Learning Shapes For Efficient Segmentation of 3D Medical Images using Point cloud

Mohammad Sadil Khan

Master Thesis Defence
M.Sc in Machine Learning and Data Mining
University Jean Monnet

CREATIS

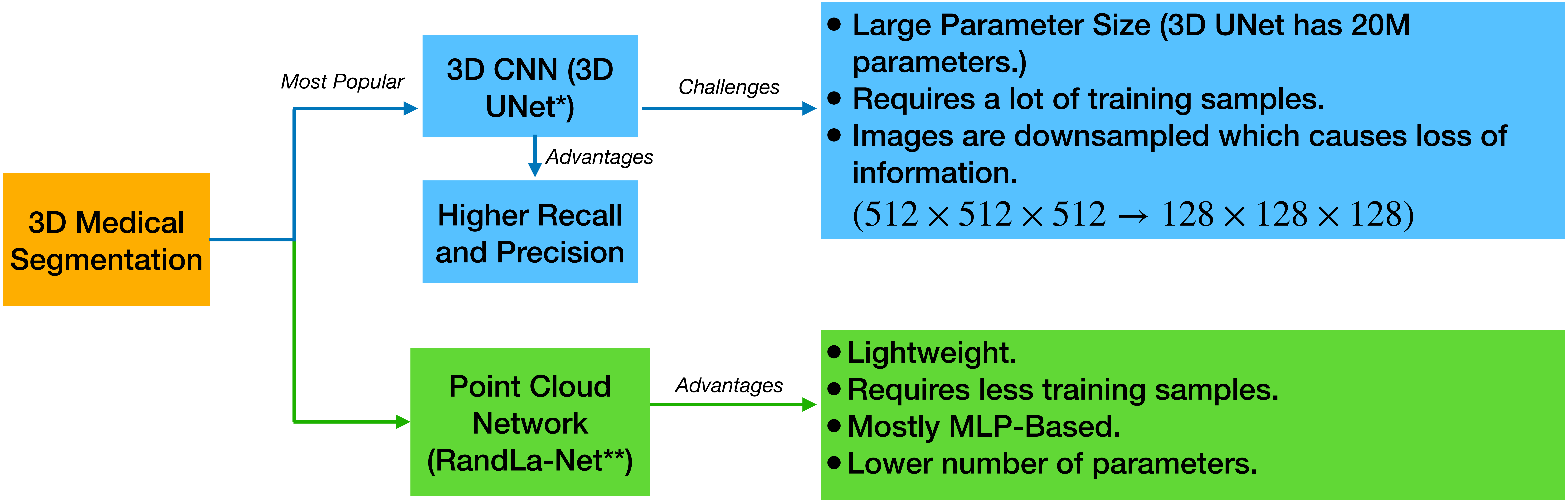
Supervisors : Razmig Kéchichian, Sébastien Valette and Julie Digne

30th June 2022

Contents

Motivation

- The goal of this internship to develop a novel approach to perform image segmentation of organs in 3D Medical Image using point clouds.



*[*3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*]*

*[**RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds*]*

Internship Task

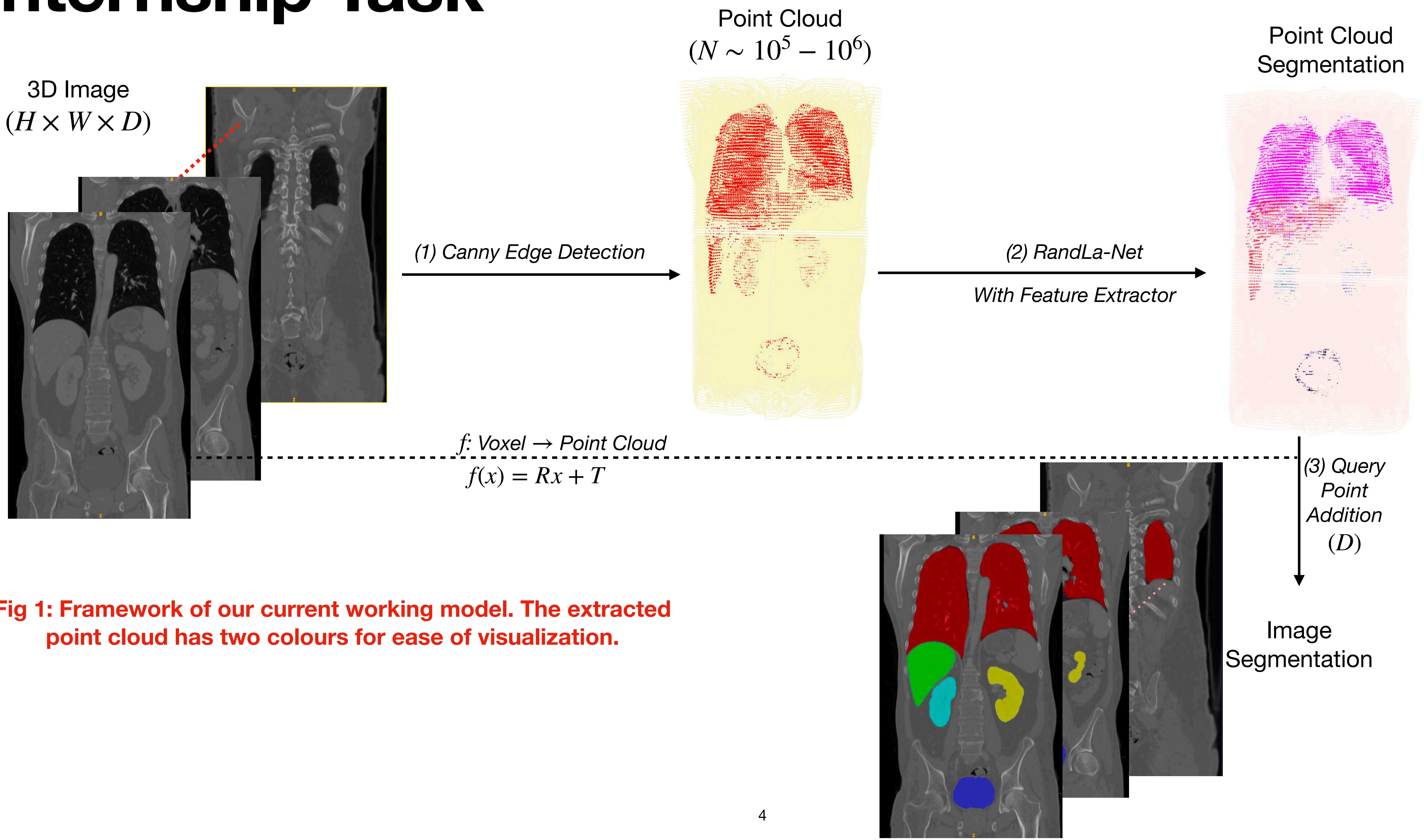


Fig 1: Framework of our current working model. The extracted point cloud has two colours for ease of visualization.

Internship Task - Point Cloud

- Lightweight and requires less memory than voxels.

Advantages over Voxels

- Unordered
- Irregular
- Lack of connectivity

Segmentation Challenges

Point Cloud

Segmentation Models

- Voxel Based
- Graph Based
- MLP Based

Mathematical Expression

$$(P = \{(p_i, f_i)\}_{i=1}^{N}, p_i \in \mathbb{R}^3, f_i \in \mathbb{R}^d)$$

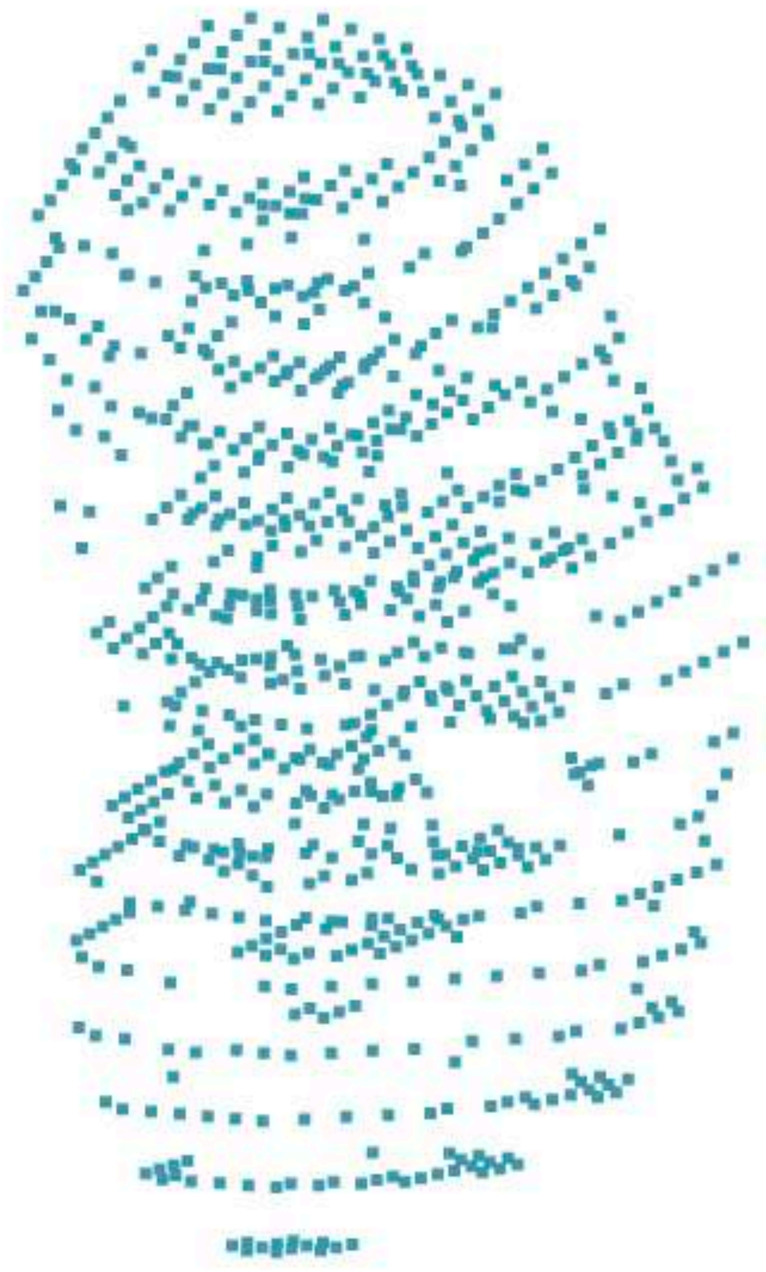


Fig 2: Point cloud of left kidney

We used RandLa-Net, an attention based large-scale point cloud segmentation network for our task.

Data Preparation - Dataset

Dataset

- 20 Contrast Enhanced CT Images from Visceral Dataset*.
- Dimension $(512 \times 512 \times 450 \rightarrow 128 \times 128 \times 112)$ for faster training.

Task

- Binary Segmentation (*Liver and Lungs*)
- Multi-Class Segmentation (*Liver, Lungs, Bladder, Left and Right Kidney*)

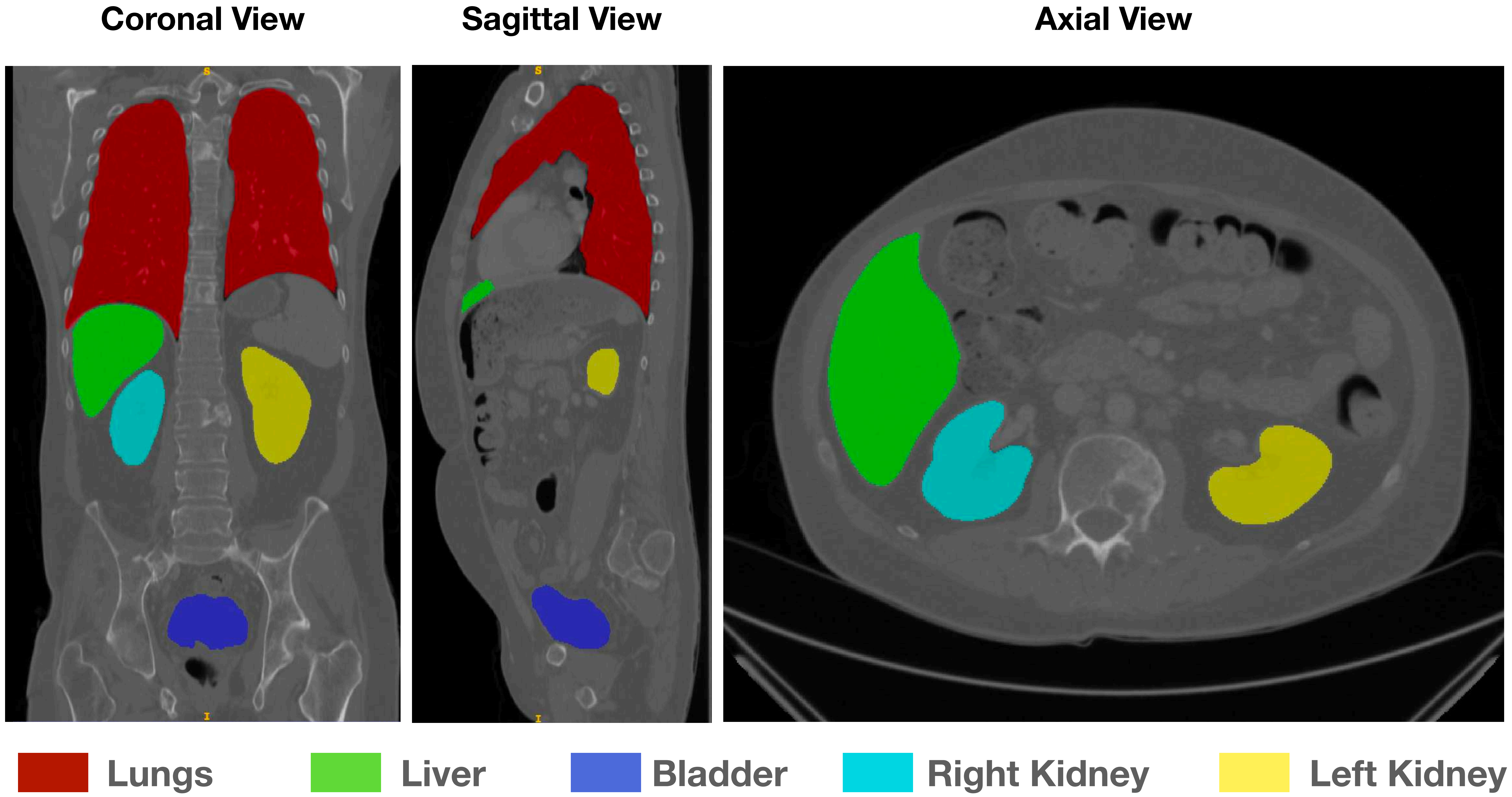
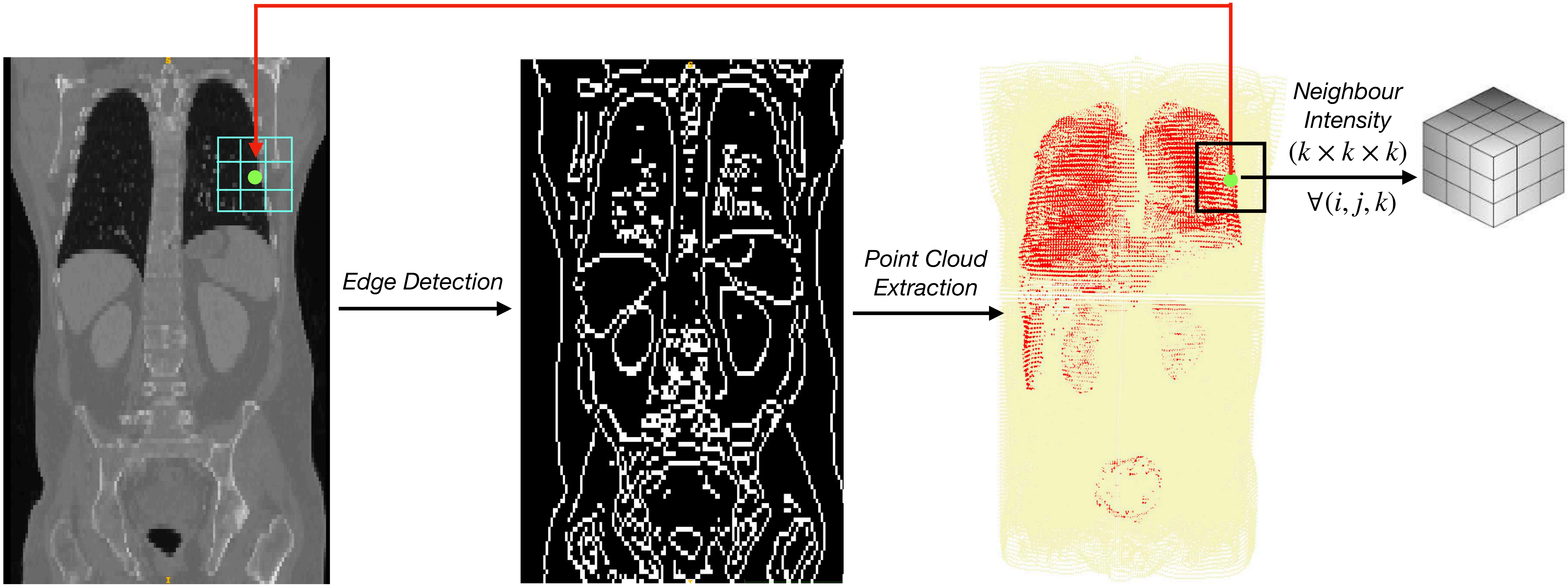


Fig 3: 2D slices of a visceral image from different views with ground truth segmentation mask

*https://visceral.eu/closed-benchmarks/anatomy3/

Data Preparation - Point Cloud Extraction



(1) Volume Image **(2) Edges** **(3) Point Cloud**$((N,3), (N, k, k, k))$

Model - RandLaNet

- **RandLaNet**: UNet-like architecture for large scale point cloud segmentation.
- It uses *random downsampling* method to learn features from small representative samples.
- It uses *attention-based local feature aggregation(LFA)* to counter the loss of points.

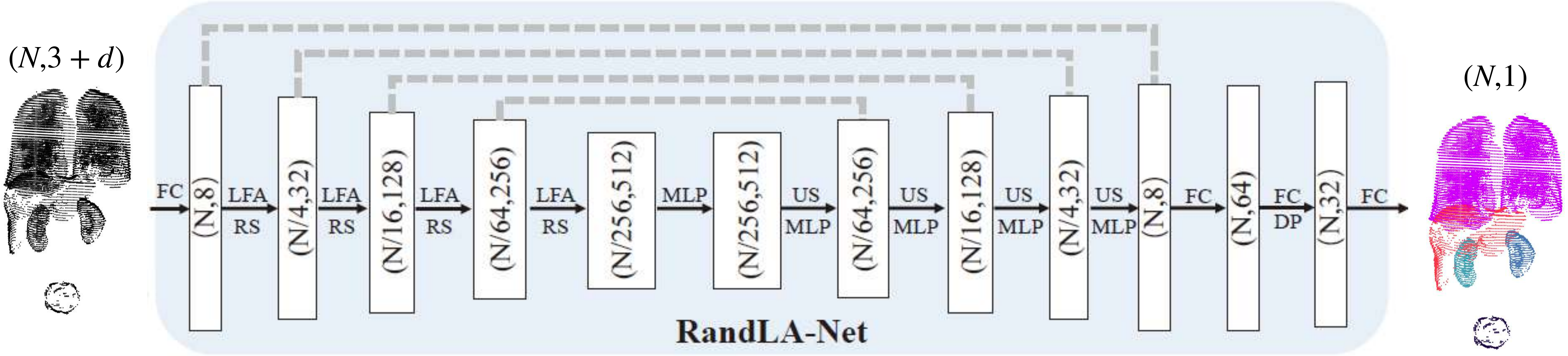
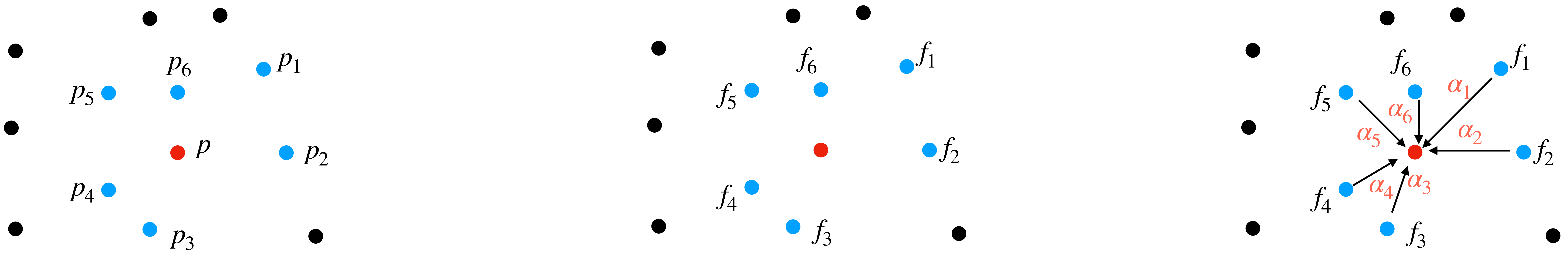


**Fig 4: RandLaNet Architecture (Source*). RS is Random Sampling and US is UpSampling.
For ease of visualisation, background points are removed from the point cloud.**

******RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds***

RandLaNet - Encoder

Local Feature Aggregation (LFA)*



1. Sampling using KNN

2. Message Generation

$$f_i = MLP\left(p; p_i; (p - p_i); ||p - p_i||\right)$$

3. Message Passing

$$f = \sum_{i=1}^{6} \alpha_i f_i$$

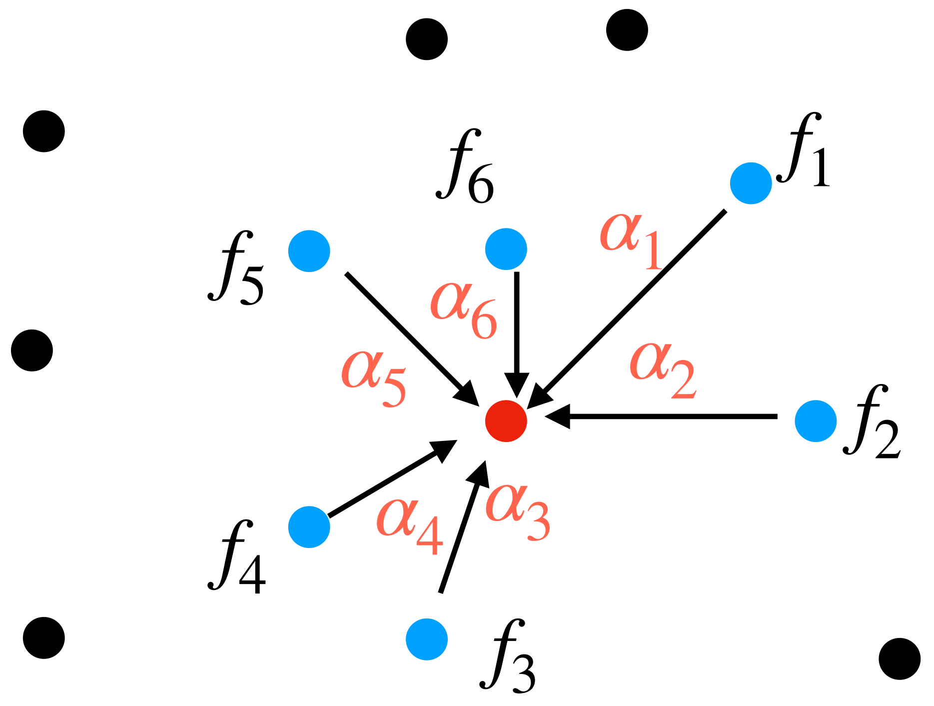
Fig 5: Feature sharing in RandLa-Net*

[RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds](#)

RandLaNet - Encoder

Local Feature Aggregation (LFA)*

Fig 6: LFA Module



$$f = \sum_{i=1}^{6} \alpha_i f_i$$

$$E = MLP\left(p_i; p_k^{(i)}; (p_i - p_k^{(i)}); ||p_i - p_k^{(i)}||\right)$$

Fig 7: Feature sharing in RandLa-Net

*RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds

RandLaNet - Decoder

Upsampling

- In every encoder when a point is removed, it is stored as a reference. In the subsequent decoder we upsample the points to match the number of points equal with the connected encoder.

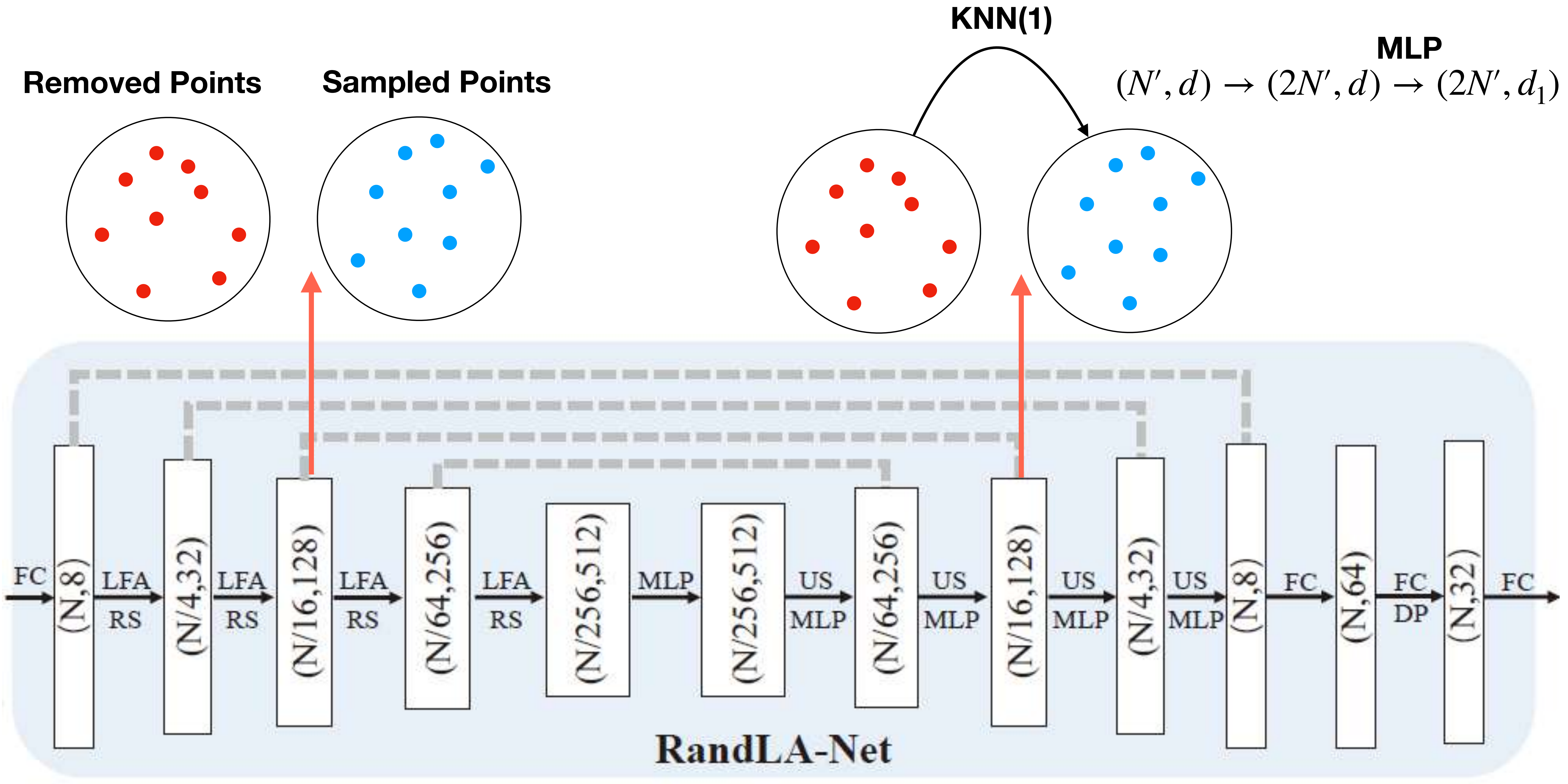


Fig 8: RandLaNet Architecture (Source*). RS is Random Sampling and US is UpSampling

*RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds

Modified RandLaNet

RandLaNet + Feature Extractor (FE)

- **Challenges**: Model does not learn good local features.
- **Contribution**: A Feature Extractor layer to learn the *local spatial information* using the neighbourhood intensity values.

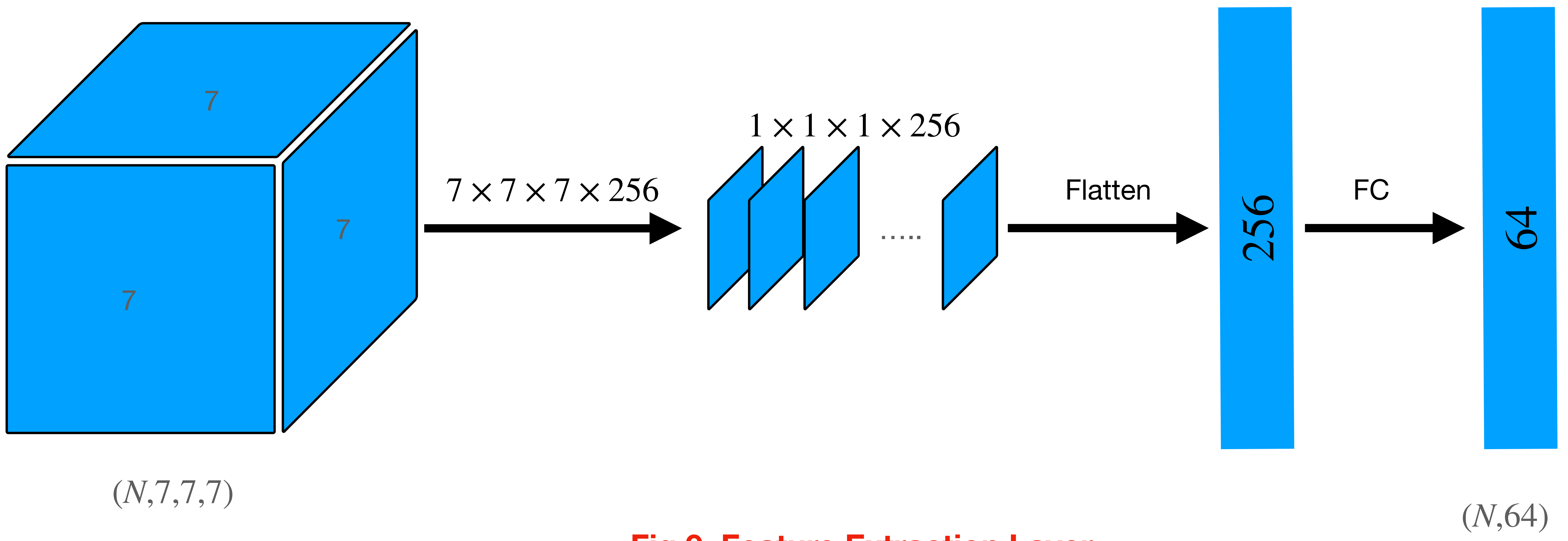


Fig 9: Feature Extraction Layer

Modified RandLaNet

RandLaNet + Feature Extractor (FE)

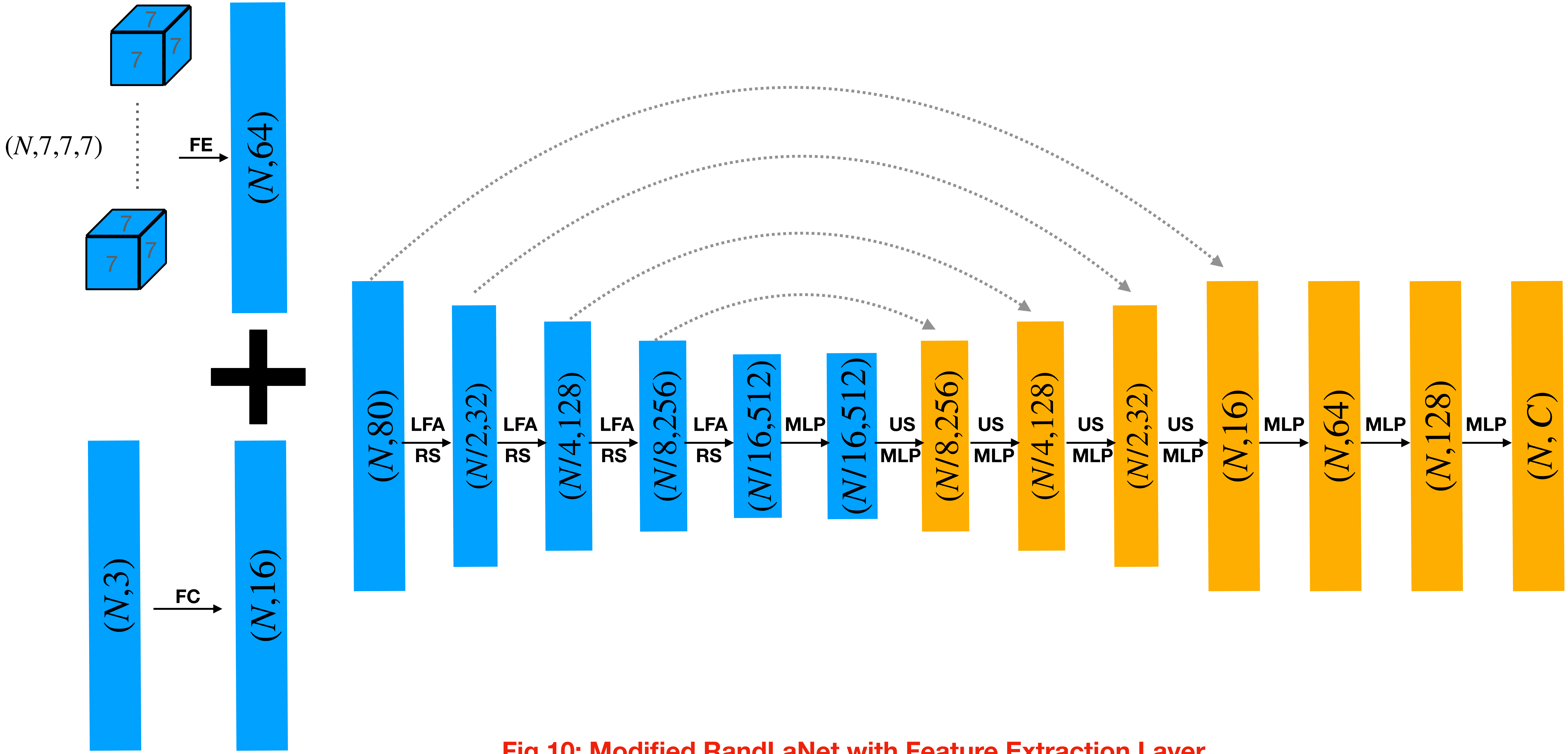


Fig 10: Modified RandLaNet with Feature Extraction Layer

Point Cloud Segmentation Result

Feature Extraction (Binary Segmentation)

$$Recall = \frac{TP}{TP + FN}$$

$$IoU = \frac{TP}{TP + FN + FP}$$

Experiment	Recall		IoU	
	Background	Lungs	Background	Lungs
RandLaNet	0.9568	0.7936	0.9348	0.5702
RandLaNet+ FE	0.9844 (+2.8%)	0.8872 (+11.8%)	0.9722 (+4%)	0.7766 (+36.2%)

Table 1: Effect of Feature Extractor on feature learning for lungs segmentation. Loss is CMCE. Blue is for better result.

- Our proposed feature extraction layer learns local geometrical information for all the points.
- It performs better than original RandlaNet model.

Experiment	Recall		IoU	
	Background	Liver	Background	Liver
RandLaNet	0.9830	0.6008	0.9730	0.3426
RandLaNet+ FE	0.9924 (+1%)	0.76 (+26.5%)	0.9862 (+1.4%)	0.5652 (+65%)

Table 2: Effect of Feature Extractor on feature learning for liver segmentation. Loss is CMCE. Blue is for better result.

Point Cloud Segmentation

Class Imbalance

- **Dataset Problem:** Class Imbalance
- **Experiments**: Three types of weighting in Cross Entropy Loss.

- **ComboLoss** where $w = \alpha_1 + \frac{\alpha_2}{r + 0.02} + \frac{\alpha_3}{r}$, r=ratio. $\sum_{i=1}^{3} \alpha_i = 1.$

- **Challenges with Frequency based weight**$(w = \frac{1}{r})$: High misclassification rates in background classes for *lower weights of background compared to minority organs.*

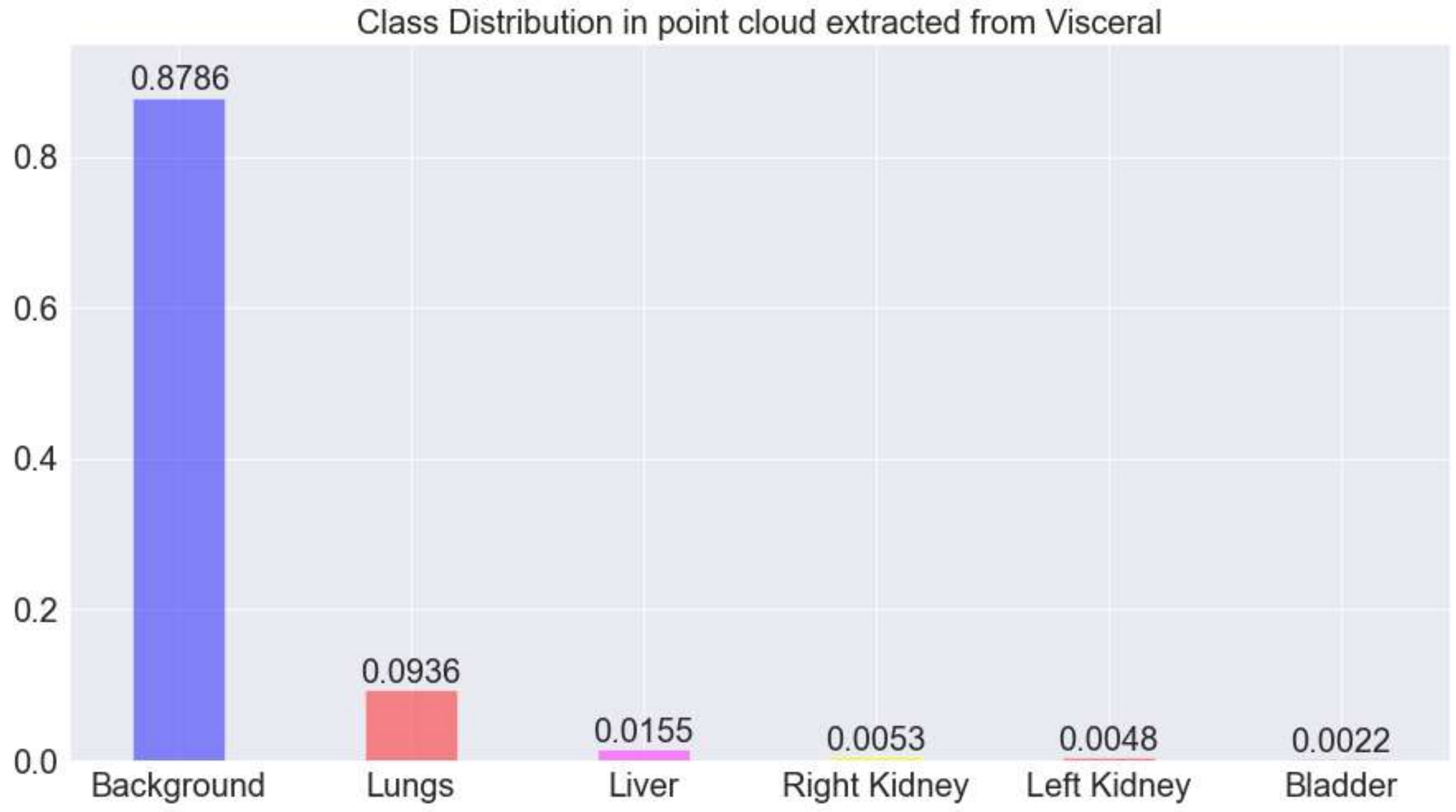


Fig 11: Point Cloud Dataset Class Distribution

Point Cloud Segmentation

Cost Matrix Cross Entropy loss function

Background

Model

0.2	0.06	0.04	0.5	0.1	0.1	**Output Probability**
Background	Liver	Lungs	Bladder	Left Kidney	Right Kidney	**Label**
1.1	24.2	6.81	125	97.9	104	**Weights**

CE

$$-1.1 \times log(0.2)$$

CMCE

$$-125 \times log(0.2)$$

Fig 12: Motivation behind CMCE loss.

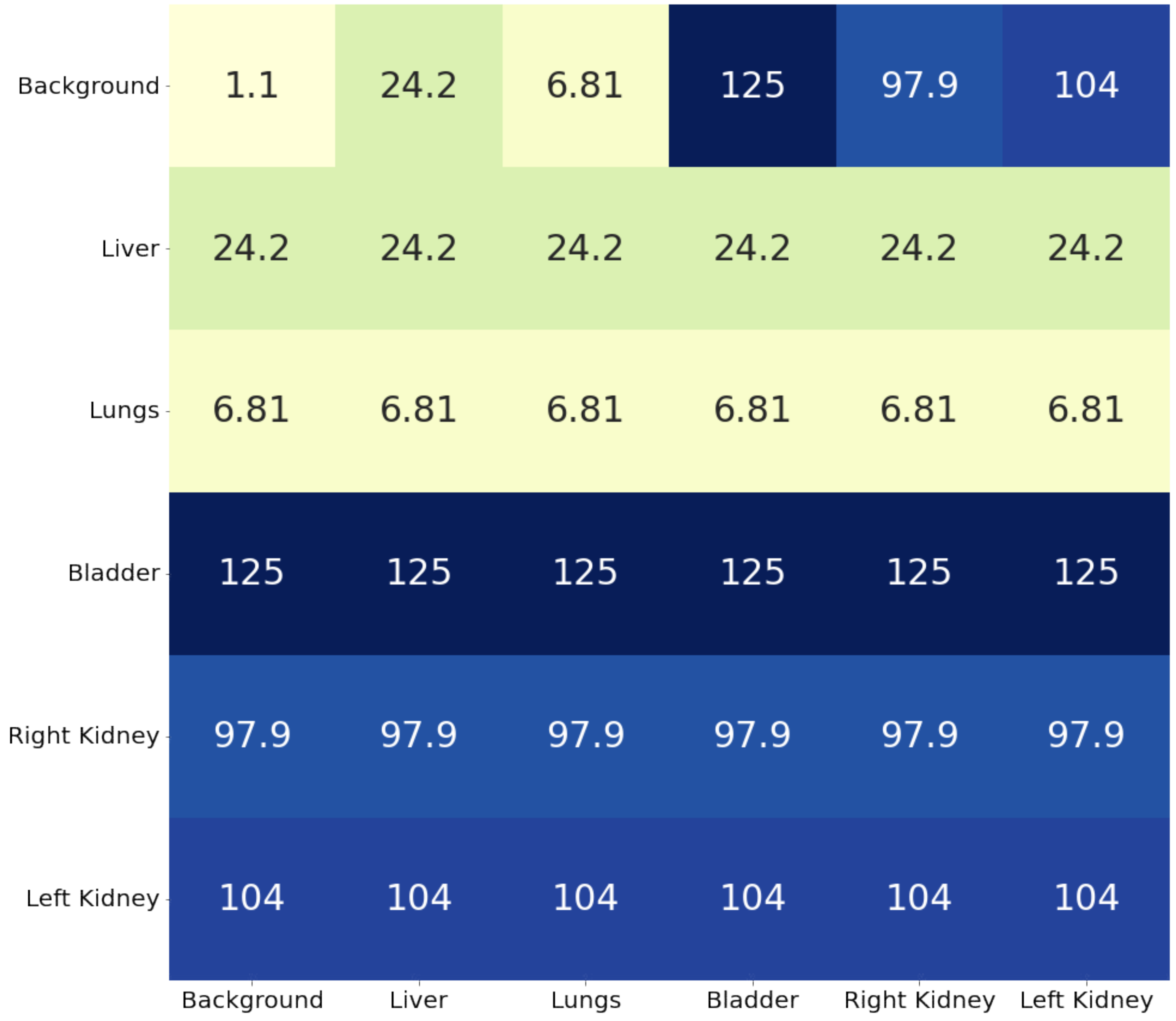


Fig 13: Cost Matrix with prediction cost. Weights are calculated using ComboLoss.

Point Cloud Segmentation

Result: Loss Function (Multi-Class Segmentation)

Before CMCE

Actual Label \ Predicted Label	Background	Liver	Lungs	Bladder	Right Kidney	Left Kidney
Background	185097	6217	7905	3844	2176	1956
Liver	512	4736	160	0	5	215
Lungs	1638	472	21842	0	4	2
Bladder	103	0	0	757	0	0
Right Kidney	82	0	1	0	1031	0
Left Kidney	71	62	0	0	0	916

Figure 14: Confusion Matrix for Multi-Class Segmentation with RandLa-Net+FE *before applying CMCE.*

After CMCE

Actual Label \ Predicted Label	Background	Liver	Lungs	Bladder	Right Kidney	Left Kidney
Background	195830	2695	6747	704	677	540
Liver	1365	4010	120	0	0	133
Lungs	2419	189	21351	0	0	0
Bladder	564	0	0	296	0	0
Right Kidney	302	0	0	0	812	0
Left Kidney	292	25	0	0	0	732

Figure 15: Confusion Matrix for Multi-Class Segmentation with RandLa-Net+FE *after applying CMCE*.

Point Cloud Segmentation

Result: Loss Function (Binary Segmentation)

Experiment	Recall		IoU	
	Background	Lungs	Background	Lungs
RandLaNet+FE+ CE(No Weights)	0.99	0.88	0.9776	0.806
RandLaNet+ FE+ComboLoss	0.9642	0.979	0.9620	0.7372
RandLaNet+FE+ CE(CMCE)	0.9844	0.8872	0.9722	0.7766

Table 3: Results of Lungs Segmentation for different weights in CE Loss

Experiment	Recall		IoU	
	Background	Liver	Background	Liver
RandLaNet+FE+ CE(No Weights)	0.996	0.6918	0.9878	0.5748
RandLaNet+FE+ CE(Frequency Weights)	0.9706	0.9562	0.9692	0.4210
RandLaNet+FE+ CE(CMCE)	0.9924	0.76	0.9862	0.5652

Table 4: Results of Liver Segmentation for different weights in CE Loss

$$Recall = \frac{TP}{TP + FN}$$

$$IoU = \frac{TP}{TP + FN + FP}$$

Point Cloud Segmentation

Multi Class Segmentation Results

Index	Model	Metric	Background	Liver	Lungs	Bladder	Right Kidney	Left Kidney	Mean
1	RandLaNet + CE(w=1/ratio)	Recall	0.486	0.803	0.832	0.235	0.973	0.654	0.523
		IoU	0.482	0.12	0.271	0.024	0.035	0.104	0.173
2	RandLaNet+FE(7,7,7)+CMCE+ Random Downsampling Rate(2)	Recall	0.9446	0.8112	0.9118	0.4564	0.7924	0.852	0.795
		IoU	0.9282 (+93%)	0.4964 (+314%)	0.7488 (+176%)	0.2266 (+844%)	0.4006 (+1044%)	0.3978 (+283%)	0.533 (+208%)

Table 5: Final improvement of result for multi-class segmentation with all the modifications.

Voxel Segmentation

One-to-one mapping

- Our final task is to segment all the voxels in the original $128 \times 128 \times 112$ image in Visceral dataset.
- Every classifier has a decision boundary. The decision boundary of RandLaNet is the shape of the organ.

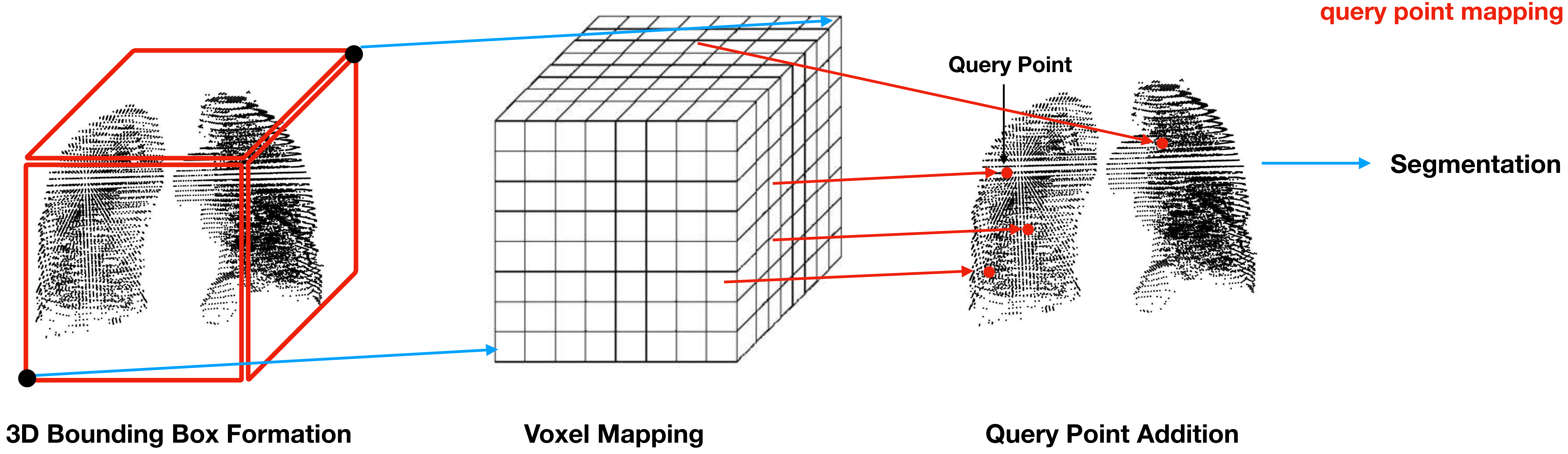


Figure 16: Framework of query point mapping

Voxel Segmentation

Results for Lungs

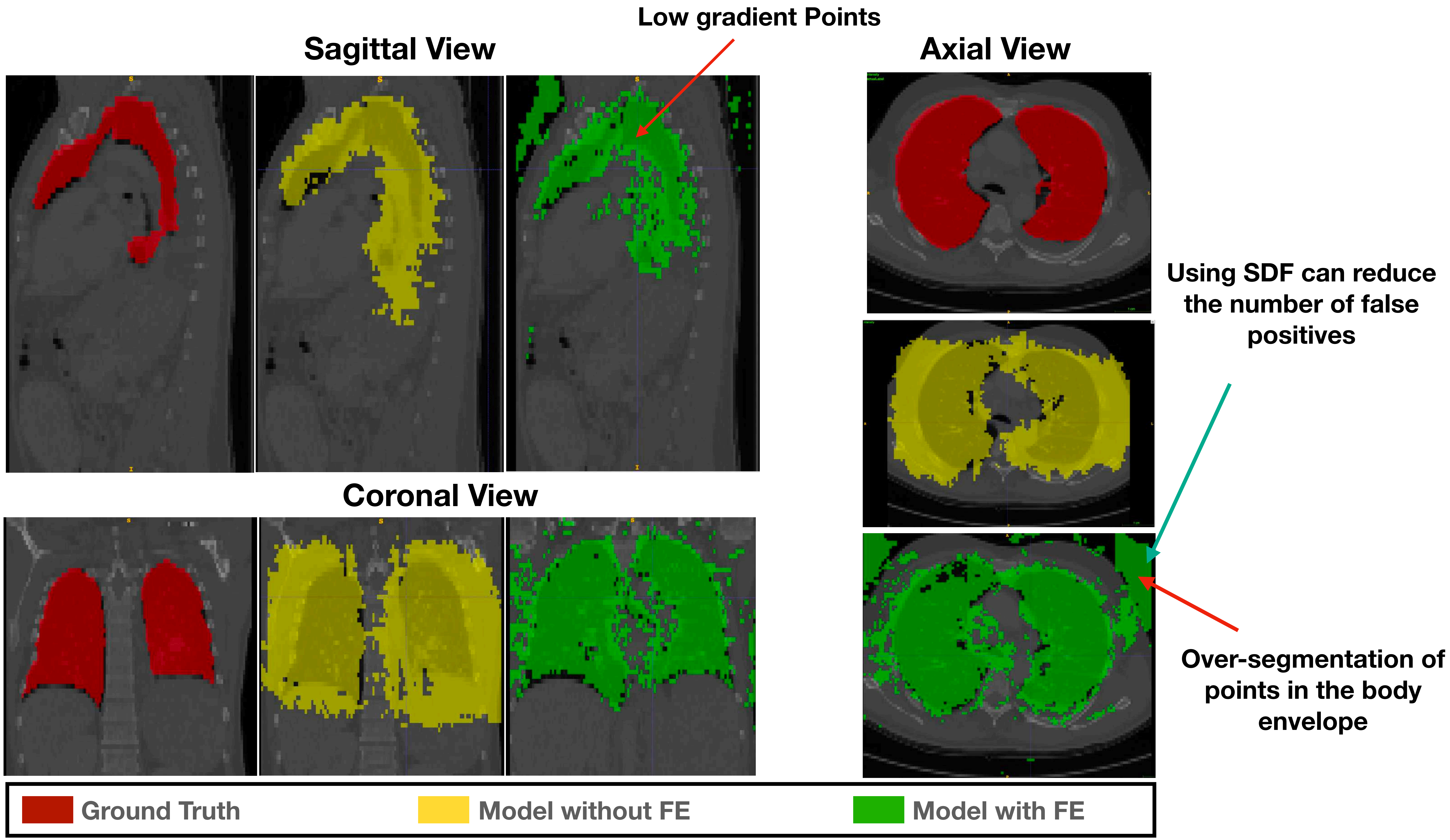


Figure 17: Voxel Segmentation for Lungs

Future Work

- *Learnable weights* for weighted cross entropy*.
- Signed Distance Function for *smooth shape prior information* to reduce over-segmentation**.
- Use loss function to constraint the latent vector. (Reduce variance in intra-class and increase distance between inter-class latent vectors)***.

Cost-Sensitive Learning of Deep Feature Representations from Imbalanced Data

** *Shape-Aware Organ Segmentation by Predicting Signed Distance Maps*

****3D Instance Embedding Learning with a Structure-Aware Loss Function for Point Cloud Segmentation*

Thank You

Appendix

Experiment: Latent Vector

- Since all the weighting based CE methods didn't improve the Intersection Over Union Results(IoU) much for multi-class segmentation.
- So we wanted to verify if the latent vectors leant in the network are discriminative for different classes.
- The downsampling step reduces the number of points in latent space. Hence the global features for an organ is influenced by only a small number of points.
- So we plotted inter-class and intra-class difference of the latent vectors for lungs.

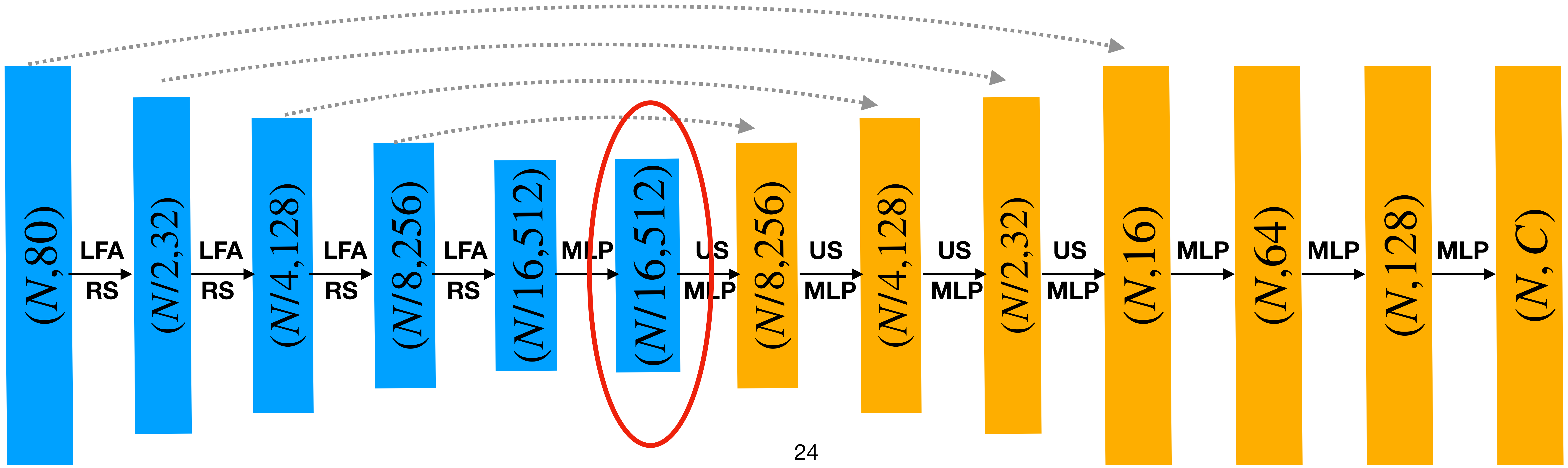


Figure 18: Latent vectors

Appendix

Latent Vector Difference Calculation

How the difference is calculated

Lungs Embedding

Lungs Embedding

Lungs Embedding

Lungs Embedding

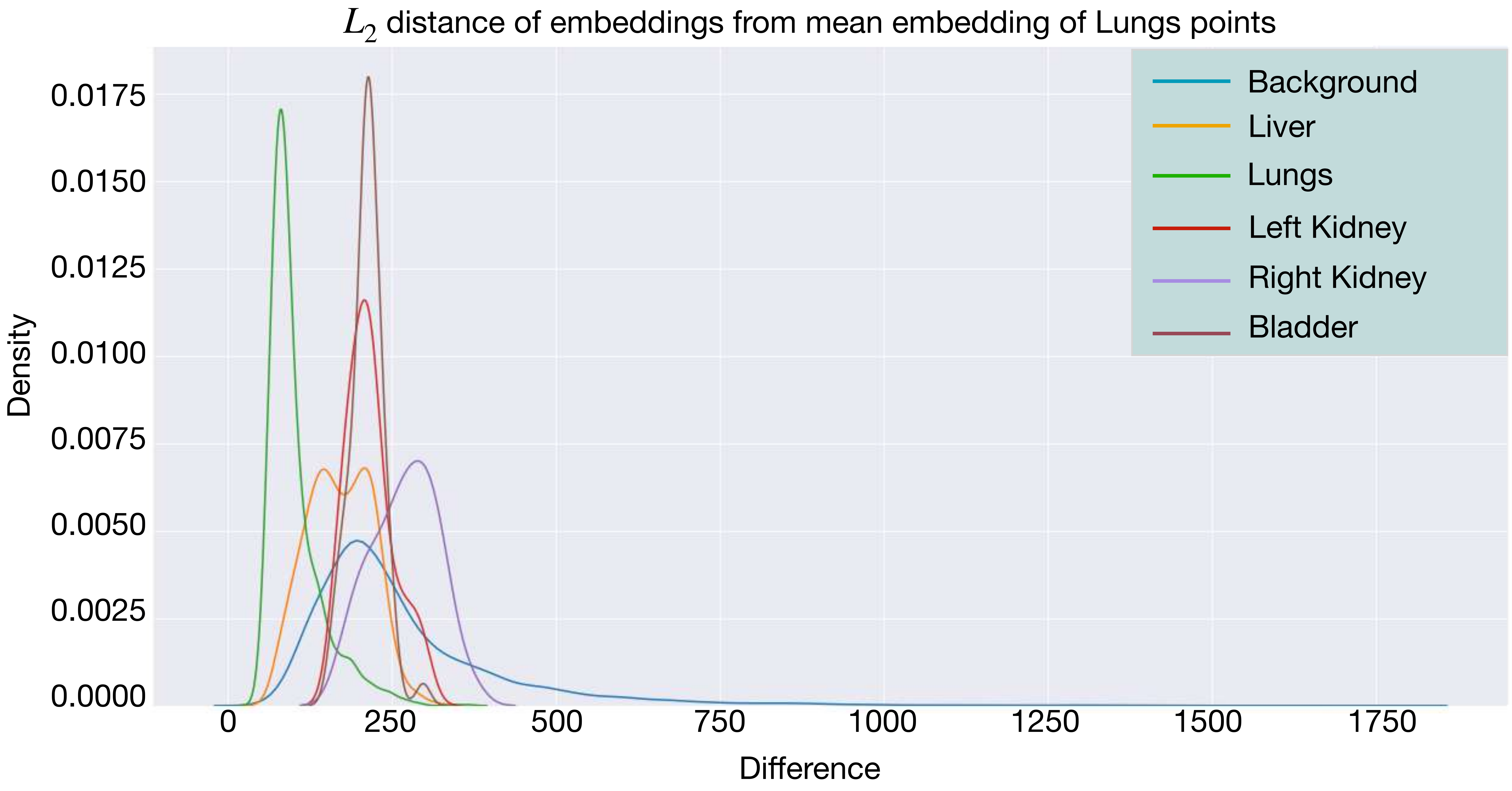
Mean Embedding Lungs

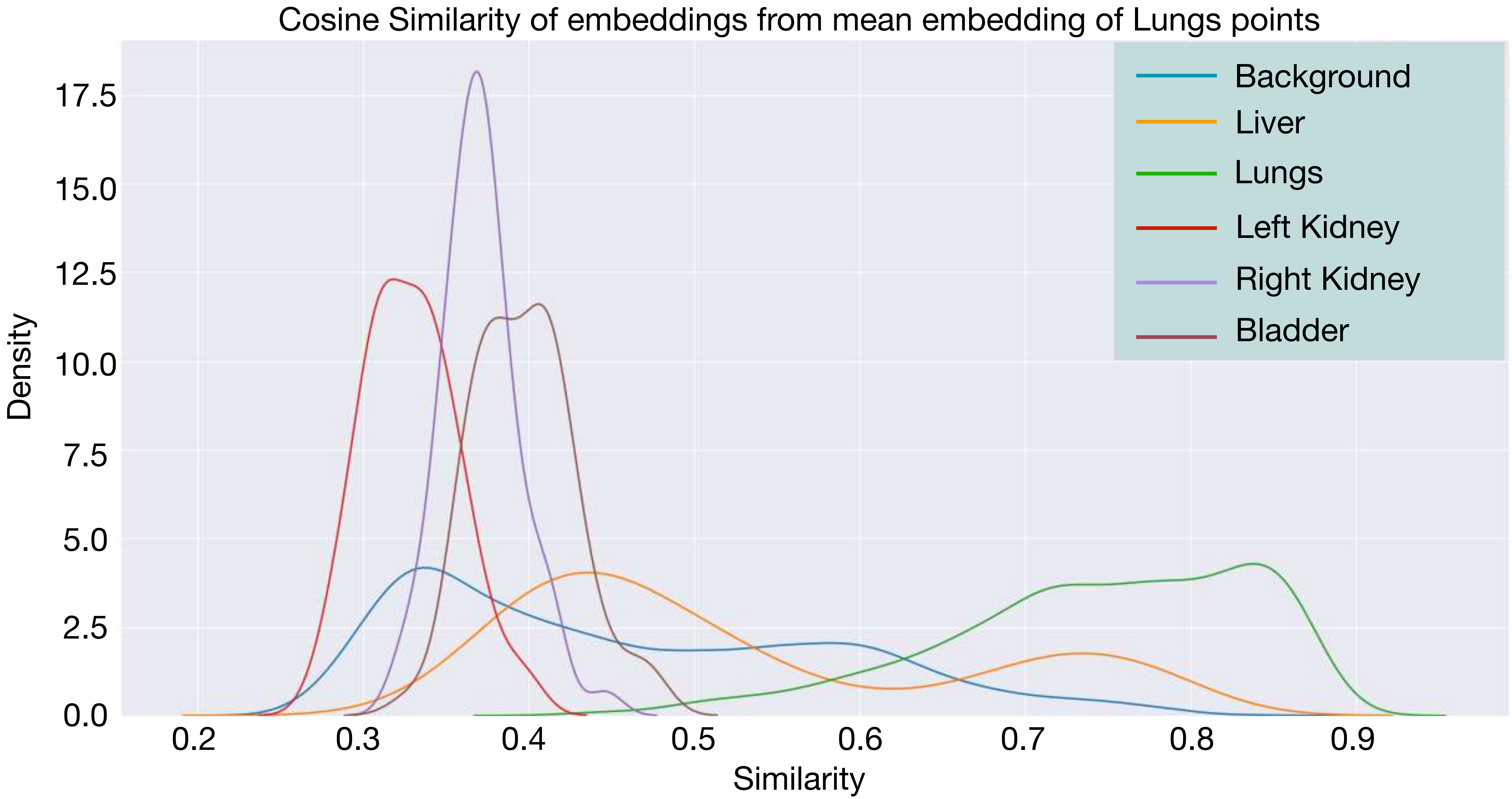
Background Embedding

Lungs Embedding

Liver Embedding

Bladder Embedding

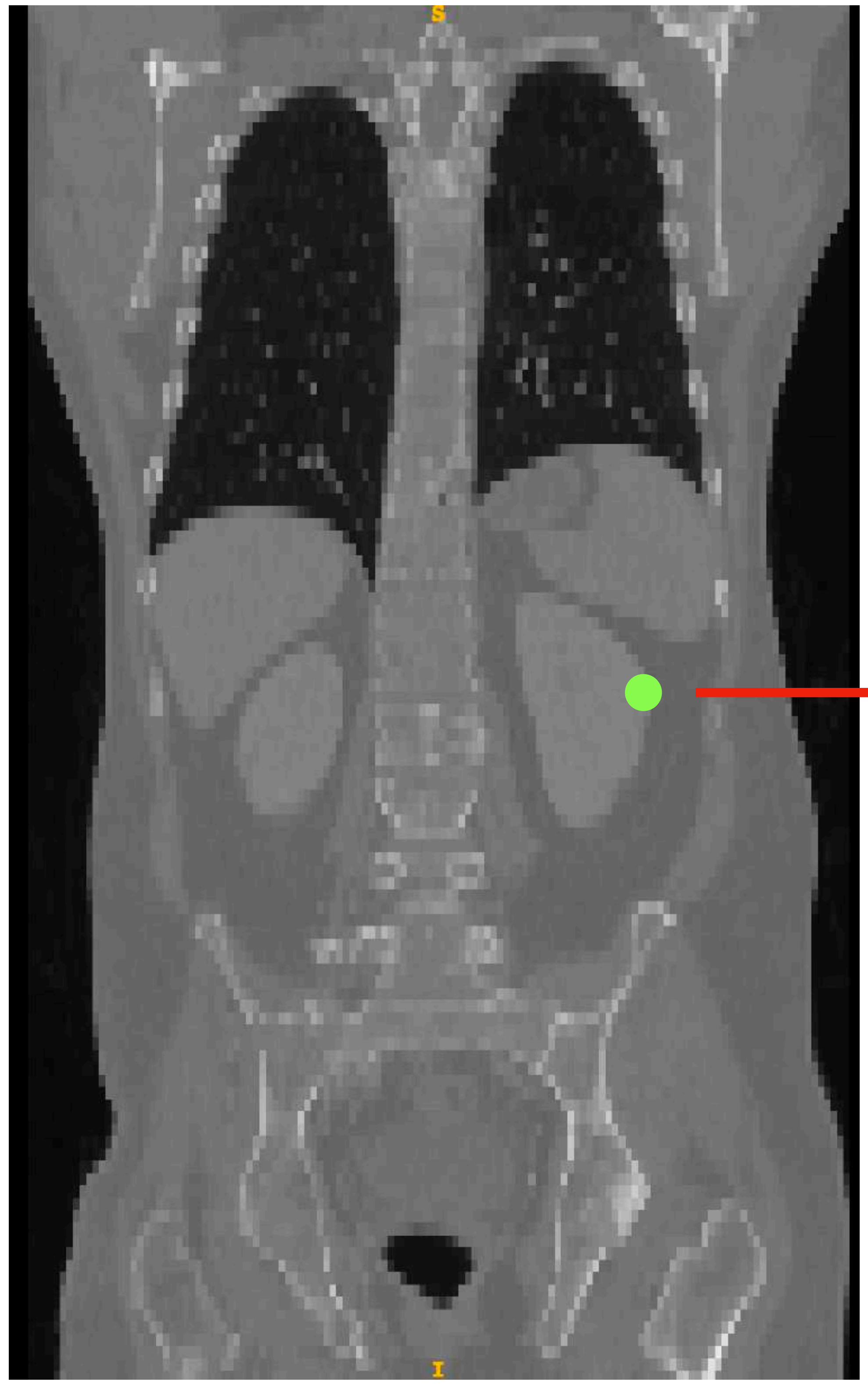

L_2 distance of embeddings from mean embedding of Lungs points
Density
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
250
500
750
1000
1250
1500
1750
Difference
Background
Liver
Lungs
Left Kidney
Right Kidney
Bladder



Cosine Similarity of embeddings from mean embedding of Lungs points
Background
Liver
Lungs
Left Kidney
Right Kidney
Bladder
Density
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Similarity

Appendix

Reference Space



$$f(i, j, k) = R[i, j, k] + T$$

Voxel Space
(i, j, k)

Object Space
$f(i, j, k)$