Degree Project in Mathematical Statistics

Second cycle, 30 credits

Transformer-Based Learned Primal-Dual Reconstruction for PET

ANTON ADELÖW

Transformer-Based Learned Primal-Dual Reconstruction for PET

ANTON ADELÖW

Master's Programme, Applied and Computational Mathematics, 120 credits
Date: May 24, 2024

Supervisors: Massimiliano Colarieti-Tosti, Hamidreza Rashidy Kanan
Examiner: Ozan Öktem

Swedish title: Transformer-baserad lärd primal-dual rekonstruktion för PET

Abstract

Positron Emission Tomography (PET) is a medical imaging technique that involves injecting a short-lived radioactive tracer, chemically bound to a biologically active molecule, into a subject in order to measure cellular metabolic activity within bodily tissues. By revealing abnormal metabolic activity, PET serves as an important method in cancer diagnosis. In a realistic setting, there are multiple sources of noise affecting the measurements. The Learned Primal-Dual (LPD) reconstruction algorithm, proposed by Adler and Öktem, utilizes Convolutional Neural Networks (CNNs) in the unrolling to achieve state of the art results for image reconstruction. The CNNs in the LPD algorithm impose a locality assumption on features in both the image and scanner data, which could potentially lead to inaccuracies. The Transformer architecture could offer advantages over CNNs for this particular problem, due to its ability to capture global dependencies. Three Transformer-based architectures were incorporated into the LPD algorithm, compared against a baseline model, and evaluated on synthetic and experimental data from a preclinical system. The results show promise in Transformer-based LPD algorithms, which can provide better reconstructions than previously proposed CNN-based methods, based on three different figures of merit. Additionally, a synthetic data generation process designed to mimic a preclinical system is introduced. The results indicate effective transfer learning from synthetic to preclinical data.

Keywords

Positron Emission Tomography, Inverse Problems, Deep learning, Learned Primal-Dual Reconstruction, Vision Transformer, Transfer Learning

Sammanfattning

Positronemissionstomografi (PET) är en medicinsk avbildningsteknik där ett kortlivat radioaktiv spårämne, kemiskt bundet till en biologiskt aktiv molekyl, injiceras för att mäta cellulär metabolisk aktivitet inom kroppsvävnader. Genom att avslöja onormal metabolisk aktivitet fungerar PET som en viktig metod för cancerdiagnos. I en realistisk miljö finns det flera källor till brus som påverkar mätningarna. Den Lärda Primal-Dual (LPD) rekonstruktionsalgoritmen, föreslagen av Adler och Öktem, använder faltningsnätverk i utrullningen för att uppnå ett av dom bästa resultaten hitills. Faltningsnätverken i LPD-algoritmen inför ett lokalitetsantagande i både bild och skannerdata, vilket potentiellt kan leda till felaktigheter. Transformer-arkitekturen kan erbjuda fördelar över faltningsnätverk för detta specifika problem, på grund av dess förmåga att fånga globala beroenden. Tre Transformer-baserade arkitekturer implementerades i LPD-algoritmen, jämfördes med en tidigare föreslagen faltningsbaserad modell och utvärderades på syntetisk och verklig data. Resultaten är lovande för Transformer-baserade LPD algoritmer som kan prestera bättre än tidigare föreslagna modeller. En metod för syntetisk datagenering designad för att efterlikna egenskaper hos ett prekliniskt system presenteras. Resultaten visar på effektiv extrapolering från den syntetiska till den prekliniska datan.

Nyckelord

Positronemissionstomografi, Inversa Problem, Djupinlärning, Lärd Primal-Dual Rekonstruktion, Visions-Transformer

Acknowledgments

I would like to extend my gratitude to Massimiliano Colarieti-Tosti (Mamo), Hamidreza Rashidy Kanan and Alessandro Guazzo for their support and our insightful discussions throughout the project. I am particularly grateful to Mamo for his guidance and assistance with scanner related matters at KTH Flemingsberg. I would also like to acknowledge Hamidrezas crucial contributions to the development of the architectures discussed. Lastly, my sincere thanks go to Alessandro, whose ideas brought a fresh perspective to our discussions.

Stockholm, May 2024
Anton Adelöw

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

CAB	Cross-Attention Block
CNN	Convolutional Neural Network
CT	Computed Tomography
LOR	Line of Response
LPD	Learned Primal-Dual
MLEM	Maximum-Likelihood Expectation-Maximization
MLP	Multilayer perceptron
MSE	Mean Squared Error
PET	Positron Emission Tomography
PSNR	Peak Signal-to-Noise Ratio
SLP	Shepp-Logan Phantom
SSIM	Structural Similarity Index Measure

Chapter 1

Introduction

Positron Emission Tomography (PET) is a medical imaging technique that involves injecting a short-lived radioactive tracer, chemically bound to a biologically active molecule, into a subject in order to measure cellular metabolic activity within bodily tissues. By revealing abnormal metabolic activity, PET serves as an important method in cancer diagnosis. After a positron decays and is subsequently annihilated with an electron, a scanner detects in coincidence the resulting back-to-back emitted gamma rays. Based on measurements collected over a period of time, the task is to reconstruct an image from scanner data. The problem inherently has two domains, the image (primal) domain and the data (dual) domain.

In a realistic setting, there are multiple sources of noise affecting the measurements. Various approaches have been proposed to reconstruct an image from the noisy scanner data. With the advent of deep learning, the reconstruction problem has been cast into the supervised learning domain. Algorithm unrolling is a method that combines a conventional iterative algorithm with a deep neural network by mapping each of the (finite) iterations to a single layer and stacking these layers. By utilizing the inherent structure of the problem, unrolled reconstruction algorithms can reduce the amount of data needed for training [1].

The Learned Primal-Dual (LPD) reconstruction algorithm, presented by Adler and Öktem in [2], utilizes Convolutional Neural Networks (CNNs) in the unrolling to achieve state of the art results. The CNNs in the LPD algorithm could impose a locality assumption on features in both the image and scanner data, which could potentially lead to inaccuracies, particularly in the dual domain.

Due to its ability to capture non-local features, the Transformer

architecture may offer advantages over CNNs for this particular problem. This project involves implementing Transformer-based architectures and integrating them into the LPD algorithm, thereby either replacing or aiding the convolutional layers in the primal and dual iterates.

1.1 Related Works

Perhaps the most important related work for this thesis is the one mentioned in the introduction, Learned Primal-Dual Reconstruction by Adler and Öktem [2], which introduces the basic architecture of which this project is based upon. Furthermore, the LPD algorithm is evaluated and modified for PET in [3], by Guazzo and Colarieti-Tosti, which serves as a starting point for this project.

Also of great importance is the introduction of the Transformer Architecture, in [4], by Vaswani *et al.*, for natural language processing. In [5], the Transformer is applied to image data for the first time, laying the foundation for Transformer-based image processing. Further improvements to Vision Transformer has been made since, such as the Swin Transformer [6], which uses shifted windows, instead of simply patching the image like in [5], to achieve improved results on classification and prediction, and FasterViT [7] which utilizes CNNs to increase throughput compared to previously mentioned architectures. Closer to the problem adressed in this thesis, the Uformer [8] and the Restormer [9] draws inspiration from the U-Net architecture, but employs Transformers in the place of the CNNs for image restoration.

Several Transformer-based architectures have been proposed for medical image denoising and reconstruction. In [10], Vision Transformers are used to denoise sinograms, by patching using the rows of the sinogram. The Hformer [11] employs a Uformer-like architecture with lightweight self-attentive modules to denoise medical image images.

1.2 Research Methodology

The research methodology in this thesis resembles the methodology typically used in supervised learning. General architectures for Transformer-based LPD models will be proposed based on the problem. Models using the architectures are fit by utilizing the training data, and then evaluated on two test sets. A few different metrics for image reconstruction quality are used for evaluation. The test sets provide an indication of how the models generalize to unseen data.

1.3 Delimitations

The project will concentrate on the modeling and analysis aspects. Thus, questions regarding the practical application of models similar to the ones proposed in this thesis are left to practitioners within the field. Additionally, better results than what is presented in this thesis are likely achievable by tuning parameters and/or increasing the depth of the presented architectures. Though important, tuning parameters to maximize image reconstruction quality is a compute intensive task, and should be explored in future works.

Chapter 2

Background

The two most important fields in this thesis are PET reconstruction and deep learning. In order to design a successful reconstruction algorithm, the original problem of PET reconstruction must be understood in detail, as must the use of the Transformer architecture in image processing.

2.1 PET Imaging

A positron emitting radio tracer is injected into the subject. The decay process, and thus the positron emission, can be modeled using the Poisson distribution. When the positron comes into contact with an electron, they annihilate and two gamma rays are emitted in opposite directions. This is described in equation 2.1.

$$e^+ + e^- \rightarrow \gamma + \gamma \tag{2.1}$$

The two gamma rays are released in approximately opposite directions. The scanner, located around the subject, detects the gamma rays. Two gamma rays, detected within a coincidence window, are assumed to originate from the same emission. Based on this information, it is assumed that the annihilation occurred along the linear segment connecting the two detector cells that registered the gamma rays. These linear segments are known as Line of Responses (LORs). A visualisation of a basic PET-setup can be seen in Figure 2.1.

There are a few factors that introduce noise into the scanner measurements, complicating the reconstruction problem. If the photon of one (or both) of the gamma rays interacts with charged particles in the matter, it can scatter, which results in a change of its path and an incorrectly detected LOR. Additionally,

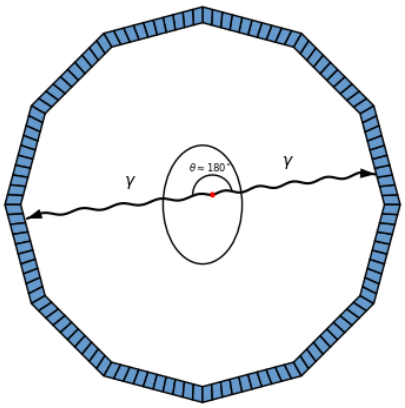


Figure 2.1: A basic PET-setup. A positron emission occurs at the red dot within the elliptic subject. Two gamma rays propagate through space and are detected by the scanner.

if two positrons are emitted almost simultaneously so that their coincidence windows overlap, the LORs may be incorrectly assigned. Furthermore, due to electron-positron momentum, there can be a minor angular displacement causing the two emitted gamma rays to not be collinear. The positron also gets displaced a short distance prior annihilation, causing a displacement of the LOR from the point of emission.

2.1.1 Mathematical Modelling

The activity concentration f at point $\mathbf{x}$ is related to the emissions along a LOR as

$$E\left[N_{LOR}\right] \propto \int_{LOR} f(\mathbf{x}) d\mathbf{x},$$

where N_{LOR} is the number of events detected along a LOR in a given time frame and f is the activity distribution. N_{LOR} depends on the frequency of radioactive decay along (close to) the LOR, and so is clearly stochastic. Since the particles decay exponentially, N_{LOR}, measured at time t, depends on the original concentration, t and a rate parameter λ. A common approach is to model the events as a Poisson distributed random variable. To model N_{LOR} in this way may not be entirely accurate, however, due to the infinite support of the probability mass function (see section A.1).

The task is to, based on the data obtained by the scanner, obtain a reconstruction $\hat{f}(\mathbf{x})$ of the activity distribution $f(\mathbf{x})$ that is as close as possible.

However, since we can only measure the values

$$N_{LOR} + \delta, \tag{2.2}$$

where δ is some variable to account for the noise in the measurements, this is not a simple task. Due to the strong presence of noise in PET, this problem is ill-posed.

For a polygon scanner ring, consisting of k detectors, the total number of LORs is $\frac{k(k-l)}{2}$, where l is the number of detector combinations which cannot form a LOR, considering no LORs between detector rings. The LORs can be parameterized by (θ, ρ, r), where θ is the angular displacement, ρ is the radial displacement from the origin in the xy-plane, and r is the z coordinate of the detector ring considered. The parameterization for a single detector ring is visualized in Figure 2.2. By plotting the counts along the LORs, described by

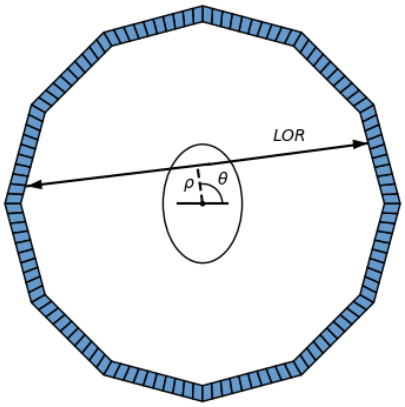


Figure 2.2: A parametrization of the LORs for the scanner geometry.

the angular displacement θ and radial displacement ρ, for a given z, we obtain a sinogram. A sinogram is a visualization of the data collected by the scanner. A sinogram from a slice of the Shepp-Logan Phantom (SLP) [12], a standard test image for reconstruction, can be seen in Figure 2.3.

2.1.2 Forward Projection

We define the ray transform $\mathcal{R}$, applied on the function f as

$$\mathcal{R}f(l) = \int_l f(\mathbf{x}) d\mathbf{x} \tag{2.3}$$

Figure 2.3: Left: A slice of the Shepp-Logan phantom. Right: A sinogram of the slice.

over a line l. For an object in 3D space, let the forward operator $\mathcal{T}$ be the defined as the ray transform applied over a set of LORs, and let g be the data collected by the scanner. Then, we have that

$$g = \mathcal{T}f + \delta \tag{2.4}$$

where δ is a random variable representing the noise.

2.1.3 Back Projection

In the most basic case, to reconstruct an image from scanner data, we can for each point in image space assign an activity value that is the average of the activity detected along all LORs passing through the point. Mathematically, the activity detected along a LOR can be described as a line integral over the activity distribution along the LOR.

This process is known as the back projection. Back projecting does, however, produce blurry and poorly reconstructed images. An example of a back projected slice of the Shepp-Logan phantom can be seen in Figure 2.4.

2.1.4 Problem Formulation

The inverse problem is thus to estimate $f \in X$ from $g \in Y$ (eq. 2.4), where X and Y are Hilbert spaces. By minimizing the negative log likelihood $\mathcal{L}$, the problem can be formulated as

$$\underset{f}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{T}f, g) + \lambda S(f), \tag{2.5}$$

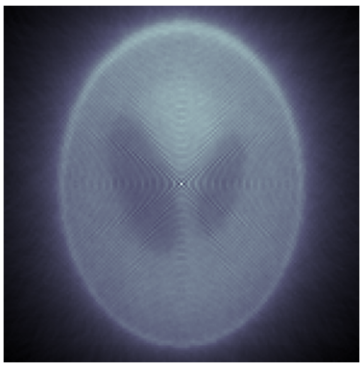


Figure 2.4: Back projection of a slice of the Shepp-Logan phantom, generated with a noise level of 0.3.

where $S(f)$ is a regularization term for inclusion of prior knowledge about f, and λ is a positive constant.

2.2 Deep Learning for Image Reconstruction

In supervised learning, a neural network learns its parameters (weights) by utilizing backpropagation to estimate the gradients, which can be used to update the weights. Mathematically, given a loss function h, a neural network α, and features x_i with corresponding labels y_i, we aim to solve

$$\min_{\theta \in \Theta} \mathbb{E}\left[h(\alpha(x, \theta), y)\right],$$

given some random weight initialization θ_0. The model parameters are updated iteratively

$$\theta_{n+1} = \theta_n - \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta h(\alpha(x_i, \theta_n), y_i),$$

where η is the learning rate, $\nabla_\theta h$ is the gradient estimate, or optimizer, and N is the number of examples considered for each update (batch size). Although out of the scope of this thesis, there are interesting connections between neural networks and stochastic differential equations, see for example [13].

Returning to the reconstruction problem, by utilizing deep neural networks, the task is to find $\mathcal{T}_\theta^\dagger : Y \to X$ such that $\mathcal{T}_\theta^\dagger g \approx f$, where $\theta \in \Theta$ are the model parameters in some parameter space Θ. After settling on an

architecture for $\mathcal{T}_\theta^\dagger$, we seek to optimize the model by learning θ from training data.

2.2.1 Learned Primal Dual Reconstruction

The LPD algorithm is an unrolled algorithm which utilizes CNNs for medical image reconstruction. The unrolling in the LPD algorithm is constructed by iteratively back and forward projecting the data, while processing it in each iteration using deep neural networks. The originally proposed version by Adler and Öktem [2] for Computed Tomography (CT) used 10 iterations and shallow CNNs, while the version proposed by Guazzo and Colarieti-Tosti [3] for PET utilized deeper CNNs in the form of U-nets, but used only 3 iterations in the unrolling.

More formally, let $\Lambda_{\gamma_i}^i : \overbrace{X \times X ... \times X}^{i+1 \text{ times}} \to X$ and $\Xi_{\phi_i}^i : \overbrace{Y \times Y ... \times Y}^{i+1 \text{ times}} \to Y$ be two families of operators, with parameters γ_i and ϕ_i. By iterating over the index i, we form the LPD algorithm by applying $\Lambda_{\gamma_i}^i$, back projecting, applying $\Xi_{\phi_i}^i$ and forward projecting. The LPD algorithm utilizes residual connections in both domains to allow important features to propagate through the network unchanged. The outputs of all previous layers are included as channels in the input to layer i, providing a sort of memory for the algorithm. The LPD algorithm is summarized in Algorithm 1. A visualization of the LPD algorithm can be found in Figure 2.5.

Algorithm 1 The LPD algorithm

Given: g_0, N
$g^1 \leftarrow \Xi_{\phi_0}^0(g^0)$
$f^1 \leftarrow \mathcal{T}^*(\Lambda_{\gamma_0}^0(g^0))/||\mathcal{T}||^2$
for $i = 1, ..., N-1$ **do**
  $g^{i+1} \leftarrow \Xi_{\phi_i}^i(\mathcal{T}(f^i), g^{i-1}, ..., g^0)$
  $f^{i+1} \leftarrow \Lambda_{\gamma_i}^i(\mathcal{T}^*(g^{i+1}), f^i, ..., f^0))/||\mathcal{T}||^2$
end for
return f^{i+1}

By utilizing the geometry of the problem in the unrolling, the LPD algorithm can reconstruct the image iteratively. Apart from a higher level of interpretability, unrolled models tend to have much fewer parameters and require less training data than more conventional deep learning architectures [1].

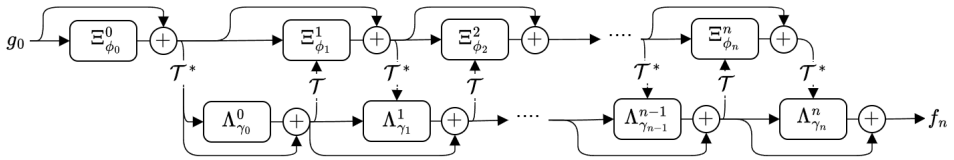


Figure 2.5: The LPD algorithm. Multiple arrows going into the same module implies concatenation. Note that memory has been omitted in this figure to increase interpretability.

2.2.2 Vision Transformers

The Transformer architecture, originally proposed for machine translation [4], utilizes multi-head attention to infer global (and local) dependencies.

When an example is processed, it is first converted into tokens. In the context of Vision Transformers, tokens are usually pixels or patches of an image. In an attempt to encode the meaning each a token, they are embedded in a high dimensional vector. The embedding of the tokens is learned during training. A positional encoding is added to each token to account for the relative positions within the example, i.e the locations of a patch/pixel within an image. The example is then processed by an attention mechanism, which has the purpose of altering the embedded token based on the other tokens in the example (its context). To calculate the attention, the tokens are converted into keys, queries and values. Linear layers are conventionally used for this, but convolutional layers could also be used. The attention function is defined, given queries Q, keys K and values V, as

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

where d_k is the dimension of key-query space. Multi-head attention is then defined as

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, ..., \text{head}_h\right)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{aligned}$$

An intuitive way to understand the use of the query matrix is that the product of the query matrix and an embedded token updates the embedding with respect to how other embedded tokens should influence it. The product of the key matrix and an embedded token then represents an embedded tokens

influence on other embedded tokens. The matrix multiplication between the query and key is a similarity measure which, for each token, indicates how relevant a token is for the rest of the embedded tokens. Applying the softmax function normalizes the product to a probability distribution and the factor $\frac{1}{\sqrt{d_k}}$ is included for numerical stability. The change in token embeddings is then computed by multiplying the softmax of the product by the value matrix. This change is added to the original embedded tokens after which they are normalized.

After the tokens have been updated they are usually processed by a neural network. In [4] a Multilayer perceptron (MLP) is used, but convolutional layers could be used instead for image processing applications, as in [9]. The change in tokens from this neural network is then added to the updated embedded tokens and normalized. A visualisation of a Vision Transformer architecture can be seen in Figure 2.6.

One aspect which limits the usability of the architecture is the complexity of the self attention mechanism. Calculating attention is $\mathcal{O}(n^2 d)$, where n is the context size and d is the dimension of the feature vectors. Transformers were first applied to images in [5], where the authors used patches of images as tokens. Since then, CNNs have also been used to downsample the image before the tokenization, as a way to reduce sequence length while preserving features (for example, see [14]).

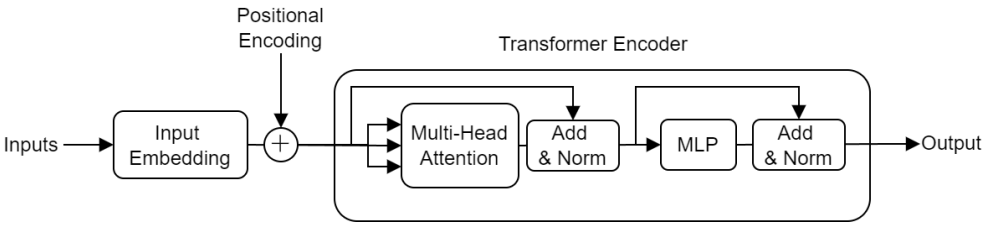


Figure 2.6: A basic Vision Transformer Architecture.

Chapter 3

Methods

In the following sections the data generation and collection, network architectures, training procedure, evaluation framework and system configuration is described.

3.1 Data Generation and Collection

The models are trained using only synthetic data, and then tested on both synthetic and experimental data acquired from the preclinical miniPET-3 system [15]. Both the synthetic and experimental data considered in this project is three dimensional. However, due to the extensive memory requirements associated with processing an extra dimension, the architectures presented operate only on 2D images, and the third data dimension is handled in the same way as if it were a batch of different examples. The forward and back projections operate on the 3D data which decreases training time substantially.

3.1.1 Scanner Definition

The preclinical system [15] has 12 detector modules, each with 35×35 crystals. To generate synthetic data, the acquisition geometry is modeled using a regular polygon PET projector with 35 rings, 12 sides, each with 35 crystals. The scanner used to generate the synthetic data and perform the projections is defined to match the preclinical system [15] in order to facilitate transfer learning from the synthetic to the experimental data.

3.1.2 Synthetic Data Generation

The synthetic data consists of randomly generated three dimensional ellipsoids of shape $147 \times 147 \times N$ pixels. The number of ellipses in an image is drawn from a Poisson distribution with rate parameter $\lambda = 10$, with center points drawn uniformly from the image dimensions, axis lengths drawn from an exponential distribution with rate parameter $\lambda = 0.3$ and angling drawn uniformly from the interval $[0, 2\pi)$. The activity concentrations for the ellipsoids are drawn uniformly from the interval $[0, 1]$. Since the architectures assume independence of the two dimensional slices of a 3D image, N can be chosen to optimize memory allocation.

In order to generate hollow structures, the generation described above is repeated but this time with rate parameter $\lambda = 0.15$ for the axes of the ellipsoids, and then subtracted from the original ellipsoids. To ensure that the image consists of only positive pixel values, negative pixel values are set to 0.

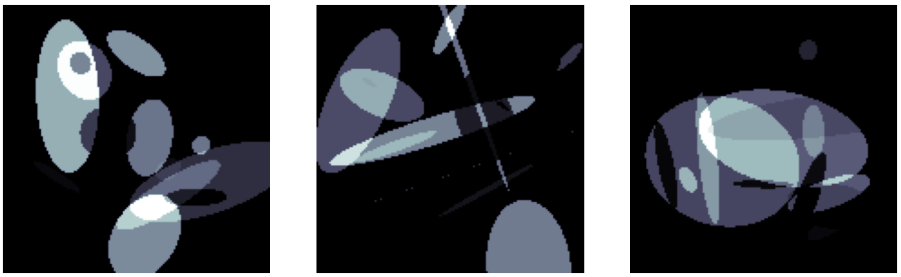


Figure 3.1: Slices from three different examples of the ground truth.

The ground truth data is generated as described above and a few examples can be seen in Figure 3.1. In order to account for lower spatial resolution of the preclinical system [15], Gaussian blur, with $\sigma = 2$ and kernel size 5, is applied to the images.

Since fludeoxyglucose, which is used in the real data generation for this project, has an average positron range of 0.6 and a maximum positron range of 2.4, and 1mm corresponds to approximately 1.8 pixels for the acquisition geometry, a kernel size of 5 is reasonable for the Gaussian blurring. Similarly, σ is determined to match the spatial resolution of the target system.

The images are forward projected, after which Poisson noise is added to the resulting sinograms. To produce the noisy pixel values, samples are drawn from a Poisson distribution, with rate parameter equal to the original pixel value divided by the noise level, which are then multiplied by the noise level. Producing noisy images in this way creates variations in the data while

preserving the overall distribution. Importantly, areas with high activity will have more noise, which is also expected for experimental data.

The noise level during training is drawn uniformly from the interval $[0.1, 1.2]$ for each example. Since the noise level of the experimental data is unknown, we choose a wide span of noise levels to train on. The training data is generated randomly on the fly for each batch. An example of the training data can be seen in Figure 3.2.

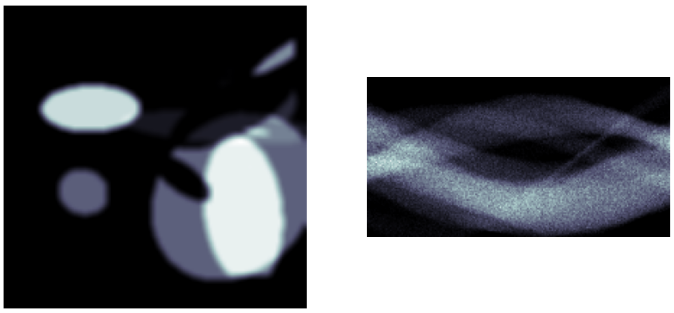


Figure 3.2: An example of a blurry image and its noisy sinogram.

The synthetic test set is generated from the SLP. As before, the Gaussian blur is applied to the images prior to sinogram generation. Different levels of noise can then be added to the sinogram for evaluation purposes. An example from the synthetic test set can be seen in Figure 3.3.

Figure 3.3: A slice of the SLP along with its blurry test example as well as noisy sinogram.

3.1.3 Preclinical Data Collection

To evaluate the performance of the models in a more realistic setting, a mouse-like phantom has been 3D printed, see Figure 3.4. The activity concentrations for the different organs in the mouse, for two variations, can be seen in Table

3.1. This data consists of sinograms collected from the preclinical system [15]. The data was collected every minute for an hour and so the test set contains a variety of measurements. The phantom design and data acquisition from the preclinical system [15] was performed by Guazzo and Colarieti-Tosti for [3] and is used in this thesis for both evaluation on experimental data and as a way to experiment with transfer learning.

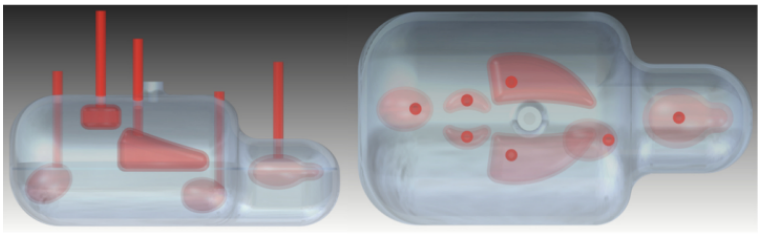


Figure 3.4: Side and top view of the mouse-like phantom.

Table 3.1: Activity concentrations for test phantom with two different measurements.

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

In Figure 3.5 a sinogram of the mouse-like phantom captured by the preclinical system [15] is included. Apart from the noisy sinogram data, note the black grid overlay. This grid is likely due to gaps between the detectors of the scanner.

3.2 Network Architectures

3.2.1 U-Net LPD

First, we consider the U-Net LPD to establish a baseline. A three iteration model is fit and evaluated. This is the approach considered by Guazzo and Colarieti-Tosti in [3]. A visualization of the U-Net architecture can be seen in Figure 3.6. The convolutional blocks consist of a convolutional layer, followed by batch normalization over the channels, followed by a ReLU activation function, repeated twice. In each block followed by a downsampling operation,

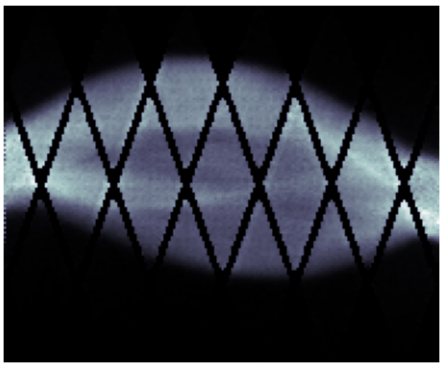


Figure 3.5: Sinogram of the mouse-like phantom.

the channels are expanded by a factor of two in the first convolutional layer. Similarly, in each block followed by an upsampling operation, the channels are reduced by a factor of two in the first convolutional layer. The kernel size in all convolutional blocks within the U-Net is 3. The downsampling is applied using maxpooling and the upsampling is applied using a transposed convolution operator. The down- and upsampling is applied so that the image dimension is either reduced or expanded by a factor of 2. The U-Net LPD is trained on images of size $147 \times 147 \times 21$.

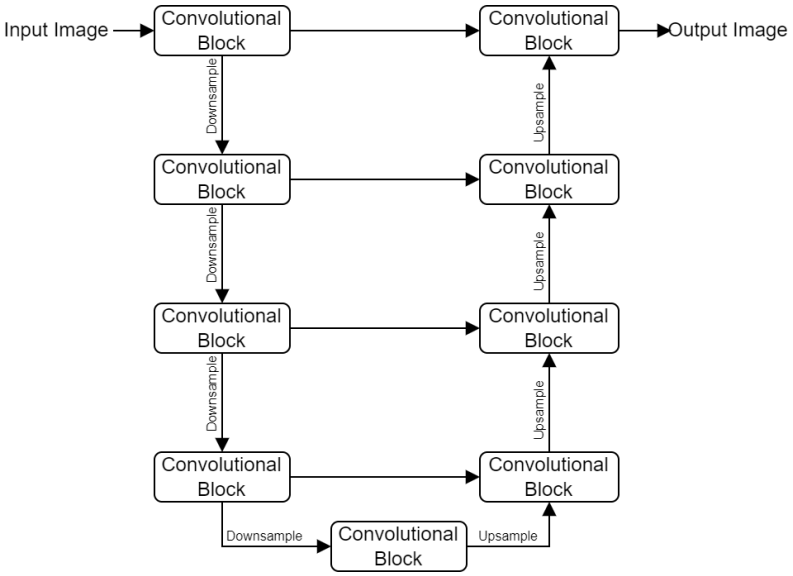


Figure 3.6: The U-Net architecture. As before, multiple arrows going into the same module implies concatenation.

3.2.2 Dual-Domain Transformer LPD

To produce a feasible embedding for a Transformer, the sinogram either has to be down-sampled in some way or separated into sub-problems for which attention can be used. The common approaches for down-sampling, mentioned in Chapter 2 assume feature locality within the sinogram domain, and so may not be suitable. Instead, we utilize the structure of the reconstruction problem by forward projecting a patch from the image domain, to see which pixels of the sinogram correspond to the patch. The pixel values of are then extracted according to which indices are active. This approach has the advantage that the resulting sinusoidal strips from the pixel extraction at the corresponding indices provide a representation where the assumption of feature locality within the sinogram holds, enabling down-sampling of the sinogram and an appropriate separation of the original problem. A visualization of the patching process can be seen in Figure 3.7.

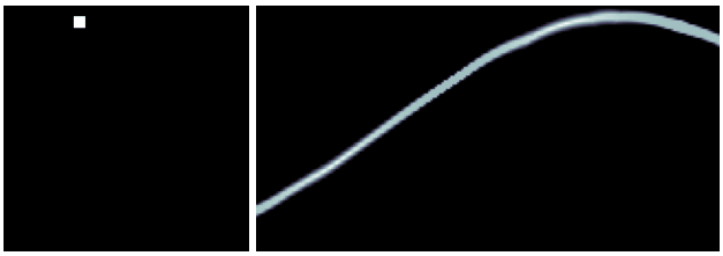


Figure 3.7: Visualization of the sinogram patching. One patch in image space corresponds to a curve in the sinogram domain.

For the Dual-Domain Transformer LPD, we apply attention using a Transformer on embedded tokens originating from sinusoidal curves from the patching process, and then sum the weighted output to produce a sinogram from the curves. A patch size of 7×7 pixels is used. The embedding is produced by a linear layer mapping the values of each curve (including the channels) to a vector of dimension $d = 196$. A learnable positional encoding is added after the embedding. The embedded tokens are then updated by a Transformer block (see Figure 2.6). To reconstruct an image from the updated embedded tokens, they are mapped using a linear layer back to the original dimension of the sinusoidal curves. The output is weighted according to the level of activation within each curve and added to the original sinogram.

Mathematically, the indices resulting from the forward projection of a

patch $p_{i,j}$ is

$$I_{i,j} = \left\{(k,l) : (\mathcal{T} p_{i,j})_{k,l} > 0\right\}.$$

The patches $p_{i,j}$ considered are images including non overlapping (from image to image) squares of 7×7 pixels where the activity is set to one and the remaining values of the image are set to zero. The extracted values from a sinogram g are

$$z_{i,j} = \{g_{k,l} : (k,l) \in I_{i,j}\},$$

for indices k, l of the sinogram. The sinusoidal curves $z_{i,j}$ are then embedded using a linear layer, and updated by a Transformer

$$\begin{aligned} z_{i,j}^{emb} &= \text{Linear}(z_{i,j}) + P \\ Q,\ K,\ V &= \text{Linear}(z_{i,j}^{emb}),\ \text{Linear}(z_{i,j}^{emb}),\ \text{Linear}(z_{i,j}^{emb}) \\ z_{i,j}^{emb} &= \text{LayerNorm}(z_{i,j}^{emb} + \text{MultiHead}(Q,K,V)) \\ z_{i,j}^{emb} &= \text{LayerNorm}(z_{i,j}^{emb} + \text{MLP}(z_{i,j}^{emb})), \end{aligned}$$

where P is a learnable positional encoding. The output is constructed as

$$\begin{aligned} z_{i,j} &= \text{Linear}(z_{i,j}^{emb}) \\ g_{k,l} &= g_{k,l} + \sum_{i,j} (z_{i,j})_{k,l} (W_{i,j})_{k,l}, \end{aligned}$$

where the activation weights $(W_{i,j})_{k,l}$ are calculated as

$$(W_{i,j})_{k,l} = (\mathcal{T} p_{i,j})_{k,l}.$$

The Dual-Domain Transformer is implemented using a Transformer in each dual block, with 2 heads and an embedding dimension of 196 per head, with 3 iterations, where the primal channel consists of U-Nets. The Dual-Domain Transformer LPD is trained on images of size $147 \times 147 \times 21$.

3.2.3 Restormer LPD

Even though the U-net LPD may seem to violate the assumption of global features within the sinogram, it works surprisingly well. This is likely due to the structure of the U-net architecture, which could provide a way for the model to infer global dependencies, even though the operations involved are

inherently local. In light of this, we consider the Restormer architecture as an alternative to the U-net and Dual-Domain Transformer architectures. By applying channel-wise attention, the Restormer has been highly successful for image restoration [9]. A visualization of the Restormer architecture can be seen in Figure 3.8.

In each Transformer block, a series of Transformers (similar to Figure 2.6) are applied, but here the embedding is calculated using a convolutional layer with a kernel size of 1, which expands the channels by a factor of 3. The Transformers in the Restormer use a depth-wise convolutional layer with a kernelsize of 3 instead of linear layers to compute the key, query and value matrices. Depth-wise convolutions are often used to capture features channel wise instead of spatially. Multi-head attention is applied on the channel dimension, contrary to the more conventional approach of flattening the spatial dimensions and applying attention.

Furthermore, instead of an MLP, the Transformers in each Transformer block use a Gated-Dconv Feed-Forward Network. The up- and down-sampling is performed using convolutional layers, where the channels are expanded/reduced by a factor of 2. For more information, see [9].

The implementation of the Restormer LPD uses 3 iterations, where both the primal and dual channels of the LPD use Restormers. The number of Transformers in each block (except for the refinement) is 2, 3, 3, 4, 3, 3, and 2, respectively, with 1, 2, 4, 4, 4, 2, 1 number of heads. The refinement uses 4 Transformers with 1 head. The Restormer LPD is trained on images of size $147 \times 147 \times 2$.

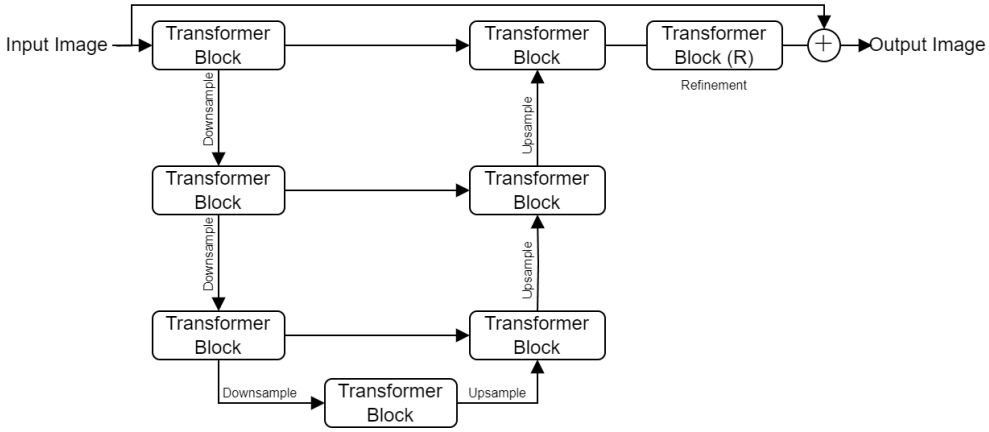


Figure 3.8: The Restormer architecture.

3.2.4 U-Net LPD with Cross Attention Blocks

An image and its corresponding sinogram essentially consists of the same information, although in different forms and domains. With this in mind, we consider an architecture designed to utilize the similarity of the two domains. By implementing Cross-Attention Blocks (CABs) in between the primal and dual channels of the LPD algorithm, we attempt to utilize the duality of the problem.

A CAB takes both a sinogram and an image as inputs. The patching process for the Dual-Domain Transformer is repeated for the sinogram data, and the image data is patched using the same patch size and embedded to the same dimension. The queries, keys, and values are calculated separately for the domain by linear layers. Given the primal queries, keys, and values Q_p, K_p, V_p and the dual queries, keys, and values Q_d, K_d, V_d, we calculate Cross-Attention as

$$\begin{aligned} \text{att}_p &= \text{MultiHead}(Q_d, K_p, V_p)), \\ \text{att}_d &= \text{MultiHead}(Q_p, K_d, V_d)). \end{aligned}$$

The attention in each domain is added to the corresponding embedded tokens, which are then normalized and processed by a MLP. The output of the MLPs are normalized and added to the input sinogram and image. A visualization of the LPD algorithm with CABs implemented in between the primal and dual channels can be seen in Figure 3.9.

The fact that no information is allowed to be exchanged between the domains of the LPD could be seen as regularizing, and there may be advantages of allowing exchange of information without applying projections.

The U-Net LPD with CABs is implemented with three iterations, where the CABs consists of two Transformers, one for each domain, where the queries have been swapped. The Transformers use 2 heads, each with a dimension of 96. Two CABs are added in between iterations 1-2, and 2-3. A patch size of 7×7 is used. The CAB LPD is trained on images of size $147 \times 147 \times 21$.

The models along with their number of parameters can be seen in Table 3.2.

3.3 Training Procedure

The models are trained using Mean Squared Error (MSE) loss for 150 epochs with 10500 2D examples per epoch. Normal initialization with $\sigma = 0.02$ is

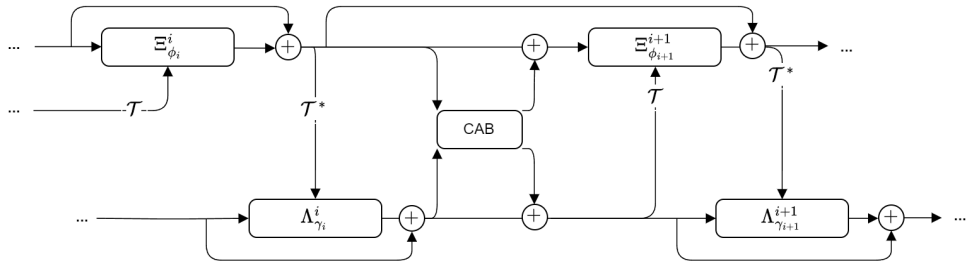


Figure 3.9: The LPD algorithm with CABs.

Table 3.2: Models and their number of parameters.

Model	**#Params**
U-Net	12,862,278
Dual-Domain Transformer	13,560,012
Restormer	23,248,584
U-Net CAB	15,240,686

used for the weights. AdamW is used as the optimizer with weight decay of 0.01 and initial learning rate of 10^{-4}. 1cycle learning rate policy is used for scheduling the learning rate, with a maximum learning rate of $5 \cdot 10^{-4}$. Mixed precision training is used to reduce the memory requirements and decrease training time.

3.4 Evaluation Framework

There are four main ways in which the performance of each model is evaluated. Firstly, the loss function itself MSE is a measure of similarity between the ground truth and the reconstructed image. Furthermore, two common metrics for image restoration are included for evaluation: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). The PSNR between the ground truth I and the reconstruction K is calculated as

$$PSNR(I,K) = 10 \cdot \log_{10}\left(\frac{MAX(I)^2}{MSE(I,K)}\right).$$

The SSIM is calculated as

$$SSIM(I,K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)},$$

where μ_I and μ_K are the pixel averages of the images, σ_I and σ_K are the pixel variances of the images, σ_{IK} is the pixel covariance between the images and c_1 and c_2 are constants included for computational reasons.

Additionally, visual inspection of the reconstructions is used as a qualitative method. Visual inspection can reveal artifacts and shape deformations that the other metrics may be oblivious to, but are important for reconstruction quality.

For comparison against the clinical reconstruction standard, the Maximum-Likelihood Expectation-Maximization (MLEM) algorithm was implemented and run for 20 iterations on the test sets.

3.5 System Configuration

3.5.1 Software

All code for this project is written in Python. Pytorch [16] is used for defining and training the models, ODL [17] is used for generating the ellipsoids and parallelproj [18] is used for defining the projector geometry and calculating the forward and back projections. By utilizing parallelproj [18], forward and back projections can be calculated using the GPU, decreasing the time required for training and inference.

3.5.2 Hardware

Training was conducted in parallel on six NVIDIA Tesla K80 GPUs.

Chapter 4

Results

In this chapter, the performance of the models discussed in the previous chapter is evaluated on the synthetic test set and on the preclinical data. The models are also analyzed using visualizations of the unrolling and an attempt is made to evaluate model robustness.

4.1 Synthetic Data

In Figure 4.1 the results are plotted for different levels of noise. We see that the Restormer LPD seems to give the best results for all figures of merit, followed by the U-Net LPD. The Dual-Domain Transformer LPD and the U-Net CAB LPD perform the worst on all metrics. The results for three different noise levels are reported in Tables 4.1, 4.2 and 4.3. We observe a similar pattern as in Figure 4.1, where the Restormer LPD gives the best results for all figures of merit.

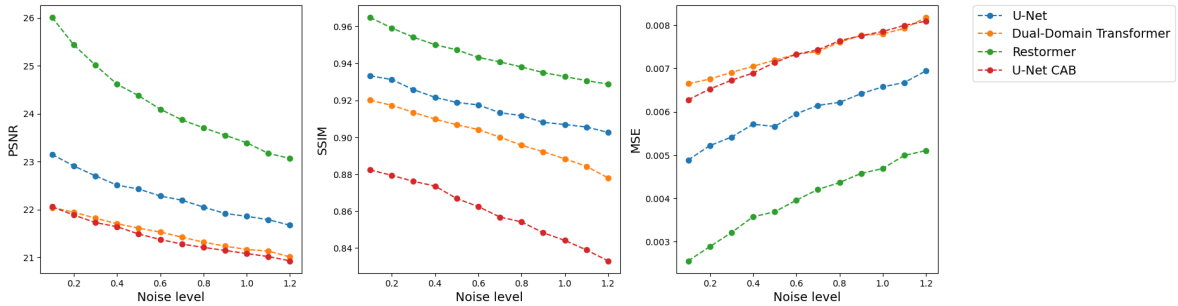


Figure 4.1: Plots of PSNR, SSIM and MSE for different noise levels evaluated on the Shepp-Logan phantom.

Table 4.1: SSIM of the synthetic test set for different noise levels.

Model	**SSIM (n=0.3)**	**SSIM (n=0.7)**	**SSIM (n=1.0)**
U-Net	0.927	0.916	0.909
Dual-Domain Transformer	0.913	0.900	0.887
Restormer	0.955	0.940	0.933
U-Net CAB	0.877	0.858	0.844
MLEM	0.942	0.926	0.917

Table 4.2: PSNR of the synthetic test set for different noise levels.

Model	**PSNR (n=0.3)**	**PSNR (n=0.7)**	**PSNR (n=1.0)**
U-Net	22.729	22.275	21.944
Dual-Domain Transformer	21.826	21.397	21.140
Restormer	25.064	23.835	23.339
U-Net CAB	21.729	21.305	21.077
MLEM	24.288	23.031	22.576

Table 4.3: MSE of the synthetic test set for different noise levels.

Model	**MSE (n=0.3)**	**MSE (n=0.7)**	**MSE (n=1.0)**
U-Net	5.42e-3	5.98e-3	6.45e-3
Dual-Domain Transformer	6.94e-3	7.49e-3	7.84e-3
Restormer	3.18e-3	4.22e-3	4.74e-3
U-Net CAB	6.74e-3	7.42e-3	7.86e-3
MLEM	6.45e-3	6.45e-3	6.45e-3

(a) Ground truth. (b) Blurred ground truth. (c) MLEM reconstruction with a noise level of 0.7.

Figure 4.2: Images of SLP included for comparison with the reconstructions.

(a) U-Net LPD (b) Dual-Domain Transformer LPD (c) Restormer LPD (d) U-Net CAB LPD

Figure 4.3: Reconstructed images from the test set with noise level 0.3.

For visual comparison of the results on the synthetic test set, Figure 4.2 includes the ground truth of the SLP with and without Gaussian blur, as well as a MLEM reconstruction.

Comparing the reconstructed images in Figure 4.3 with the ground truth, it is not as obvious which reconstruction is better. For example, the U-Net LPD performs better than the Dual-Domain Transformer LPD with respect to all figures of merit but the reconstructions are visually similar. As for the Restormer LPD, it is clearly the best model for reconstructing the shell of the phantom, where it shows impressive similarity to the (non blurred) ground truth. Furthermore, none of the models seem to be able to effectively reconstruct the smaller white areas in between the cavities. This could be expected, due to their challenging placement and size. Gaussian blurring is also likely to exacerbate these difficulties.

In Figures 4.4 and 4.5 the synthetic test data is reconstructed from more noisy sinograms. Here we observe degradation for all models. We primarily observe that the cavities are increasingly misshaped, while the shape of the shell and the shape and contrast of the white area in between the cavities remain relatively well reconstructed. From visual inspection, the quality of

(a) U-Net LPD (b) Dual-Domain Transformer LPD (c) Restormer LPD (d) U-Net CAB LPD

Figure 4.4: Reconstructed images from the test set with noise level 0.7.

(a) U-Net LPD (b) Dual-Domain Transformer LPD (c) Restormer LPD (d) U-Net CAB LPD

Figure 4.5: Reconstructed images from the test set with noise level 1.

the Restormer LPD and the U-Net LPD seem higher than the reconstructions provided by the other two models.

4.2 Preclinical Data

The MLEM reconstruction of the experimental data can be seen in Figure 4.6, and the reconstructions provided by the models can be seen in Figure 4.7. The reconstructions differ notably more than the reconstructions of the synthetic test data. The U-Net LPD provides a relatively misshaped reconstruction with beam-like artifacts originating from the activity sources. The Dual-Domain Transformer LPD is able to reconstruct a more homogeneous phantom, albeit still with some artifacts. The Restormer LPD produces a brighter reconstruction with substantial beam-like artifacts and the reconstruction provided by the U-Net CAB LPD contains the most artifacts. From visual inspection, we prefer reconstructions that are homogeneous within each organ and the body, since we know that the activity distribution there is the same. Based on this, the Dual-Domain LPD produces the best reconstruction.

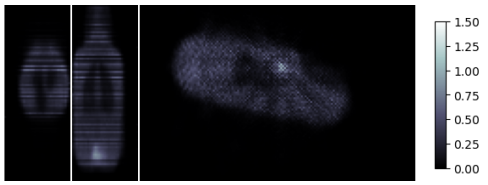
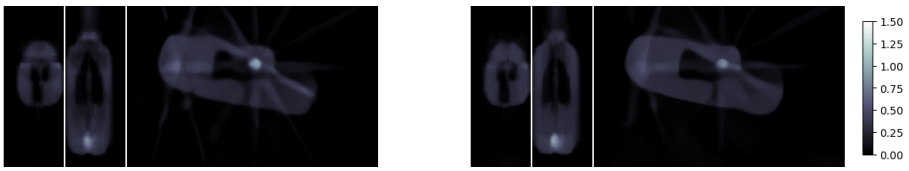
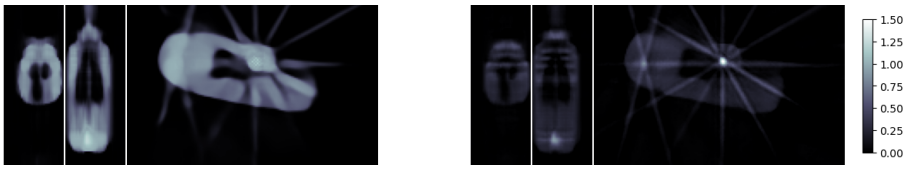


Figure 4.6: MLEM reconstruction of the mouse-like phantom.



(a) U-Net LPD

(b) Dual-Domain Transformer LPD



(c) Restormer LPD

(d) U-Net CAB LPD

Figure 4.7: Reconstructions of the mouse-like phantom.

4.3 Model Analysis

4.3.1 Visualizing the Unrolling

Each iteration in the LPD produces an image, and to gain knowledge of how each model operates, the output of all iterations of the primal channels are plotted in Figures 4.8, 4.9, 4.10 and 4.11. Interestingly, the Restormer LPD seems to learn some form image inversion in the first iteration for reducing the noise. There is also a notable difference in how the U-Net CAB LPD operates, where the second iteration appears noisy. The seemingly progressive denoising present in Figures 4.8 and 4.9 differs notably from the unrolling visualized in [3].

4.3.2 Model Robustness

In an attempt to analyze model robustness, the pixel variance of the different reconstructions, when tested on the SLP with a noise level of 0.3, was calculated and plotted. This could give an indication of which areas of the

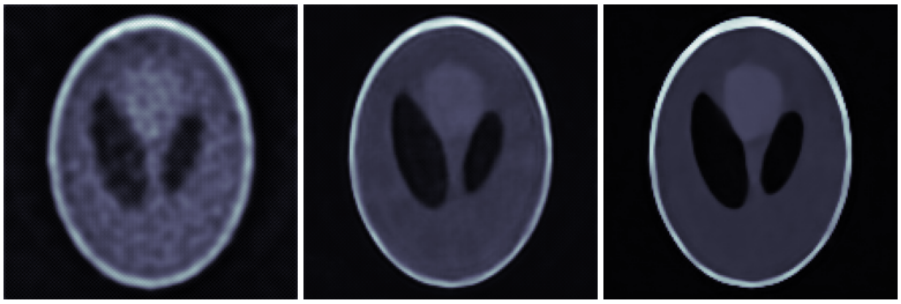


Figure 4.8: Outputs of the iterations of the U-Net LPD.

Figure 4.9: Outputs of the iterations of the Dual-Domain Transformer LPD.

reconstructions are the most sensitive to the noise. The results are plotted in Figure 4.12. For the Restormer LPD, we see lower levels of variance for certain parts of the shell. Furthermore, it seems that the Dual-Domain Transformer LPD and the U-Net CAB LPD show higher variance around the cavities than the other two models.

It is worth noting that since the noise is higher in areas with high activity, one could expect this to only yield high variance in areas where activity is high. This is partially true, but does not explain the increase in variance around the cavities and the low level of variance in the middle part of the shells.

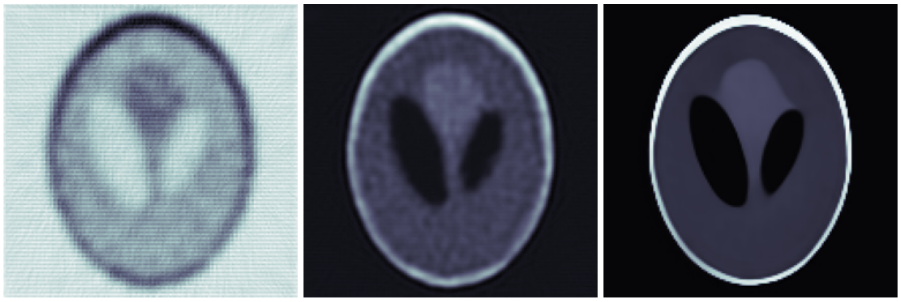


Figure 4.10: Outputs of the iterations of the Restormer LPD.

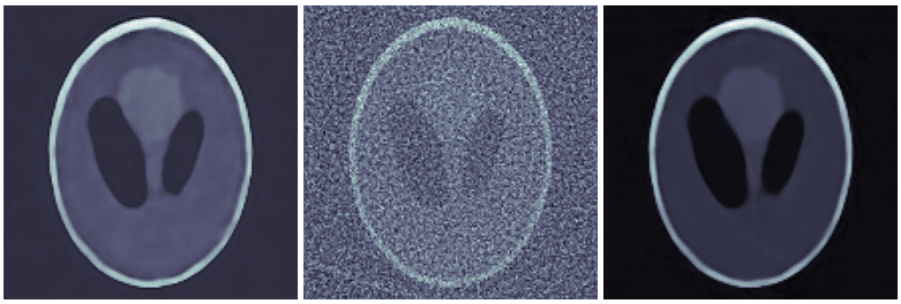
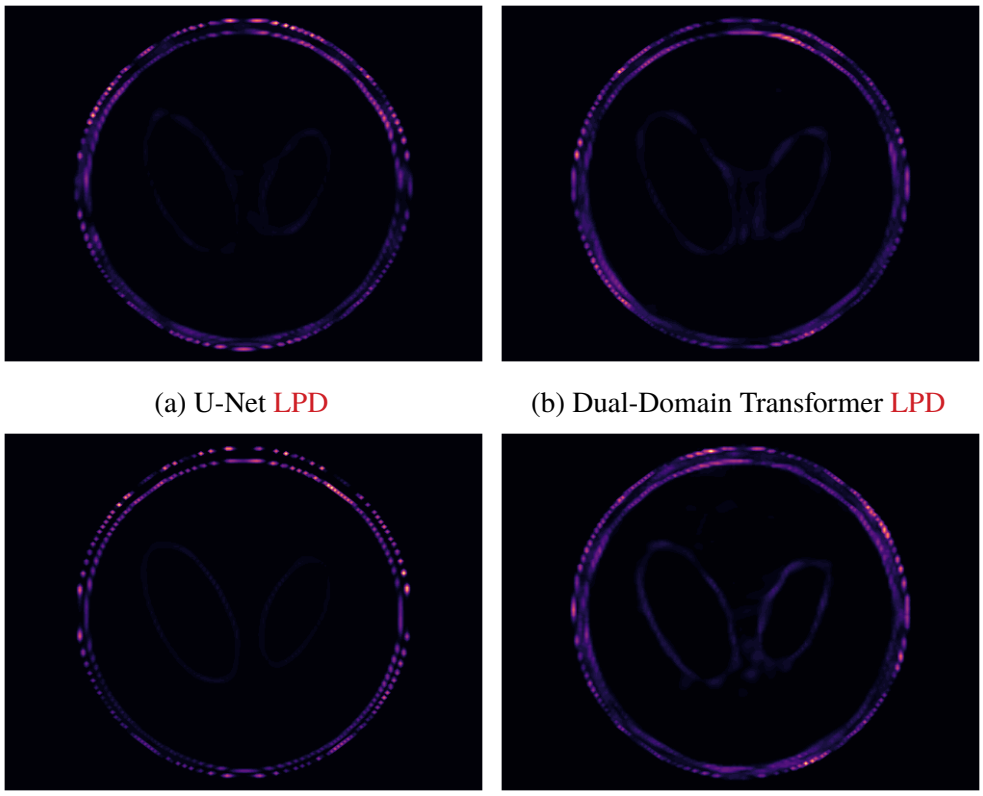


Figure 4.11: Outputs of the iterations of the U-Net CAB LPD.



(a) U-Net LPD (b) Dual-Domain Transformer LPD

(c) Restormer LPD (d) U-Net CAB LPD

Figure 4.12: Variances of the model outputs for noise level 0.3.

Chapter 5

Discussion

The results show promise in Transformer-based LPD algorithms for PET. The Restormer LPD performs the best on the synthetic test set quantitatively and arguably also visually. Furthermore, even though the Dual-Domain Transformer LPD shows worse performance than the baseline U-Net LPD, visual inspection of the reconstructions of the preclinical data are not as misshaped and include fewer artifacts. The material within each organ and the body of the mouse-like phantom is more homogeneous than for the other reconstructions and the beam-like artifacts are less visible. The U-Net CAB LPD likely requires a more thoughtful implementation, since it performs significantly worse than the U-Net LPD.

The memory requirements and throughput of the models dictate the practicality in compute constrained systems. The U-Net LPD, the Dual-Domain Transformer LPD and the U-Net CAB LPD seem comparable with respect to throughput and memory requirements, while the Restormer LPD took significantly longer to train, see Table 5.1. This is mostly due to the high memory requirements which necessitate a low batch size.

Table 5.1: Total training time and memory usage for the models.

Model	**Training time (hours)**	**Memory usage (per GPU)**
U-Net	32.42	12.69GB
Dual-Domain Transformer	38.06	9.04GB
Restormer	298.23	18.95GB
U-Net CAB	32.29	14.01GB

Revisiting the original assumption of imposed feature locality of the CNNs in the U-Net LPD, it is worth considering that this may be incorrect. The U-

Net LPD shows impressive performance and the down- and up-sampling of the architecture could allow the model to infer global dependencies even though the CNNs themselves are local in nature. Even so, the Transformer-based approaches seem to surpass the U-Net LPD in different aspects on the synthetic test data and the preclinical data. It is worth noting, however, apart from the difference in memory requirements, that there is a substantial difference in the number of parameters between the Restormer LPD and the other models (see Table 3.2).

5.1 Generalization from Synthetic data

The method outlined in this work relating to data generation seems promising. The models are fit solely on synthetic data, where the acquisition geometry has been constructed to match a real world system, and the images are blurred prior to data generation to match the resolution of the target system. The models are able to provide decent reconstructions from the preclinical data.

Another approach was taken in [3], where an attempt to find the ground truth of the preclinical data was made after which the the model was trained on a mixture of synthetic and preclinical data. Interestingly, apart from the presence of the beam-like artifacts in the reconstructions provided in Chapter 4, the results of the method used in this work are comparable if not superior.

5.2 The Importance of Data Quality for Reconstruction

The baseline U-Net LPD is the architecture proposed in [3]. Even though the model is fit using similar amounts of data points, the U-Net LPD trained using the data generation described in Chapter 3 outperforms the identical model in [3], while being trained using a wider range of noise levels. The reason for this could lie in the data generation, which differs in a few ways. One factor could for example be the inclusion of cavities in this work, which may capture the structure of the synthetic test set more accurately. It is important to note that there are also other differences in the training process which could contribute to the observed difference in reconstruction quality, such as the usage of a different loss function and learning rate scheduler in [3].

5.3 The Quality of Figures of Merit

SSIM, PSNR and MSE are commonly used figures of merit for image and reconstruction quality. Yet, in light of the results, they do not seem to necessarily correspond to the perceived image quality of the reconstruction. This can be noted particularly when comparing the reconstructions from the synthetic test set, where reconstructions which may visually look similar have vastly different values when judged quantitatively on the figures of merit.

Chapter 6

Conclusions

A few conclusions can be drawn from the results, and they have been partially discussed in previous chapters. Firstly, it is not trivial how a feasible Transformer-based architecture should be implemented to solve the reconstruction problem. A relevant question is how to provide an embedding which is representative while being attainable with respect to memory and throughput.

Secondly, it can be concluded that Transformer-based models can, with respect to most of the metrics outlined in Chapter 3, outperform previously proposed CNN-based methods. Furthermore, it is clear that the figures of merit used do not accurately represent perceived image reconstruction quality.

For the quality of the reconstructions, data generation is likely to be a very important component. Comparing the results of the U-Net LPD with the results achieved in [3] indicates advantages of improved data generation.

6.1 Limitations

The limited use of data collected from real world systems is clearly a limitation of this work. Realistic data may also be of higher image resolution, which would require increased levels of memory to handle and would also increase the training time. The problem is by nature of higher dimension than presented in this work, since the images are three dimensional and there is also a temporal dimension. While the simplifications made in Chapter 3 alleviate memory issues and simplify potential reconstruction architectures, modelling the complete problem has the potential to improve the results substantially.

6.2 Future work

In light of the improvements made to the data generation compared to [3], further exploring how synthetic data can be leveraged may prove fruitful. This is especially interesting since we observe decent generalization from purely synthetic data to the data from the preclinical system [15]. Training the models on clinical data is an important extension which could render the models more useful for practical applications.

Furthermore, to gain more knowledge about how the LPD algorithm and similar reconstruction algorithms operate, the explainability of such architectures could be explored further, which may help guide the search for future models.

Methods for uncertainty estimation could be explored for the LPD algorithm and other deep reconstruction methods. Not only would this provide more insight into how the models operate, but it could also guide further developments. This is not necessarily a difficult task, see for example [19].

There are also a few types of modifications to the LPD algorithm itself that could be explored. For example, exploring using either different architectures and/or architectures of different depths in each iteration in the unrolling could allow the layers to be more specialized at certain tasks. Different iterative schemes than the LPD algorithm, for example schemes inspired by diffusion models, could also be explored further.

Lastly, modeling the complete problem with three spatial dimensions and one temporal dimension would likely allow for improved results.

6.3 Reflections

The theoretical nature of this work, especially the use of synthetic data, places the findings far from the practical applications of use for the patient. Even so, if the findings of this work prove equally hold for clinical data, the improvements compared to previous variants of the LPD algorithm may eventually improve medical imaging in practice.

References

[1] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *arXiv preprint arXiv:1912.10557*, 2020. [Online]. Available: https://arxiv.org/abs/1912.10557 [Pages 1 and 10.]

[2] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1322–1332, 2018. [Pages 1, 2, and 10.]

[3] A. Guazzo and M. Colarieti-Tosti, "Learned primal dual reconstruction for pet," *Journal of Imaging*, vol. 7, no. 12, p. 248, 2021. [Pages 2, 10, 16, 29, 34, 37, and 38.]

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762 [Pages 2, 11, and 12.]

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929 [Pages 2 and 12.]

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021. [Online]. Available: https://arxiv.org/abs/2103.14030 [Page 2.]

[7] A. Hatamizadeh, G. Heinrich, H. Yin, A. Tao, J. M. Alvarez, J. Kautz, and P. Molchanov, "Fastervit: Fast vision transformers with hierarchical

attention," *arXiv preprint arXiv:2306.06189*, 2023. [Online]. Available: https://arxiv.org/abs/2306.06189 [Page 2.]

[8] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," *arXiv preprint arXiv:2106.03106*, 2021. [Online]. Available: https://arxiv.org/abs/2106.03106 [Page 2.]

[9] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," *arXiv preprint arXiv:2111.09881*, 2022. [Online]. Available: https://arxiv.org/abs/2111.09881 [Pages 2, 12, and 20.]

[10] L. Yang, Z. Li, R. Ge, J. Zhao, H. Si, and D. Zhang, "Low-dose ct denoising via sinogram inner-structure transformer," *arXiv preprint arXiv:2204.03163*, 2022. [Online]. Available: https://arxiv.org/abs/2204.03163 [Page 2.]

[11] S.-Y. Zhang, Z.-X. Wang, H.-B. Yang, Y.-L. Chen, Y. Li, Q. Pan, H.-K. Wang, and C.-X. Zhao, "Hformer: highly efficient vision transformer for low-dose ct denoising," *Nuclear Science and Techniques*, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s41365-023-01208-0 [Page 2.]

[12] 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, 1974. doi: 10.1109/TNS.1974.6499235. [Online]. Available: ^1^ [Page 7.]

[13] S. Malladi, K. Lyu, A. Panigrahi, and S. Arora, "On the sdes and scaling rules for adaptive gradient algorithms," *arXiv preprint arXiv:2205.10287*, 2022. [Online]. Available: ^1^ [Page 9.]

[14] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," 2021. [Page 12.]

[15] A. K. Krizsan, I. Lajtos, M. Dahlbom, F. Daver, M. Emri, S. A. Kis, G. Opposits, L. Pohubi, N. Potari, G. Hegyesi, G. Kalinka, J. Gal, J. Imrek, F. Nagy, I. Valastyan, B. Kiraly, J. Molnar, D. Sanfilippo, and L. Balkay, "A promising future: Comparable imaging capability of mri-compatible silicon photomultiplier and conventional photosensor preclinical pet systems," *Journal of Nuclear Medicine*, vol. 56, no. 12,

pp. 1948–1953, 2015. doi: 10.2967/jnumed.115.157677. [Online]. Available: ^1^ [Pages 13, 14, 16, and 38.]

[16] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. [Page 23.]

[17] J. Adler, H. Kohr, and O. Oktem, "Operator Discretization Library (ODL)," 2024, software available from GitHub. [Online]. Available: https://github.com/odlgroup/odl [Page 23.]

[18] G. Schramm and K. Thielemans, "PARALLELPROJ—an open-source framework for fast calculation of projections in tomography," *Front. Nucl. Med.*, vol. 3, 2024. doi: 10.3389/fnume.2023.1324562 Sec. PET and SPECT. [Online]. Available: https://doi.org/10.3389/fnume.2023.1324562 [Page 23.]

[19] M. Havasi, R. Jenatton, S. Fort, J. Z. Liu, J. Snoek, B. Lakshminarayanan, A. M. Dai, and D. Tran, "Training independent subnetworks for robust prediction," 2020. [Page 38.]

[20] S. Keeling, "Probabilistic models for radioactive decay: Addendum to mathematical modeling in the natural sciences," https://imsc.uni-graz.at/keeling/modII_ss19/radiodecay.pdf, accessed: 2024-04-09. [Page 43.]

Appendix A

Supporting materials

The source code relevant to this project can be found at https://github.com/antonadelow/Transformer-Based-Learned-Primal-Dual-Reconstruction-for-PET.

A.1 Modeling PET as a Bernoulli Process

The following model for radioactive decay is based on [20] and may provide a more accurate representation.

Let $p_0, ..., p_n$ be some subsets of the 3D space of which the subject is in and let

$$\alpha_i(t) = \begin{cases} 1, & \text{if particle } i \text{ decays in } [0, t] \\ 0, & \text{otherwise} \end{cases} \tag{A.1}$$

where

$$P(\alpha_i(t) = 0) = e^{-\lambda t}, \quad P(\alpha_i(t) = 1) = 1 - e^{-\lambda t}, \quad t \geq 0 \tag{A.2}$$

Since

$$P(\alpha_i(s+t)|\alpha_i(t) = 0) = \frac{P(\alpha_i(s+t) = 0)}{P(\alpha_i(s) = 0)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda t}} = P(\alpha_i(s) = 0),$$

the probability of a future decay of particle i is independent of how long it has existed prior. Now, let

$$N_{p_j}(t) = \sum_{i=1}^{n_{p_j}} \alpha_i(t) \tag{A.3}$$

denote the total number of particles having decayed in subset p_j at time t,

where n_{p_j} is the number of particles in region p_j. It can be shown that

$$P(N_{p_j}(t) = k) = \binom{n_{p_j}}{k} \left(1 - e^{-\lambda t}\right)^k \left(e^{-\lambda t}\right)^{n-k}. \tag{A.4}$$

Thus, the expected number of emissions in region p_j at time t is

$$E[N_{p_j}(t)] = n_{p_j}(1 - e^{-\lambda t}) \tag{A.5}$$

To model the decay in this way is feasible since it preserves the property of exponential decay and the independence of future decays, while accounting for a decreasing amount of particles in the subject. In contrast, modelling the decay using a homogeneous Poisson process does not account for a decreasing number of expected decays (particles) over time. Furthermore, considering a non-homogenous Poisson process would allow for modelling a decreased expected decay rate, but this approach is still flawed due to the theoretical assumption of an infinite number of particles. Since the scanner detects emissions along LORs, we let the subsets $p_0, ..., p_n$ consist of such lines.

While this way of modeling the problem may be more accurate, it requires more compute for data generation. Additionally, since we refrain from utilizing the temporal dimension, the advantages of using this method are likely negligible for this project.

TRITA –
Stockholm, Sverige 2024

www.kth.se