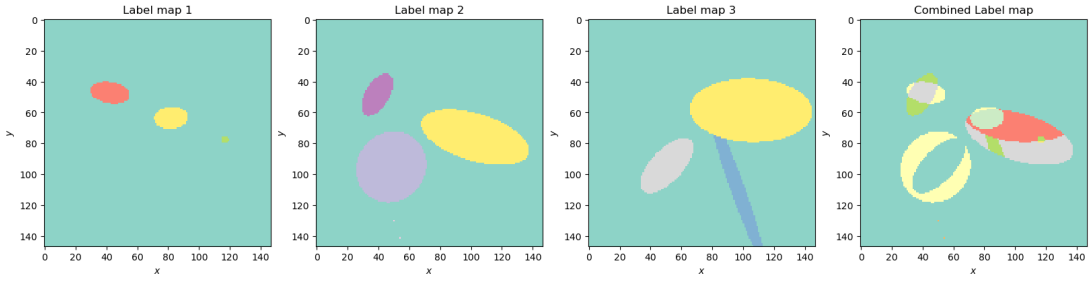

Figure 3.2. Three randomly generated ellipsoids before (left) and after (right) merging

In the image above, label maps 1 and 2 are added together, while label map 3 is subtracted.

3.1.1.2 Random Shape Labels

The randomly shaped labels were created utilising an approach similar to the one used by Synthmorph [31]. Firstly, as many 3D volumes as the number of desired labels, set again to 6, were created and filled with Peril noise.

A 3D gradient was subtracted from each of these volumes to limit the Peril noise in the centre of the volumes as shown in Figure 3.3. The 3D gradient was computed by calculating the Euclidean distance of each voxel from the centre of the volume and then multiplying it by a random scaling factor to allow for different gradient sizes.

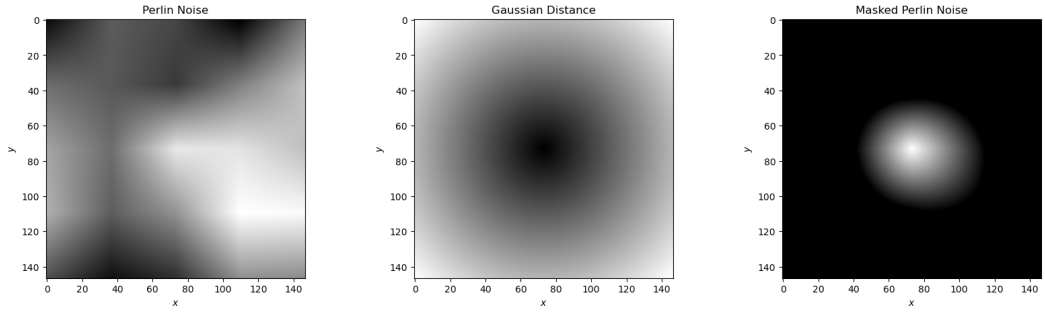

Figure 3.3. Peril noise and gradient before (left) and after the subtraction (right)

This step, which is not present in Synthmorph, was necessary to better simulate real acquisitions, where there is usually an object in the centre of the scanner's field of view and nothing at the edges.

These so-processed 3D Peril noise volumes are organised in a object with shape of (H, W, D, B), where H, W, D are the spatial dimensions and B is the number of 3D volumes, equal to the number of labels. For every position (h, w, d) in the (H, W, D) dimensions:

- the values of all 3D volumes (B) at that specific position are compared;
- the volume (index along B), which contains the maximum of these values, is identified;
- the identified index is assigned at the position (h, w, d) in the 3D label map.

Therefore, the range of possible values in the final 3D label map includes integer values between 0 and $B - 1$. Since one of the 3D volumes could not contain the maximum value at any of the positions (h, w, d), it could happen that its index is not present in the 3D label map and thus the total number of unique labels is less than B.

Figure 3.4 illustrates this process for the generation of a simple 2D label map with dimensions 3×3 and n_{labels} equal to 3.

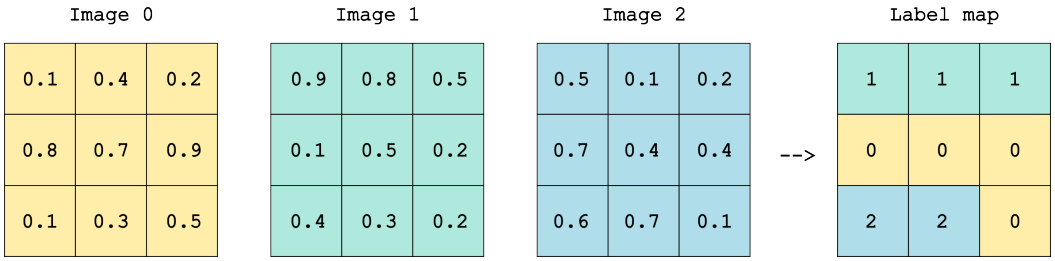

Figure 3.4. Label map generation process
Source: Figure created by the author

Similarly to the previous approach, two different label maps were added together to form the final label map as displayed in Figure 3.5.

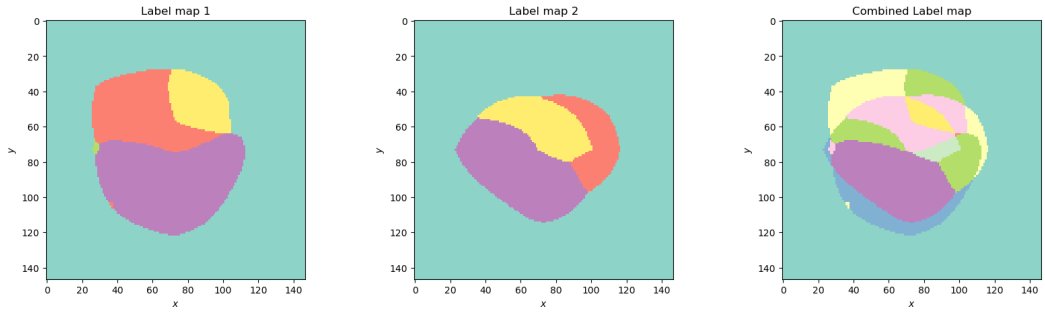

Figure 3.5. Two randomly generated shapes before (left) and after (right) merging

3.1.1.3 From Labels to Images

A function processed these label maps to generate the synthetic images like the one presented in Figure 3.6.

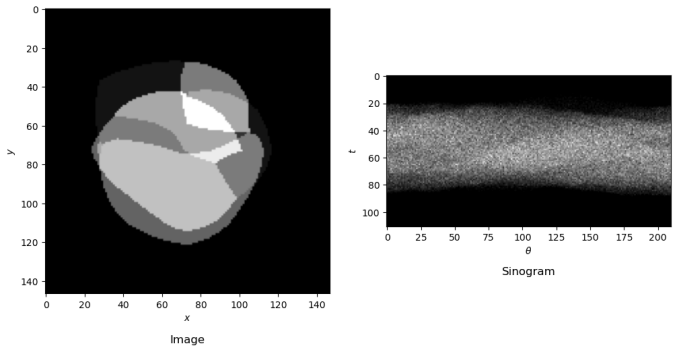

Figure 3.6. A randomly generated image (left) and the respective noisy sinogram (right)

This process started by assigning unique intensities to each label in the 3D label map. The intensities were evenly distributed in the range from 0 to 255, each perturbed with a small random offset (within one-third of the distance between the intensities) to maintain a clear distinction between intensity levels and sufficient variability. Min-max normalisation was finally applied to the images to scale the values between 0 and 1, as described by Equation 3.2:

$$x_{\text{min-max}} = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{3.2}$$

where x is a voxel value and X is the set of values from which x is taken.

The associations between labels and intensity levels were saved to allow the creation of the label maps from the images reconstructed by the model.

3.1.2 Synthetic Test Data

A 3D Shepp–Logan phantom [36] tested the reconstruction methods on synthetic, yet realistic data. This phantom is an abstract representation of a human brain and it's the most common test object for testing reconstruction algorithms [37]. Figure 3.7 illustrates one of the central slices of a 3D Shepp–Logan phantom with its noisy sinogram.

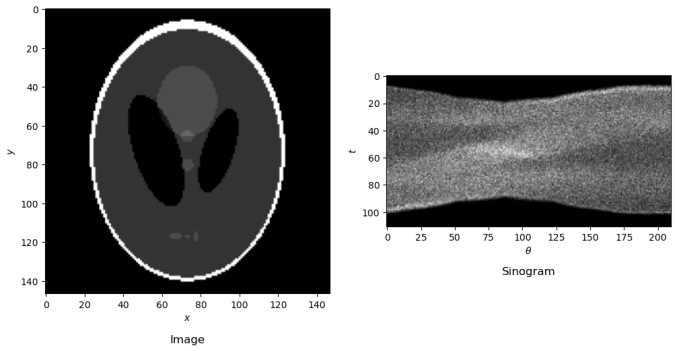

Figure 3.7. A slice of the Shepp-Logan Phantom (left) and the respective noisy sinogram (right)

3.1.3 Pre-clinical Test Data

Experimental data were used to validate the reconstructions in a realistic scenario. These data were acquired by A. Guazzo and M. Colarieti-Tosti using the MiniPET-3, for the "Learned Primal Dual Reconstruction for PET" article [5]. The test object is a mouse-like phantom whose 3D-printed model can be seen in Figure 3.8.

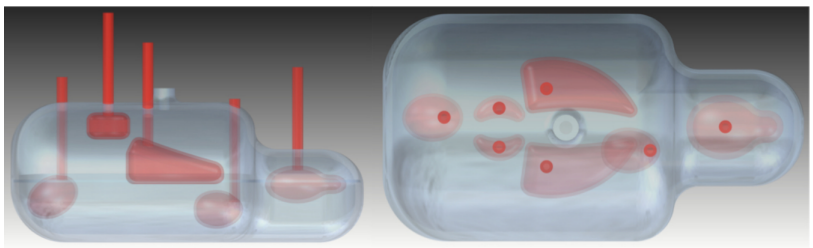

Figure 3.8. Side view (left) and top view (right) of mouse-like test phantom
Figure taken from [5]

The mouse-like phantom was loaded with Fludeoxyglucose (FDG), and the various zones' activity concentrations are described in Table 3.1. Using this experimental setup, 60 one-minute-equivalent long acquisitions were collected.

Table 3.1. Activity concentrations in the different areas of the mouse-like phantom
Table taken from [5]

T	Body $[\frac{MBq}{mL}]$	Brain$[\frac{MBq}{mL}]$	Heart$[\frac{MBq}{mL}]$	Lungs$[\frac{MBq}{mL}]$	Kidneys$[\frac{MBq}{mL}]$	Bladder$[\frac{MBq}{mL}]$
M1	0.5	1.1	0.1	0.15	0.8	1.3
M2	0.4	1.1	0.1	0.07	0.9	-

Since the nature of these data is experimental, the original image is unknown. However, the scanner has its reconstruction algorithm, based on MLEM, which provides an image at the end of the scan. This coarse reconstruction is based not only on the direct sinograms, but also on the indirect ones, and was used as a visual reference to evaluate the quality of the other reconstructions. This visual reference with the acquired experimental sinogram is shown in Figure 3.9.

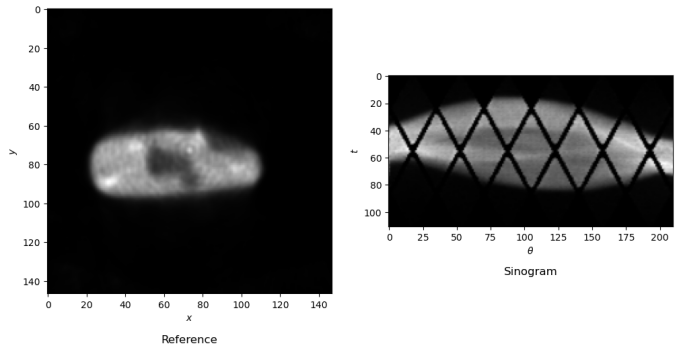

Figure 3.9. A slice of the MiniPET-3 reconstructed image (left) and the respective sinogram (right)

The oblique lines in the sinograms are due to the space between the crystals in the scanner.

3.2 Reconstruction Methods

Except for the clinical standard algorithm, all the architectures described in this Section are different implementations of the Learned Primal-Dual reconstruction algorithm. Two of these, the U-Net LPD and the Dual Domain Transformer LPD, are briefly presented for comparative purposes and are implemented as described in [5] and in [6], respectively.

3.2.1 OSEM

The OSEM algorithm, a fast variant of MLEM, was selected as baseline method. It was implemented using the built-in methods provided by the ”parallelproj” project [38], on which all the forward and back projection operations were based. The number of subsets was set to 2 and the number of iterations to 9 as this is the one that produces the best results as shown in Figure 3.10. Nine OSEM iterations correspond to 18 MLEM iterations.

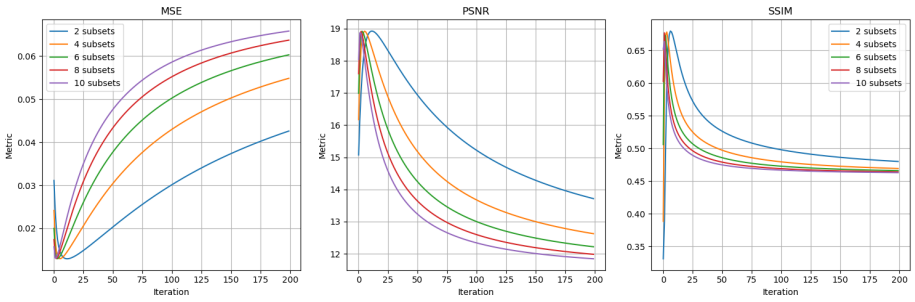

Figure 3.10. Metric values as the number of iterations and subsets increases

3.2.2 U-Net LPD

The U-Net LPD architecture uses 3 iterations and U-Nets [39] as operators for the denoising in both the sinogram and the image domains. The U-Net is arguably the most popular architecture for medical image processing tasks and it is based on a symmetric encoder-decoder structure with skip connections to capture both fine and global details.

The encoding path used convolutional layers followed by down-sampling layers, while the decoding path used the same convolutional layers, but was followed by up-sampling layers to restore the original resolution. The convolutional layers consisted of a 3×3 convolution, a batch normalisation layer and a ReLU activation function. 2×2 max pooling and transposed convolutions were used for down-sampling and up-sampling respectively. A 1×1 convolution brought back the output to a single channel and produced the final model result.

As these U-Nets were designed to process 2D inputs, each input 3D volume was treated as a series of 2D images during the processing steps. The chosen number of layers of depth was 3, as shown by Figure 3.11.

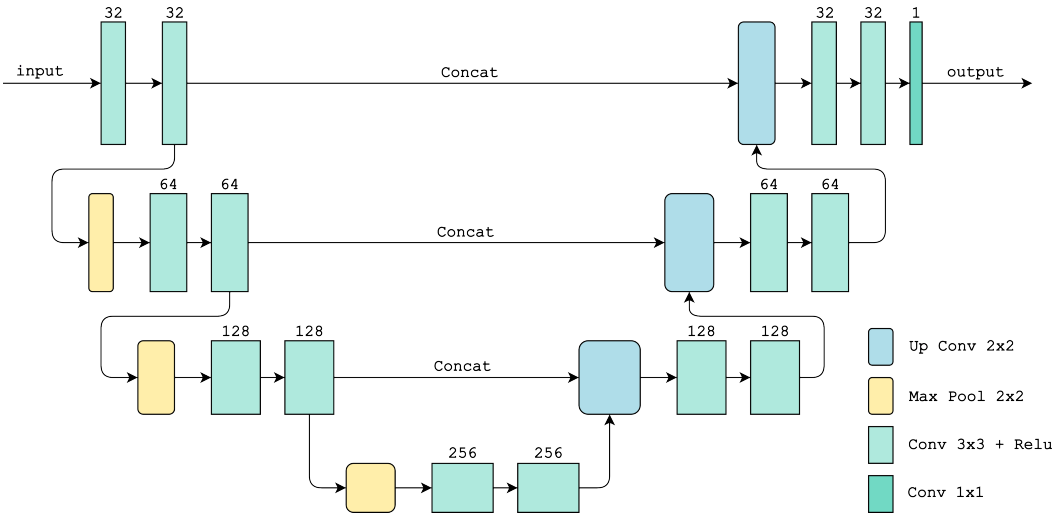

Figure 3.11. The 3-layers U-Net used for the Primal and Dual domains
Source: Figure created by the author

3.2.3 Dual Domain Transformer LPD

The architecture of the Dual Domain Transformer LPD differs from the U-Net LPD implementation because it employs 2D Vision Transformer blocks instead of U-Nets in the sinogram domain (dual domain).

To create embeddings in the sinogram domain, the input had to be treated as a sequence of projections for the assumption of the information's locality contained in a image patch to hold. Points belonging to the same bundle of projections were identified by forward projecting small 7×7 patches from the image domain to the sinogram domain, as illustrated in Figure 3.12.

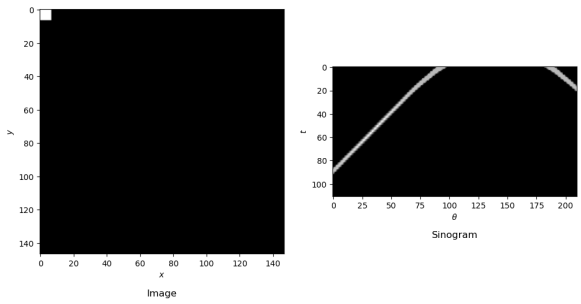

Figure 3.12. Process for the sinogram patching (7×7 patches)

Once the input sinograms were patched, a linear layer mapped the values of each curve into an embedding of length of 312 to which a learnable positional encoding was added. The embedded vectors were processed by a Multi-Head Attention block, normalised, fed through to a MLP and then normalised again. To obtain the output sinogram, the embeddings were unembedded using another linear layer and weighted according to the activation level of the curves. This structure is presented in Figure 3.13.

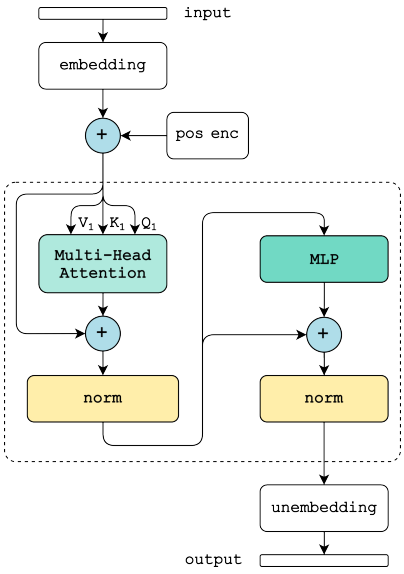

Figure 3.13. The Self-Attention Transformer block used in the Dual Domain
Source: Figure created by the author

The image processing was done in 2D, as before, and the Self-Attention

Transformer block implemented with 2 heads.

3.2.4 3D U-Net LPD

This architecture is a modified version of the U-Net LPD that was adapted to work directly in 3D. In each U-Net, the two-dimensional convolutions and max poolings were replaced by the three-dimensional ones, allowing the input volumes to be processed in 3D in all steps of the LPD algorithm.

3.2.5 Cross Image U-Net LPD

The Cross Image U-Net LPD adds Cross-Attention Blocks (CABs) in the primal domain to further process the image. Each Cross-Attention Block was implemented as displayed in Figure 3.14, with two parallel Transformer blocks that exchange their respective Queries vectors to compute the Cross-Attention. The outputs of the two blocks were added together to create the final output.

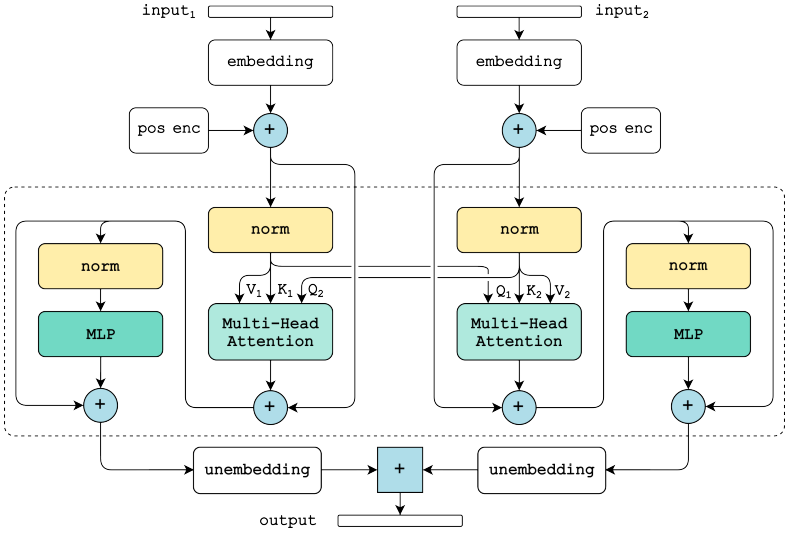

Figure 3.14. The Cross-Attention Transformer block
Source: Figure created by the author

In particular, two CABs were added, one after Λ^1 and another after Λ^2. Both take as input the output of the block Λ^i and the back projection of the output of the previous block Ξ^i. Figure 3.15 shows these modifications on the LPD architecture diagram.

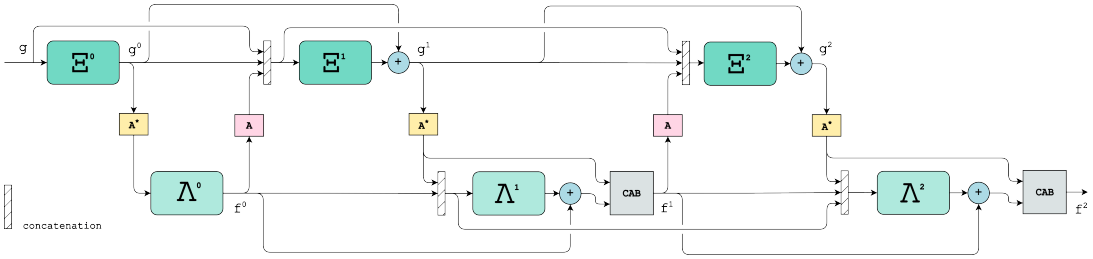

Figure 3.15. The Cross Image U-Net LPD
Source: Figure created by the author

In the CABs, the input image was divided into 7×7 patches and each patch was linearly transformed into a 64 long embedding. The size of 64 was chosen as the closest power of 2 to 49, such as the number of values in each patch. A fixed positional encoding, based on sine and cosine functions was used to encode the spatial information into the input embeddings and 2 heads were employed in the Multi-Head Attention blocks to compute the Attention matrices.

By using as inputs of each CAB, the output processed by the main operator and the most recent forward/back projection output, this modified LPD aimed to recover potential real information lost during the intensive processing performed by the U-Nets.

3.2.6 Cross Sinogram U-Net LPD

This implementation mirrors the previous one, implementing the CABs presented above in the sinogram domain instead of the image domain as illustrated by Figure 3.16.

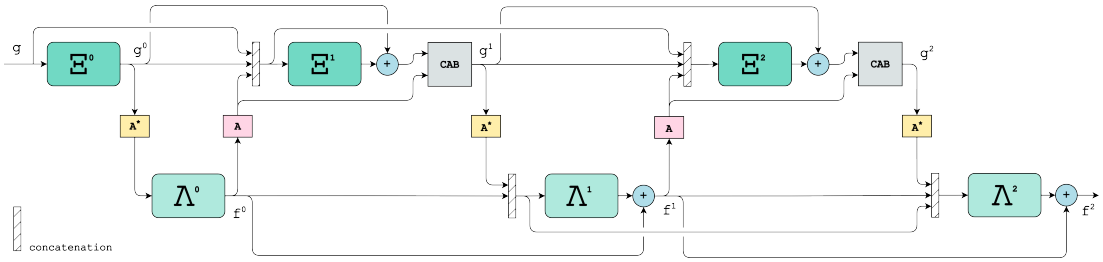

Figure 3.16. The Cross Sinogram U-Net LPD
Source: Figure created by the author

The same strategy outlined for the Dual Domain Transformer LPD was used to embed the input sinograms. In this case, the patching used 7×7 and 512 long embeddings. The positional encoding was fixed and each Multi-Head Attention block was created with 2 heads.

3.2.7 Cross Update U-Net LPD

This proposal aims to enhance the residual connections used for updating the primal and dual blocks' outputs by replacing the simple addition operations with CABs. Figure 3.17 illustrates the LPD architecture with four CABs that follow the same structure presented in Figure 3.14

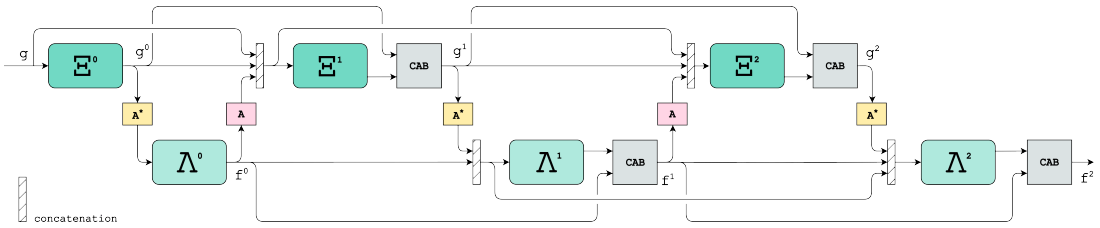

Figure 3.17. The Cross Update U-Net LPD
Source: Figure created by the author

The embedding size for the CABs in the primary domain was set to 64, while for the dual domain, the embedding was configured with a length of 512. A patch dimension of 7×7 was used in both domains.

3.2.8 Cross Concatenation U-Net LPD

Instead of enhancing the residual connections responsible for updates, the Cross Concatenation U-Net LPD focuses on enhancing those that handle concatenations to provide the inputs to the primary and dual blocks. The goal is to supply the blocks not with a direct concatenation of previous inputs, but with a processed version of these inputs using a CAB. Since the structure previously used for the CABs accommodates only two inputs and the inputs to be concatenated are typically more than two, the structure shown in Figure 3.18 was implemented. This structure can accept an arbitrary number of inputs, with Cross-Attention calculated between the Queries vector of the current input and the concatenated Values and Keys vectors of all the inputs. This CAB is inspired by one proposed in [40].

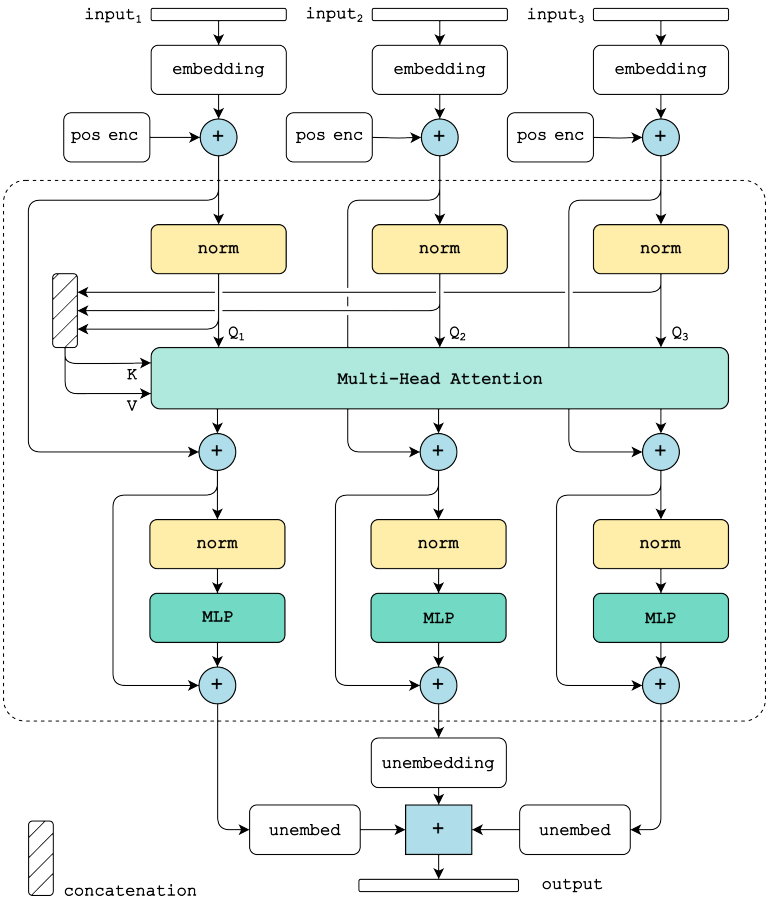

Figure 3.18. The Cross-Attention Transformer block with multiple inputs
Source: Figure created by the author

Figure 3.19 displays the modified LPD architecture. The same embeddings and patch sizes of the previous architecture were used, except for the sinogram embedding size which was set to 384 instead of 512 due to memory limitations.

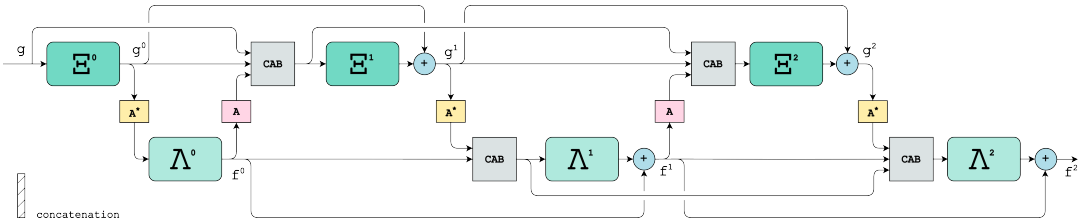

Figure 3.19. The Cross Concat U-Net LPD
Source: Figure created by the author

3.3 Training Strategy

All the 2D architectures were trained with 500 synthetic volumes of dimensions $147 \times 147 \times 21$ each epoch, while the 3D architecture was trained with 500 synthetic volumes of dimensions $147 \times 147 \times 35$ each epoch. Due to the GPU memory limitations, the batch size was set to 6.

The maximum number of epochs was set to 200. After the model had trained for 100 epochs, an early stopper was activated to end the training if the model's loss function stopped improving for 20 successive epochs.

Model weights were initialised using a normal distribution with a mean of 0 and a standard deviation of 0.01 for fully connected or convolutional layers. If the layer included a bias term, this was set to a small constant value of 0.01. This weight initialisation technique favours fast and efficient convergence during the training process.

AdamW [41] was the optimiser chosen to train all the models. It is a modified version of the Adam optimiser [42], that decouples the weight decay from the learning rate to regulate the weights better. Weight decay is an adjustment technique that favours the learning of simpler functions to generalise unseen data better. It was applied with a coefficient of $1e^{-2}$ to all linear and convolutional layers. The base learning rate used was $1e^{-4}$.

The optimiser was coupled with the 1cycle learning rate scheduler, known as "OneCycleLR" [43]. This learning rate scheduler changes the learning rate cyclically with each batch, where the minimum is the base learning rate and the maximum is set five times higher ($5 \times 1e^{-4}$). This policy helps models converge faster and achieve better performance.

To decrease training time and memory usage, Automatic Mixed Precision (AMP) training was used, which automatically utilises 16-bit floating-point precision instead of the standard 32-bit precision where possible [44]. Mixed precision was manually disabled when calculating the Attention scores (Equation 2.7) to avoid numerical instability. In addition, gradient norm clipping was implemented to prevent the gradient from exploding, with the gradient norm threshold set to 5.

3.3.1 Loss Function

The chosen loss function $\mathcal{L}$ combined a L2 loss computed between the original image and the reconstructed image and a Dice Score between the original labels and the reconstructed labels, as described in Equation 3.3.

$$\mathcal{L} = \alpha \cdot (100\mathcal{L}_{\text{L2}}) + (1 - \alpha) \cdot (1 - \mathcal{L}_{\text{Dice Score}}) \tag{3.3}$$

where α is a weighting factor (between 0 and 1) that controls the trade-off between L2 and Dice Score losses. The parameter α was set to 0.5 as the best compromise between the two metrics.

The L2 loss was scaled by a factor of 100 to make it comparable to the Dice score. This adjustment was necessary because the magnitude of L2 loss is approximately 10^{-3} in this application, whereas the Dice score is in the range of 10^{-1}.

The L2 or Mean Squared Error loss (Equation 3.5) penalises large errors more than the presence of the square, focusing on creating smooth images without large deviations. Meanwhile, the Dice Score ensures that the shapes are reconstructed correctly regardless of the precise pixel values. The Dice Score was calculated between each one-hot encoded label map, as described by Equation 3.4, and then the results were averaged.

$$\text{Dice} = \frac{2 \times \text{Area of overlap}}{\text{Total area}} \tag{3.4}$$

where the "Area of overlap" represents the region shared by both the original label map and the reconstructed label map. "The Total area" is the sum of the areas of both the original and reconstructed label maps, such as the union of both regions.

The reconstructed label maps were generated by following the process illustrated in Figure 3.20.

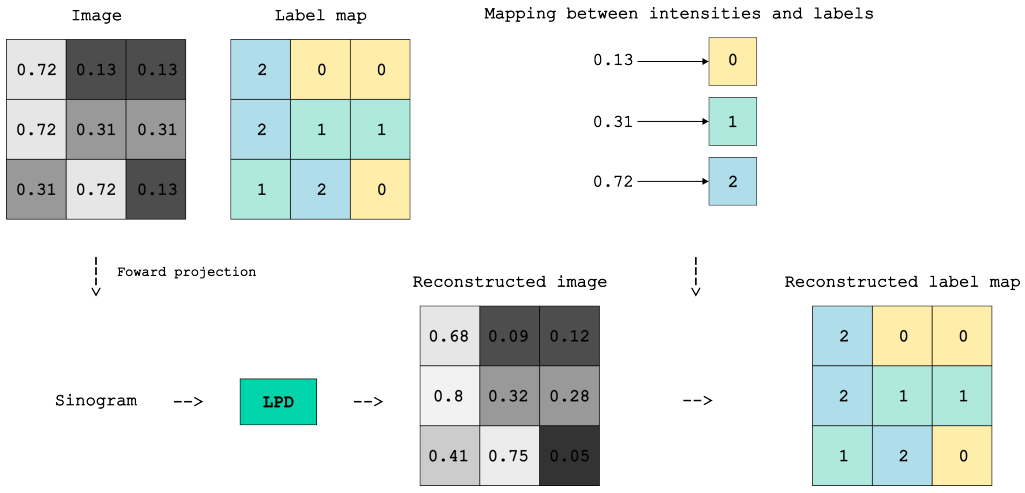

Figure 3.20. Label map reconstruction process
Source: Figure created by the author

In the example above, the two 3×3 matrices on the left represent a training image and its corresponding label map. Each label is associated with a specific value in the image, as defined by the mapping in the upper right corner. When the image is transformed into a sinogram and processed by an LPD model, a reconstructed version of the original image is obtained as output (bottom center). To obtain the label map from this reconstructed image, each value in the

reconstructed image is assigned the label corresponding to the closest value in the original mapping (top right).

3.4 Results Evaluation

Reconstruction methods were evaluated according to the quality of the reconstructed test image from its noisy sinogram. It was assessed qualitatively and quantitatively using the metrics described below for synthetic test data. However, for pre-clinical data, the assessment was limited to qualitative considerations due to the absence of the original image for comparison.

3.4.1 Image Quality

The image quality of the reconstructions was assessed with three commonly used "full-reference" metrics, which estimate how dissimilar the reconstructed image was from the original image.

The first two metrics were two measures of absolute error, the Mean Squared Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). The MSE calculates the average of the squared differences between the test and reference image pixels. Mathematically, it is defined by Equation 3.5.

$$\mathrm{MSE} = \frac{1}{x\,y} \sum_{x,y} \left[I(x,y) - \hat{I}(x,y) \right]^2 \tag{3.5}$$

where I is the reference image and $\hat{I}$ is the test image. The MSE penalises larger errors due to the presence of the square, making it sensitive to outliers. The closer the MSE is to 0, the more similar the test image is to the reference image.

On the other hand, the PSNR expresses the ratio between the maximum possible power of a signal and the power of the noise that affects the image to be tested, as shown by Equation 3.6.

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}} \right) \tag{3.6}$$

where MAX is the maximum pixel value of the reference image and MSE is the Mean Squared Error between the two images. Therefore, the PSNR decreases if the MSE increases. This metric is expressed in decibels (dB) and higher values indicate better image quality.

The last metric was the Structural Similarity Index Measure (SSIM) [45], which considers the luminance, contrast and structural similarity between the reference and test images. This approach aligns more with human visual perception than the previous two metrics [46]. Equation 3.7 describes how the SSIM is calculated.

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.7}$$

where x and y are two windows of common size $N \times N$ and μ_x and μ_y are the average of their respective intensities. The variances of x and y are represented

by σ_x^2 and σ_y^2, while σ_{xy} is the covariance between x and y. C_1 and C_2 are constants to stabilise the division. The possible values of SSIM range from -1 to 1, where 1 indicates a perfect structural similarity.

3.5 Experimental Setup

The programming language chosen for all the scripts and the Jupyter Notebooks was Python (version 3.11.9). The "PyTorch" package [47] was used to compute mathematical operations, to describe and train the Deep Neural Networks. The definition of the PET scanner geometry and the execution of forward and back projections leveraged the functionalities provided by the "parallelproj" package [38]. Lastly, the "odl" package [48] was utilised to generate the test images (Shepp-Logan phantoms) and the "TorchMetrics" package [49] to compute the selected metrics in 3D.

The project code is publicly available at https://github.com/binarypillow/HL205X-2024

3.5.1 Hardware

The training was performed on a virtual machine provided by the School of Engineering Sciences in Chemistry, Biotechnology and Health (CBH) with the following specifications:

- CPU: Intel™ Xeon™ E5-2690 v3
- RAM: 512 GB DDR4
- GPU: NVIDIA RTX™ A6000 (48GB GDDR6)

This page is intentionally left blank.

4 Results

This chapter presents the main findings, highlighting the most important research results. The first part focuses on the methodological aspects, which are important to assess the effectiveness of the training strategy. The second part presents the reconstructions produced by the proposed architectures and the corresponding metrics. To account for the variability introduced by the training's randomness, each result is derived from 3 independent experiments.

4.1 Architectures

The architectures differ in several technical characteristics, including the number of parameters, VRAM and epochs required. While these factors do not directly impact the quality of the results, they are reported in Table 4.1 as they may be important if the computational resources are limited.

Table 4.1. Comparison of architectures based on number of parameters, VRAM usage and number of epochs

	Parameters	VRAM	Number of Epochs
U-Net LPD	12 862 278	26.85 ± 0.22 GB	200.00 ± 0.00 epochs
Dual Dom. Transformer LPD	13 560 012	17.91 ± 0.41 GB	200.00 ± 0.00 epochs
3D U-Net LPD	38 527 686	31.04 ± 0.20 GB	194.00 ± 8.49 epochs
Cross Image U-Net LPD	12 946 314	26.34 ± 0.05 GB	200.00 ± 0.00 epochs
Cross Sinogram U-Net LPD	22 459 858	33.43 ± 0.12 GB	182.67 ± 24.51 epochs
Cross Update U-Net LPD	22 543 894	36.64 ± 0.14 GB	183.67 ± 23.10 epochs
Cross Concat U-Net LPD	22 792 048	43.79 ± 0.06 GB	200.00 ± 0.00 epochs

Only about 0.54 ± 0.03 GB of VRAM is necessary to process the data; the rest is divided between the model initialisation and the PyTorch autograd, which uses the most VRAM.

4.2 Synthetic Training Data

The synthetic dataset generator was constructed to generate both random ellipsoids and random shapes. Table 4.2 shows the figure of metrics computed on the Shepp-Logan phantoms reconstructed by the U-Net LPD trained with three datasets: one composed only of random ellipsoids, another of only random shapes and the third combining both random ellipsoids and shapes.

Table 4.2. Metrics calculated on the Shepp-Logan phantoms reconstructed using different datasets

	MSE	SSIM	PSNR
Random Ellipsoids	$7.58\text{e-}03 \pm 0.30\text{e-}03$	$0.92 \pm 1.59\text{e-}03$	21.21 ± 0.17
Random Shapes	$9.96\text{e-}03 \pm 0.13\text{e-}03$	$0.90 \pm 1.22\text{e-}03$	20.02 ± 0.06
Random Ellipsoids + Shapes	$8.44\text{e-}03 \pm 0.20\text{e-}03$	$0.91 \pm 3.22\text{e-}03$	20.74 ± 0.11

To obtain the results above, the sinogram was generated with a noise level of 0.5. The reconstructed Shepp-Logan phantoms and mouse-like phantoms are shown in Figures 4.1.

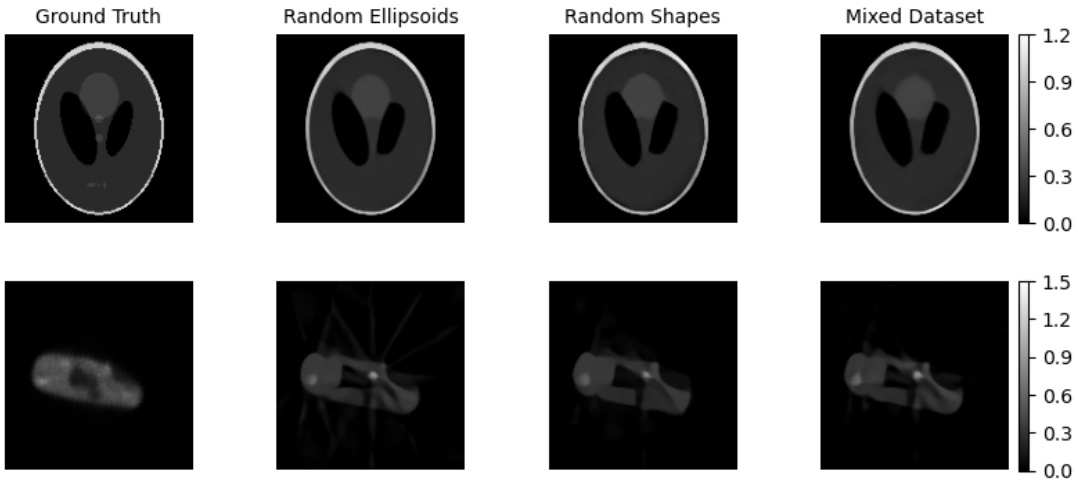

Figure 4.1. The Shepp-Logan and mouse-like phantoms reconstructed using different datasets

4.2.1 Dataset Variety

The synthetic generator produces phantoms in a wide variety of shapes and sizes due to the randomness of the generation process. In Figure 4.2, the central slices of 24 different synthetic phantoms are presented as examples.

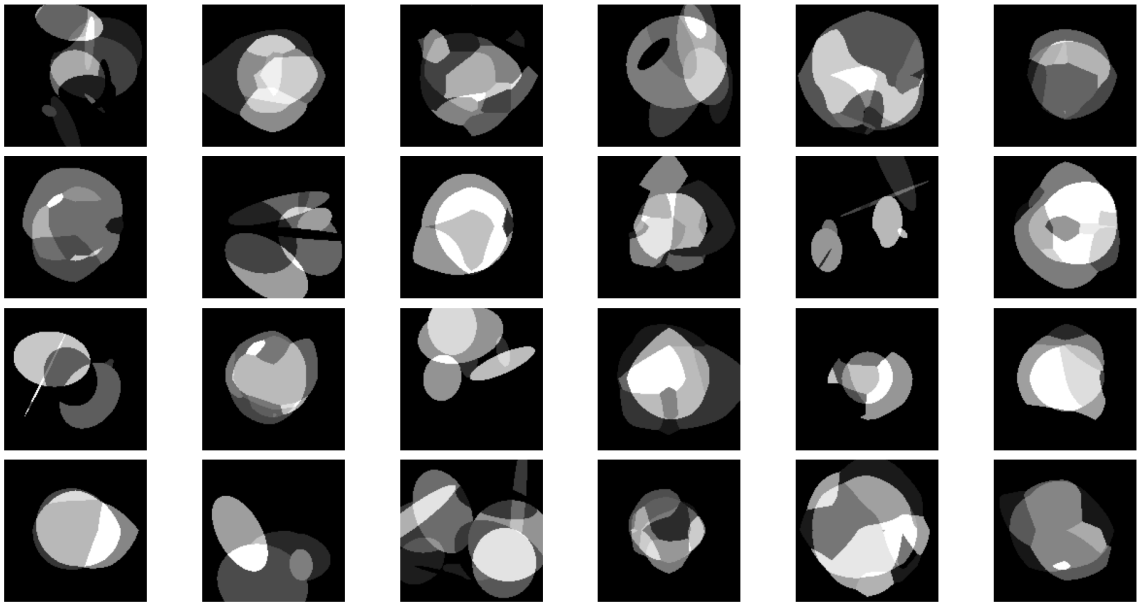

Figure 4.2. Examples of randomly generated synthetic objects

The time needed to generate a batch of 6 random volumes is 1.08 ± 0.11 seconds.

4.3 Mixed Loss Function

The effectiveness of the chosen loss function was tested against two commonly used loss functions in image restoration tasks: Mean Squared Error (MSE) and Mean Absolute Error (MAE). In Table 4.3, the metrics, obtained on the reconstruction of the Shepp-Logan phantom using the previous two loss functions and the proposed loss functions (MSE+Dice score), are reported. The model used in all the tests was the U-Net LPD and the sinogram of the Shepp-Logan phantom was generated with a Poisson noise level of 0.5.

Table 4.3. Metrics calculated on the Shepp-Logan phantoms reconstructed using different loss functions

	MSE	SSIM	PSNR
MSE	8.73e-03 $\pm$ 0.54e-03	0.90 $\pm$ 8.88e-03	20.60 $\pm$ 0.27
MAE	10.50e-03 $\pm$ 0.22e-03	0.87 $\pm$ 20.30e-03	19.78 $\pm$ 0.09
MSE+Dice score	8.44e-03 $\pm$ 0.20e-03	0.91 $\pm$ 3.22e-03	20.74 $\pm$ 0.11

Figure 4.3 illustrates the reconstructed Shepp-Logan and mouse-like phantoms.

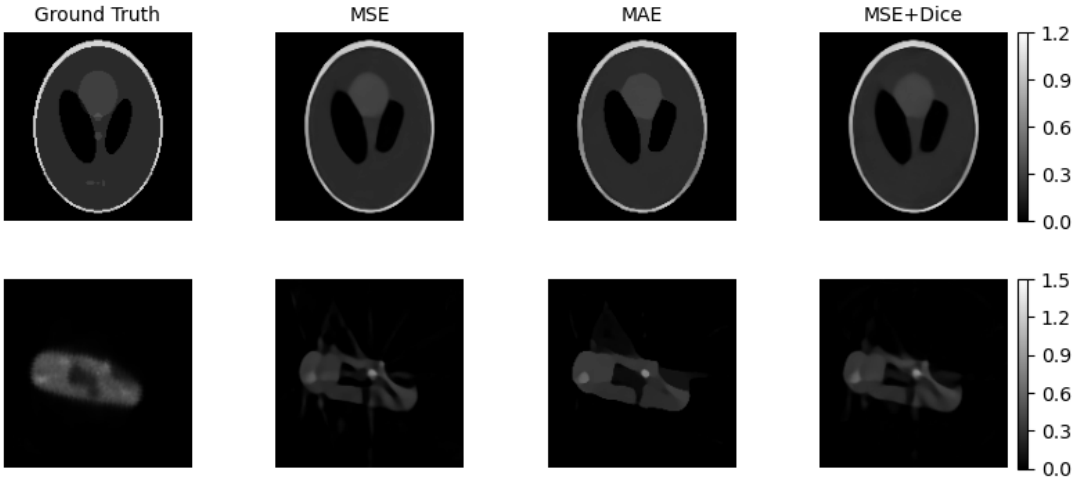

Figure 4.3. The Shepp-Logan phantoms reconstructed using different loss functions

4.4 Synthetic Test Data

The architectures described in Section 3.2 were trained using the mixed dataset and the MSE+Dice score loss function. Table 4.4 presents the metrics calculated on the reconstructions obtained from the Shepp-Logan sinogram generated with a noise level of 0.5. Correspondingly, Figure 4.4 illustrates the central slice of Shepp-Logan phantoms reconstructed by each model. In the first row, the original Shepp-Logan's slice is shown for reference.

Table 4.4. Metrics calculated on the reconstructed Shepp-Logan phantoms

	MSE	SSIM	PSNR
OSEM	13.7e-03	0.63	18.66
U-Net LPD	8.44e-03 $\pm$ 0.20e-03	0.91 $\pm$ 3.22e-03	20.74 $\pm$ 0.11
Dual Domain Transformer LPD	9.16e-03 $\pm$ 0.19e-03	0.90 $\pm$ 1.80e-03	20.38 $\pm$ 0.09
3D U-Net LPD	3.67e-03 $\pm$ 0.39e-03	0.95 $\pm$ 4.53e-03	24.38 $\pm$ 0.49
Cross Image U-Net LPD	10.10e-03 $\pm$ 2.86e-03	0.88 $\pm$ 40.09e-03	20.11 $\pm$ 1.15
Cross Sinogram U-Net LPD	9.04e-03 $\pm$ 0.55e-03	0.89 $\pm$ 18.62e-03	20.44 $\pm$ 0.27
Cross Update U-Net LPD	8.87e-03 $\pm$ 0.28e-03	0.87 $\pm$ 44.52e-03	20.52 $\pm$ 0.13
Cross Concat U-Net LPD	9.07e-03 $\pm$ 0.43e-03	0.90 $\pm$ 7.42e-03	20.43 $\pm$ 0.21

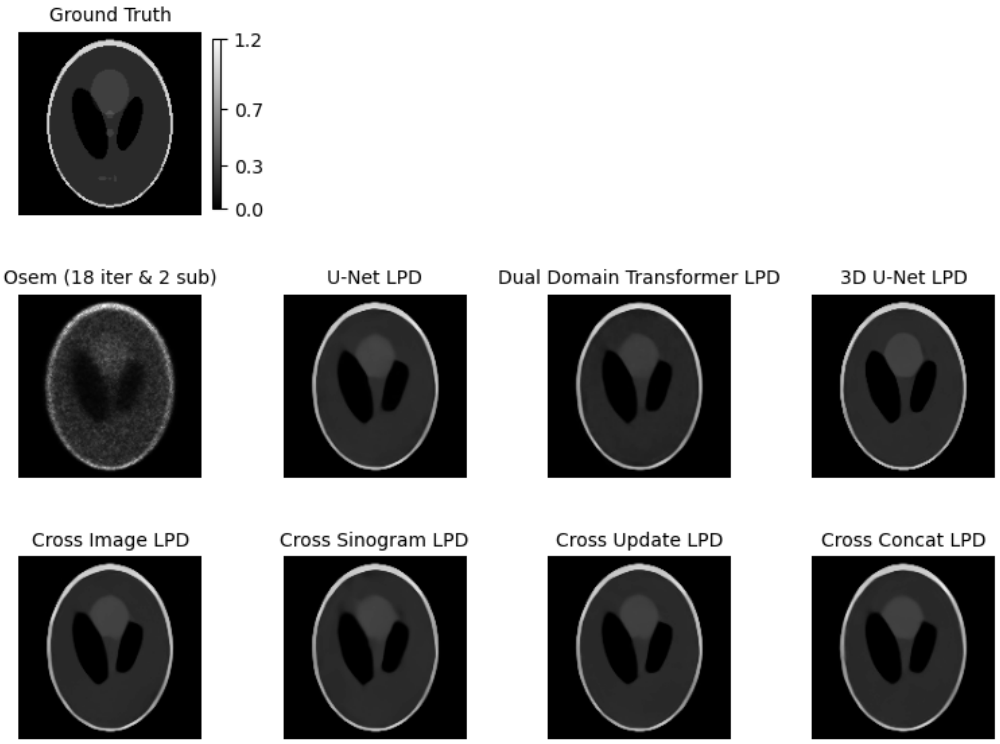

Figure 4.4. Reconstructed Shepp-Logan phantoms

4.4.1 Noise Robustness

Tests were performed to evaluate the performance of the model in the presence of different noise levels in the input sinogram. The noise levels, applied to sinograms generated from the Shepp-Logan phantom, ranged from 0.5 to 9.5 in increments of 0.5.

In Figure 4.5, each metric is plotted by a curve representing the average value, while the coloured area indicates the standard deviation.

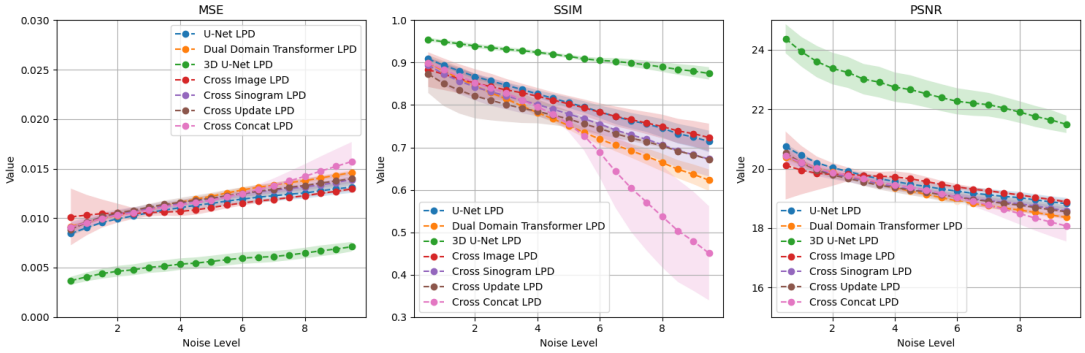

Figure 4.5. Reconstruction performance as the noise level increases

4.5 Pre-clinical Test Data

The central slice of the mouse-like phantoms reconstructed by each model are presented in Figure 4.6 (sagittal view). The MiniPET-3 sinogram used for these reconstructions was taken from the 40-minute acquisition, to be comparable with the reference (first row in the Figure) that was created from the full 60-minutes acquisition.

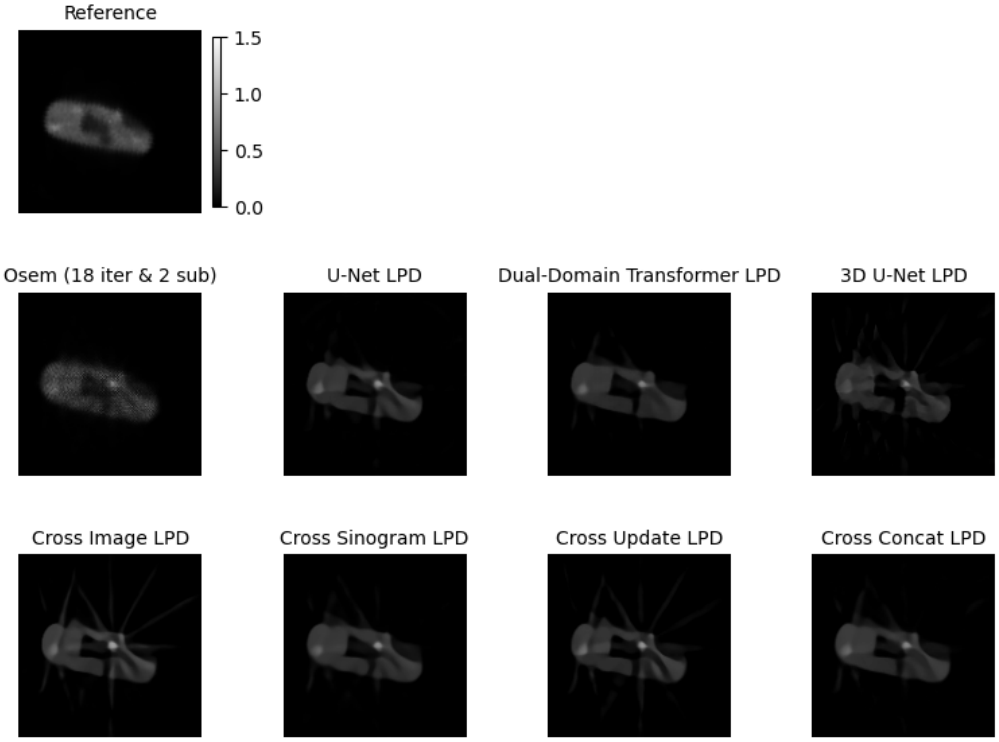

Figure 4.6. Reconstructed mouse-like phantoms (sagittal view)

4.6 Reconstruction Steps

Given the iterative nature of the LPD algorithm, it is possible to systematically examine how each sequential step contributes to and influences the final reconstruction, providing insight into the inner workings of the algorithm. It is particularly interesting to analyse the role of CABs in the reconstruction process, especially when compared to architectures that do not utilise these components.

The outputs of each block are shown in Figure 4.7 for the architectures without Cross-Attention Blocks (CABs) and Figures 4.8 and 4.9 for the ones with CABs. For simplicity, blocks belonging to the two different domains are connected directly, omitting the forward/back projection operators and the residual connections.

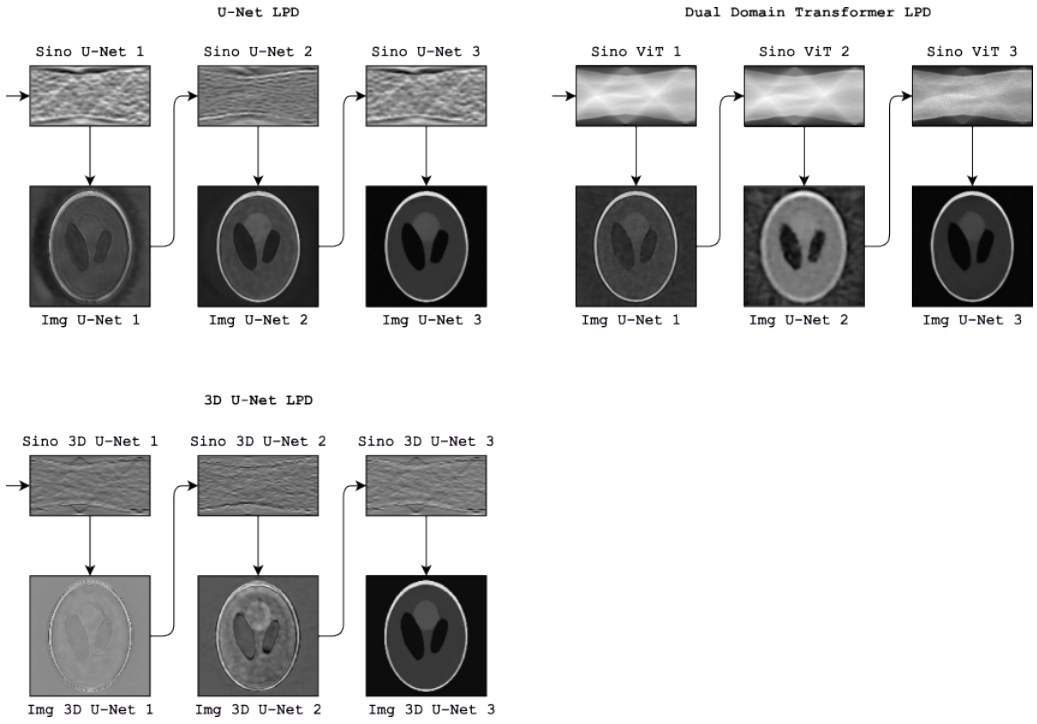

Figure 4.7. The reconstruction steps in the architectures without CABs

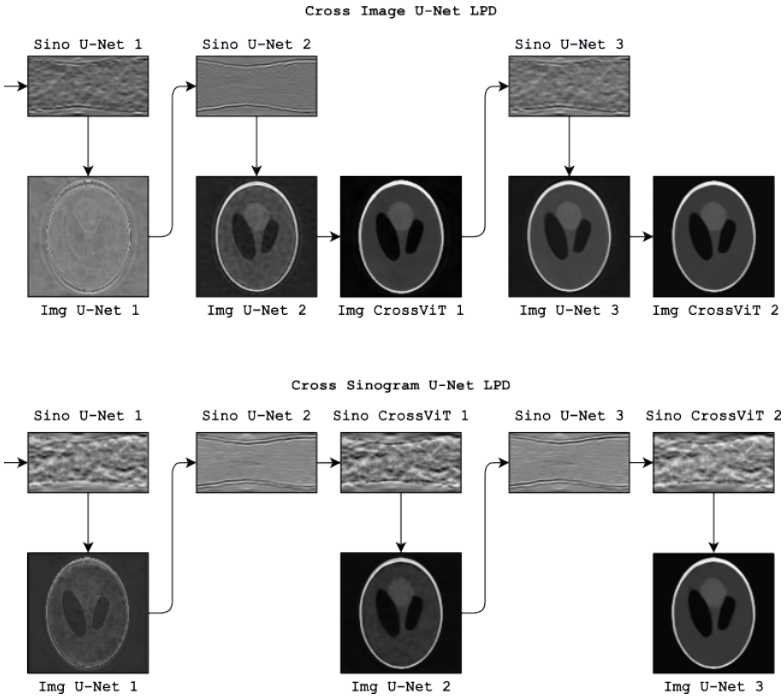

Figure 4.8. The reconstruction steps in the architectures with CABs in one of the two domains

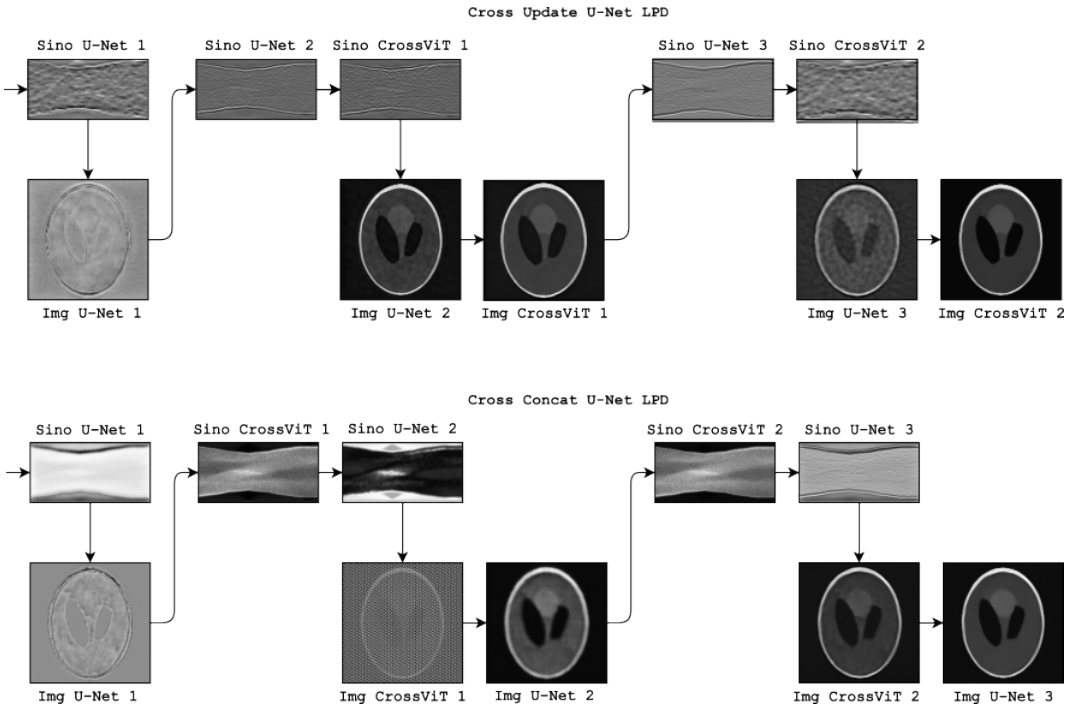

Figure 4.9. The reconstruction steps in the architectures with CABs in both domains

This page is intentionally left blank.

5 Discussion

This chapter analyses and interprets the results presented in the previous chapter, providing a deeper context and meaning. The last part explores current limitations and technical aspects.

5.1 Training Strategy Effectiveness

The choice of the optimiser and learning rate scheduler and associated hyperparameters was dictated by current best practices and was effective for training the proposed models. It is beyond this project's scope to discuss the reasons for this, as the optimiser and the learning rate scheduler have no theoretical connection to the specific reconstruction problem but are related to deep learning in general.

The maximum number of epochs was set at 200 to have enough time before the final project deadline to train each model 3 times, ensuring a basis of statistical significance in the results. This limitation does not guarantee complete convergence of all proposed models, as evidenced in Table 4.1, where the number of epochs is rarely less than 200 epochs, showing that the early stopper has not yet intervened and the model is continuing to improve. However, it is sufficient to achieve high-quality reconstructions.

The early stopping mechanism was activated only after 100 epochs, allowing enough time for the model to stabilise, especially given the learning rate variations introduced by the OneCycleLR scheduler. While OneCycleLR's strategy of increasing the learning rate in the initial epochs helped to effectively explore the solution space, it frequently induced a temporary performance decline that required several epochs to recover from.

5.1.1 Synthetic Training Data

Synthetic data was used because no datasets with enough real PET sinograms to train a DNN were found. In addition, clinical data sets cannot provide the ground truth against which to compare the models' outputs.

In previous works [6], [5], random ellipsoids were used as synthetic objects for the dataset. This choice was certainly the best to obtain a good reconstruction of the Shepp-Logan phantom as shown in Table 4.2. This was expected because the Shepp-Logan phantom is just a set of the same ellipsoids that the entire dataset is composed of.

However, as shown by the arrows in Figure 5.1, it generated a series of artefacts in reconstructing pre-clinical data. In particular, the yellow arrows highlight some rays coming from the central emission point and the blue arrows reveal two areas with a lower intensity which erroneously suggest the presence of some cavities.

Using a different dataset based on random shapes also introduced some artefacts. The red arrows indicate cavities on the outer border of the mouse-like phantom that are not present in the 3D-printed model, shown in Figure 3.8. Nevertheless, this second reconstruction shows less evident emission rays and the surfaces are generally more uniform.

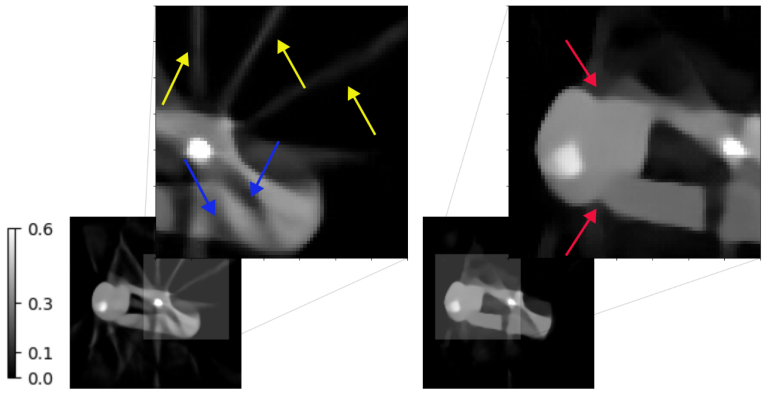

Figure 5.1. Artefacts in the mouse-like reconstructions using different types of dataset

To minimise these types of artefacts while preserving the strengths of both reconstructions, a mixed dataset of random ellipsoids and random shapes was used. This approach proved effective, as shown by the reconstruction in Figure 5.2, achieving a balanced trade-off between the best characteristics of reconstructions from each homogeneous dataset, while attenuating the artefacts discussed above.

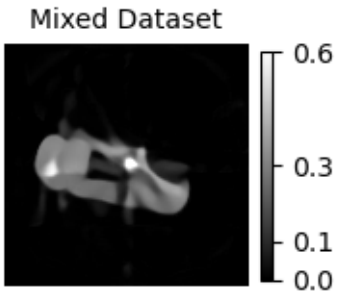

Figure 5.2. Mouse-like reconstruction using the mixed dataset

5.1.2 Loss Function

A Synthmorph-inspired data generator was developed not only to improve the generalisation capability of the models but also to allow the creation of label maps for the regions that make up the objects. These labels were considered important to guide the model in reconstructing the region shapes, beyond the intensity values of individual pixels. Common loss functions, such as the MSE, penalise pixel-level discrepancies without considering objects' morphological characteristics.

This gave rise to the idea of creating a mixed loss function, that incorporated the Mean Squared Error to reduce disparities between pixel intensities and the Dice score to minimise differences between shapes. These two metrics are different because the MSE applies a greater penalty to pixels with a greater intensity difference, while the Dice Score assigns equal importance to each pixel, classifying them as correct or incorrect based on their label.

To allow a meaningful comparison between these two losses, the Mean Squared Error was scaled by a factor of 100, as it was consistently two orders of magnitude lower than the Dice Score upon convergence. Figure 5.3 illustrates the evolution of both metrics, with the left subplot showing the raw values on a logarithmic y-axis and the right subplot showing the values after scaling. From the Figure, it can be seen that this strategy is effective in preventing one metric from dominating the other.

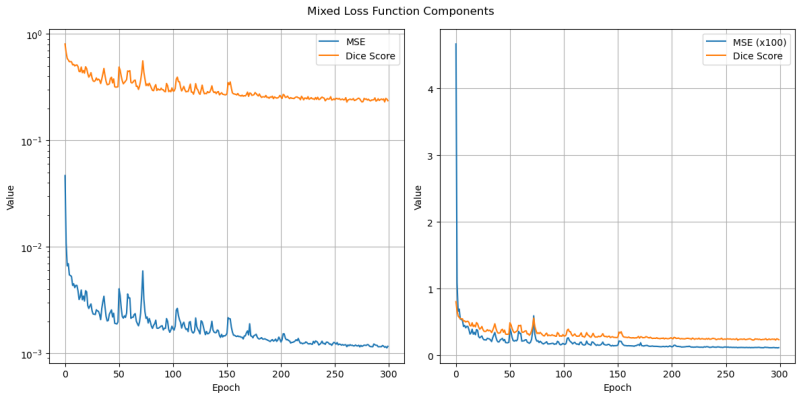

Figure 5.3. The trend of the two components of the mixed loss function before (left) and after scaling the MSE (right)

The application of this strategy has a positive impact on the performance of the Shepp-Logan phantom reconstruction, as shown in Table 4.3 since it helps to remove noise. However, the hoped-for improvement in the fidelity of the shape reconstruction is not evident, as indicated by the SSIM in Table 4.3 and by the visual inspection of Figure 4.3.

Figure 5.4 shows a label reconstructed from an unseen training set image. The green arrows show some slight improvements in shape reconstruction; however, this improvement is not substantial, confirming the result obtained on the reconstruction of the Shepp-Logan phantom.

Figure 5.4. Example of a label reconstructed from models trained with different loss functions

This behaviour probably results from the fact that the models do not directly produce the reconstructed image labels; instead, the labels must be generated

using the segmentation process described in 3.3.1. By exploiting intensity values to reconstruct the labels, that approach creates a dependency between the Dice score and the pixel intensities, making the Dice Score indirectly similar to an error distance metric. This can also be seen from the similarity between the trends of the Dice Score and Mean Squared Error in the graphs in Figure 5.3.

If a reconstructed pixel has an intensity close to its original intensity, it will be assigned to the correct label and will produce a Dice score of 1. Therefore, reconstructed images with low values of MSE or MAE can obtain an accurate reconstruction of the labels and thus a high Dice score, regardless of the loss function used for training.

Nevertheless, it is clear that applying a loss function not only to the entire image but also to each labelled region, as done above with the Dice score, improves the model's denoising and generalising capabilities.

5.2 Motivations behind the Architectures

The initial idea of adding a Cross-Attention mechanism to the LPD algorithm was based on finding a way to combine information between the sinogram and the image space, leveraging the LPD's distinctive characteristic of working on both these spaces. While this approach may prove effective in other double domain architectures, in the LPD algorithm the image data does not come from a separate source, but it is generated from the input sinogram using a back projection. This makes the image space deeply related to the sinogram space and it is difficult to imagine what information could be learnt from combining the image and sinogram domain, other than the back projection itself.

For these reasons, this project was focused on implementing Cross-Attention Blocks (CABs) between data belonging to the same domain (the inputs of CABs are only images or only sinograms) and to different intermediate steps of the LPD algorithm.

5.3 Reconstruction Quality and Performance

The use of Cross-Attention Blocks (CABs) within the LPD architecture does not seem to provide any meaningful improvement, either in terms of metrics or visual quality. As for Cross Image and Cross Sinogram U-Net LPD, this can be attributed to the fact that the integration of current data with a previous version results in a slight step backward in the reconstruction process.

As for Cross Update and Cross Concatenation U-Net LPD, CABs appear to be a viable alternative to traditional additions and concatenations in residual connections. However, even in this case, no improvement in the quality of the final reconstruction is evident.

Furthermore, as can be noticed by comparing Figures 4.8 and 4.9 with Figure 4.7, these blocks seem to mainly redistribute the reconstruction process over more parameters rather than to improve its overall effectiveness.

5.3.1 Synthetic Test Data

Table 4.4 shows that the 3D U-Net LPD architecture provides the best reconstruction of the Shepp-Logan phantom by a substantial margin. This superiority is attributed to the ability of the 3D convolutions to utilise the information contained in the depth dimension effectively. This feature is particularly advantageous since real sinograms are three-dimensional objects that contain information along all dimensions.

In contrast, the other reconstruction methods based on the LPD architecture provide comparable performance in terms of metrics and, even visually, it is difficult to distinguish substantial differences between them.

Performance decreases almost linearly with increasing noise for most architectures, except for the Cross Concatenation U-Net LPD whose reconstruction quality drastically decreases after a noise level of 5 as shown by Figure 4.5. This behaviour can be attributed to the critical role played by the CABs in this architecture, which are responsible for integrating information from multiple inputs into a single representation. Since the architecture wasn't trained on a dataset with noise levels above 1.2, the CABs struggle to effectively combine inputs at higher noise levels, probably reintroducing noise from previous steps into the current step.

5.3.2 Pre-clinical Test Data

The mouse-like phantom reconstructions shown in Figure 4.6 do not differ significantly, making it difficult to determine the best one. Strong candidates include the Dual Domain Transformer LPD and the Cross Concatenation U-Net LPD, which appear to be the most effective in mitigating the visual artefacts, already discussed in Section 5.1.1.

The 3D U-Net LPD, whose reconstruction of the Shepp-Logan phantom was the best, is the worst with the pre-clinical test data, as illustrated by Figure 4.6. This difference is probably attributable to the imperfect simulation of the clinical MiniPET-3 geometry used to generate sinograms for the training data.

5.4 Limitations

This project represents experimental research that addresses cutting-edge topics not completely explored in the current literature. The integrations attempted here represent only a subset of the potential applications of a Cross-Attention mechanism in LPD architecture, leaving significant room for further exploration. Future work should analyse the role of CABs in simpler architectures, to better understand their impact without being overwhelmed by an overstructure.

External validation should also be conducted on a wider range of test objects to reduce potential biases. Currently, results can be biased by the use of the Shepp-Logan phantom as the only object used for the metric evaluation. Expanding the range of test objects would allow for a more comprehensive evaluation of the algorithm's performance.

Furthermore, although the metrics used (MSE, SSIM and PSNR) are widely

accepted in the literature, they fail to fully capture the reconstruction quality. Incorporating perceptual metrics, carefully tuned with images similar to the test objects, could provide a more accurate assessment of the reconstruction quality. If such metrics were designed as no-reference measures, they could also allow for a quantitative assessment of experimental reconstructions, further improving the robustness of the analysis.

5.5 Future Research

Future research should focus on adapting these algorithms to real clinical data, addressing the challenges associated with scalability and practical implementation. Currently, these architectures encounter limitations in handling large-scale and complex datasets, typically found in clinical settings. Efforts should focus on developing more scalable variants of LPD-based models, optimising them for efficient training and inference on real data.

5.6 Technical issues

Working with such complex architectures involves certain technical problems that have to be addressed. First of all, there is a huge use of VRAM (Table 4.1), which makes it possible to carry out the training only on high-end professional GPUs. This requirement is due to the large number of intermediate steps PyTorch has to keep track of during the forward step to efficiently compute the gradients during the back-propagation.

The memory utilisation can be significantly reduced by implementing a technique called ”checkpointing” [50]. Checkpointing re-performs the forward pass for the intermediate steps during the back-propagation, eliminating the need to store them in memory at the cost of an increased computational burden.

Although implemented, checkpointing was not used in this project, as the system had adequate VRAM capacity for the proposed architectures. However, it is a very useful tool in cases where the system does not have sufficient VRAM.

Another problem is related to NaN values that may occur due to the complexity of the architectures and the use of mixed precision [51]. The latter is a known problem of the PyTorch package, which has not yet been solved and seems to occur particularly during the calculation of Attention in transformer-based architectures. In the project, this issue was mitigated by calculating the Attention in full precision and by employing gradient norm clipping.

6 Conclusion

The primary objectives of this project were two: developing a new training strategy to improve the generalisation power of deep learning models and introducing CABs in the LPD architecture. To achieve the first objective, a new mixed synthetic dataset and a new loss function were implemented, reducing the number of artefacts in the pre-clinical test data and improving the reconstruction quality of the synthetic test data. To complete the second objective, four modified LPD architectures, implementing CABs in different ways, were proposed. The results obtained by these architectures were quantitatively and qualitatively comparable to the ones obtained by the architectures without CABs.

Given these findings, it can be concluded that employing a mixed dataset enhances the generalisation capabilities of the DNNs by exposing them to a diverse range of shapes. It can be also said that using CABs in the LPD architecture does not meaningfully improve the final reconstruction of the test objects.

The proposed training strategy and the exploration of the role of CABs in advanced deep learning architectures, such as the LPD one, represent a small advance in addressing the reconstruction problem in PET imaging. Although no improvements have been achieved compared to previous research, this study has deepened previously unexplored concepts that may be of importance for future developments. In particular, CABs are interesting tools that hold good premises in the integration of information from different sources.

Reconstruction in medical imaging remains a challenging and critical area of research. The use of state-of-the-art reconstruction algorithms together with deep learning methods could significantly improve the interpretation of clinical data, with profound implications for both diagnostics and research.

This page is intentionally left blank.

References

[1] "Healthcare resource statistics - technical resources and medical technology." (2024), [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Healthcare_resource_statistics_-_technical_resources_and_medical_technology (visited on 09/22/2024).

[2] A. J. Reader and H. Zaidi, "Advances in PET image reconstruction," PET clinics, vol. 2, no. 2, pp. 173–190, Apr. 2007, ISSN: 1556-8598.

[3] A. J. Reader and B. Pan, "AI for PET image reconstruction," The British Journal of Radiology, vol. 96, no. 1150, p. 20 230 292, Oct. 2023, ISSN: 0007-1285.

[4] J. Adler and O. Öktem, "Learned primal-dual reconstruction," IEEE Transactions on Medical Imaging, vol. 37, no. 6, pp. 1322–1332, Jun. 2018, Conference Name: IEEE Transactions on Medical Imaging, ISSN: 1558-254X.

[5] A. Guazzo and M. Colarieti-Tosti, "Learned primal dual reconstruction for PET," Journal of Imaging, vol. 7, no. 12, p. 248, Nov. 24, 2021, ISSN: 2313-433X.

[6] A. Adelöw, H. R. Kanan, A. Guazzo, and M. Colarieti-Tosti, "Learned primal dual reconstruction with dual-domain transformers for PET," in 2024 IEEE Nuclear Science Symposium (NSS), Medical Imaging Conference (MIC) and Room Temperature Semiconductor Detector Conference (RTSD), ISSN: 2577-0829, Oct. 2024, pp. 1–1.

[7] J. Czernin, M. Allen-Auerbach, D. Nathanson, and K. Herrmann, "PET/CT in oncology: Current status and perspectives," Current Radiology Reports, vol. 1, no. 3, pp. 177–190, Sep. 1, 2013, ISSN: 2167-4825.

[8] A. Gallamini, C. Zwarthoed, and A. Borra, "Positron emission tomography (PET) in oncology," Cancers, vol. 6, no. 4, pp. 1821–1889, Sep. 29, 2014, ISSN: 2072-6694.

[9] S. G. Nerella, P. Singh, T. Sanam, and C. S. Digwal, "PET molecular imaging in drug development: The imaging and chemistry perspective," Frontiers in Medicine, vol. 9, Feb. 28, 2022, Publisher: Frontiers, ISSN: 2296-858X.

[10] N. A. Karakatsanis, E. Fokou, and C. Tsoumpas, "Dosage optimization in positron emission tomography: State-of-the-art methods and future prospects," American Journal of Nuclear Medicine and Molecular Imaging, vol. 5, no. 5, pp. 527–547, Oct. 12, 2015, ISSN: 2160-8407.

[11] H. N. Wagner, "A brief history of positron emission tomography (PET)," Seminars in Nuclear Medicine, The Coming Age of Pet (Part 1), vol. 28, no. 3, pp. 213–220, Jul. 1, 1998, ISSN: 0001-2998.

[12] D. Bailey, J. Humm, A. Todd-Pokropek, and A. van Aswegen, Nuclear Medicine Physics (Non-serial Publications). Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2015, ISBN: 978-92-0-143810-2.

[13] Planning a Clinical PET Centre (Human Health Series 11). Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2010, ISBN: 978-92-0-104610-9.

[14] S. R. Meikle and R. D. Badawi, "Quantitative techniques in PET," in Positron Emission Tomography: Basic Sciences, D. L. Bailey, D. W. Townsend, P. E. Valk, and M. N. Maisey, Eds., London: Springer, 2005, pp. 93–126, ISBN: 978-1-84628-007-8.

[15] L. M. Carter, A. Leon Kesner, E. C. Pratt, et al., "The impact of positron range on PET resolution, evaluated with phantoms and PHITS monte carlo simulations for conventional and non-conventional radionuclides," Molecular imaging and biology, vol. 22, no. 1, pp. 73–84, Feb. 2020, ISSN: 1536-1632.

[16] M. Defrise and P. Kinahan, "Data acquisition and image reconstruction for 3d PET," in The Theory and Practice of 3D PET, B. Bendriem and D. W. Townsend, Eds., Dordrecht: Springer Netherlands, 1998, pp. 11–53, ISBN: 978-94-017-3475-2.

[17] T. G. Feeman, "X-rays," in The Mathematics of Medical Imaging: A Beginner's Guide, T. G. Feeman, Ed., Cham: Springer International Publishing, 2015, pp. 1–11, ISBN: 978-3-319-22665-1.

[18] S. Tong, A. M. Alessio, and P. E. Kinahan, "Image reconstruction for PET/CT scanners: Past achievements and future challenges," Imaging in medicine, vol. 2, no. 5, pp. 529–545, Oct. 1, 2010, ISSN: 1755-5191.

[19] N. Chetih and Z. Messali, "Tomographic image reconstruction using filtered back projection (FBP) and algebraic reconstruction technique (ART)," in 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT), May 2015, pp. 1–6.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, no. 1, pp. 1–38, 1977, Publisher: [Royal Statistical Society, Oxford University Press], ISSN: 0035-9246.

[21] H. Hudson and R. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," IEEE Transactions on Medical Imaging, vol. 13, no. 4, pp. 601–609, Dec. 1994, Conference Name: IEEE Transactions on Medical Imaging, ISSN: 1558-254X.

[22] J. J. Vaquero and P. Kinahan, "Positron emission tomography: Current challenges and opportunities for technological advances in clinical and preclinical imaging systems," Annual review of biomedical engineering, vol. 17, pp. 385–414, 2015, ISSN: 1523-9829.

[23] H. R. Kanan, A. Adelöw, and M. Colarieti-Tosti, "Cross-domain reconstruction network incorporating sinogram sinusoidal-structure transformer denoiser and UNet for low-dose/low-count sinograms," 2024.

[24] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need, Aug. 1, 2023. arXiv: 1706.03762[cs].

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., An image is worth 16x16 words: Transformers for image recognition at scale, Jun. 3, 2021. arXiv: 2010.11929[cs].

[26] J. Maurício, I. Domingues, and J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review," Applied Sciences, vol. 13, no. 9, p. 5521, Jan. 2023, Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417.

[27] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, Dec. 10, 2015. arXiv: 1512.03385.

[28] P. Li, J. Gu, J. Kuen, et al. "SelfDoc: Self-supervised document representation learning," arXiv.org. (Jun. 7, 2021), [Online]. Available: https://arxiv.org/abs/2106.03331v1 (visited on 09/11/2024).

[29] C.-F. Chen, Q. Fan, and R. Panda, CrossViT: Cross-attention multi-scale vision transformer for image classification, Aug. 22, 2021. arXiv: 2103.14899[cs].

[30] O. Petit, N. Thome, C. Rambour, and L. Soler, U-net transformer: Self and cross attention for medical image segmentation, Mar. 12, 2021. arXiv: 2103.06104.

[31] M. Hoffmann, B. Billot, J. E. Iglesias, B. Fischl, and A. V. Dalca, "Learning mri contrast-agnostic registration," in 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), ISSN: 1945-8452, Apr. 2021, pp. 899–903.

[32] M. Hoffmann, B. Billot, D. N. Greve, J. E. Iglesias, B. Fischl, and A. V. Dalca, "SynthMorph: Learning contrast-invariant registration without acquired images," IEEE Transactions on Medical Imaging, vol. 41, no. 3, pp. 543–558, Mar. 2022, Conference Name: IEEE Transactions on Medical Imaging, ISSN: 1558-254X.

[33] A. K. Krizsan, I. Lajtos, M. Dahlbom, et al., "A promising future: Comparable imaging capability of MRI-compatible silicon photomultiplier and conventional photosensor preclinical PET systems," Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine, vol. 56, no. 12, pp. 1948–1953, Dec. 2015, ISSN: 1535-5667.

[34] M. Conti and L. Eriksson, "Physics of pure and non-pure positron emitters for PET: A review and a discussion," EJNMMI Physics, vol. 3, no. 1, pp. 1–17, Dec. 2016, Number: 1 Publisher: SpringerOpen, ISSN: 2197-7364.

[35] Y. Vardi, L. A. Shepp, and L. Kaufman, "A statistical model for positron emission tomography," Journal of the American Statistical Association, vol. 80, no. 389, pp. 8–20, Mar. 1, 1985, ISSN: 0162-1459.

[36] L. A. Shepp and B. F. Logan, "The fourier reconstruction of a head section," IEEE Transactions on Nuclear Science, vol. 21, no. 3, pp. 21–43, Jun. 1974, Conference Name: IEEE Transactions on Nuclear Science, ISSN: 1558-1578.

[37] C. G. Koay, J. E. Sarlls, and E. Özarslan, "Three-dimensional analytical magnetic resonance imaging phantom in the fourier domain," Magnetic Resonance in Medicine, vol. 58, no. 2, pp. 430–436, 2007, ISSN: 1522-2594.

[38] G. Schramm and K. Thielemans, "PARALLELPROJ—an open-source framework for fast calculation of projections in tomography," Frontiers in Nuclear Medicine, vol. 3, Jan. 8, 2024, Publisher: Frontiers, ISSN: 2673-8880.

[39] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation, May 18, 2015. arXiv: 1505.04597[cs].

[40] X. Xie and M. Yang, "USCT-UNet: Rethinking the semantic gap in u-net network from u-shaped skip connections with multichannel fusion transformer," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 32, pp. 3782–3793, 2024, Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering, ISSN: 1558-0210.

[41] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, Jan. 4, 2019. arXiv: 1711.05101.

[42] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, Jan. 30, 2017. arXiv: 1412.6980.

[43] L. N. Smith and N. Topin, Super-convergence: Very fast training of neural networks using large learning rates, May 17, 2018. arXiv: 1708.07120.

[44] "Train with mixed precision," NVIDIA Docs. (2023), [Online]. Available: https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html (visited on 10/07/2024).

[45] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr. 2004, Conference Name: IEEE Transactions on Image Processing, ISSN: 1941-0042.

[46] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study," Journal of Computer and Communications, vol. 7, no. 3, pp. 8–18, Mar. 4, 2019, Number: 3 Publisher: Scientific Research Publishing.

[47] J. Ansel, E. Yang, H. He, et al., "PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation," in 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24), ACM, Apr. 2024.

[48] J. Adler, H. Kohr, and O. Öktem, Operator discretization library (ODL), version 1.0.0.dev0, 2024.

[49] N. S. Detlefsen, J. Borovec, J. Schock, et al., "TorchMetrics - measuring reproducibility in PyTorch," Journal of Open Source Software, vol. 7, no. 70, p. 4101, Feb. 11, 2022, ISSN: 2475-9066.

[50] "Torch.utils.checkpoint — PyTorch 2.5 documentation." (2024), [Online]. Available: https://pytorch.org/docs/stable/checkpoint.html (visited on 11/18/2024).

[51] "Mixed precision causes NaN loss · issue #40497 · pytorch/pytorch," GitHub. (2024), [Online]. Available: https://github.com/pytorch/pytorch/issues/40497 (visited on 11/18/2024).

This page is intentionally left blank.

TRITA-CBH-GRU-2025:010
Stockholm, Sweden 2025

www.kth.se