UNIVERSITÀ DEGLI STUDI DI MILANO

POLITECNICO MILANO 1863

Learning self-supervised representations of whole body MRI for phenotype imputation

Tesi di Laurea Magistrale in
Bioinformatics for computational genomics

Author: **Emanuele Bianchi**

Student ID: 985902-995878
Advisor: Dr. Craig A. Glastonbury
Co-advisors: Prof. Matteo Matteucci
Academic Year: 2022-2023

Abstract

Large-scale non-invasive medical imaging datasets at the scale of a biobank present a remarkable opportunity to explore and understand the inherent variations in human organs. Nonetheless, annotating such large datasets poses a significant challenge. Hence, this thesis investigates the use of both unsupervised (VAEs) and self-supervised learning frameworks (DINOv2), in extracting meaningful embeddings from whole-body MRI scans within the UK Biobank dataset.

This thesis is motivated by the necessity of overcoming the limitations of supervised learning on these large biomedical datasets, by exploring whether self-supervised learning approaches are able to learn rich embeddings that can be used for multiple downstream tasks. A notable contribution of this work is the development of a customised self-supervised learning pipeline tailored to efficiently analyse a large 3D MRI dataset. Additionally, the study addresses the challenge of adapting SSL architectures to accommodate the unique attributes of MRI data compared to commonly used natural image benchmarks, notably their size and 3D nature.

The main findings underscore the potential of self-supervised learning in generating embeddings that capture meaningful morphological information from MRI scans. These embeddings demonstrate the ability to identify various anatomical structures, potentially improving our understanding of pathological characteristics. Further, this thesis demonstrates the embeddings' applicability in downstream tasks, such as their use in genetic association studies, thereby extending their utility beyond just image analysis to contribute to our understanding of the genetic underpinnings of complex traits and diseases.

This research not only showcases the adaptability and potential of self-supervised learning models in medical imaging analysis but also sets a precedent for future studies aiming to leverage unlabelled data in biomedical research. The successful application of DINOv2 to whole-body MRI data signifies a step forward in the intersection of artificial intelligence and medical imaging, offering promising avenues for enhancing diagnostic, prognostic, and genetic research.

Contents

Abstract **i**

Contents **iii**

1 Introduction **1**

1.1 Deep learning Fundamentals . . . 1

1.1.1 Perceptron and Neural Networks . . . 3

1.1.2 Multi Layer Perceptron . . . 4

1.1.3 Activation functions . . . 6

1.1.4 Backpropagation . . . 7

1.1.5 RNN and LSTM . . . 9

1.1.6 CNN . . . 11

1.2 Learning Paradigms . . . 13

1.3 Modern-day Deep learning . . . 14

1.3.1 Attention . . . 15

1.3.2 Transformers . . . 16

1.3.3 Vision Transformers (ViTs) . . . 17

1.3.4 Self-supervised Learning paradigms . . . 18

1.4 Applications of Machine Learning to biomedical imaging . . . 20

1.4.1 Magnetic Resonance Imaging . . . 21

1.4.2 DIXON MRI . . . 23

1.5 Complex Traits Genetics . . . 24

1.6 The UK Biobank . . . 25

1.6.1 The UKBB DIXON Full Body MRI . . . 26

1.7 Related Work . . . 27

2 Methods **29**

2.1 Data . . . 29

2.1.1 UKBiobank . . . 29
2.1.2 Clinical Data . . . 29
2.1.3 Whole body MRI . . . 30
2.1.4 2D training . . . 31
2.2 Image embeddings . . . 32
2.2.1 Factor VAE . . . 32
2.2.2 DINO . . . 33
2.2.3 DINOv2 . . . 35
2.3 Encoder evaluation . . . 38
2.4 embedding analysis . . . 39
2.5 GWAS . . . 39

3 Results **43**
3.1 Overview . . . 43
3.2 Variational Autoencoders fail to capture whole body information . . . 43
3.3 self-supervised learning results in improved latent representation quality . . . 47
3.4 Custom implementation of DINOv2 to improve representations . . . 49
3.5 Whole body aggregated embeddings are predictive of relevant traits . . . 51
3.6 GWAS on Abdominal Traits . . . 56

4 Conclusion **65**

5 Appendix **67**

Bibliography **93**

1 Introduction

1.1. Deep learning Fundamentals

AI, Machine Learning and Deep Learning

Artificial Intelligence (AI), Machine Learning (ML), and Deep learning (DL) are terms often used somewhat interchangeably. This is not completely erroneous, as they are related through a hierarchical structure, with AI being the broadest term and deep learning the most narrow. However, since there are subtle differences, it is worth spending time to define each term and discuss their origins.

Artificial Intelligence (AI)

The term Artificial Intelligence can be applied in an extremely broad way to all applications that are able to solve complex tasks, which previously relied on human intervention. A milestone for artificial intelligence was achieved in 1997 when the chess world champion Garry Kasparov was defeated by the computer system Deep Blue. Whilst impressive, Deep Blue was based on brute force computation, in which it enumerated all possible outcomes in a game of Chess. For a better understanding of AI a definition of what we mean by machine intelligence is needed. Alan Turing during the 1940's hypothesized about whether if machines could think, could a test be devised to determine this, now known as the Turing test: If a machine could carry out a conversation (over a teleprinter) that was indistinguishable from a conversation with a human being, it is reasonable to say that the machine is "thinking"[55].

Machine Learning (ML)

The field of Machine Learning was born during the first AI research boom during the 1950's, Machine Learning was used to develop intelligent machines that could learn to generalize and predict new outcomes in an automatic way by simply learning from data. The AI and Machine learning research of the second half of the 20th century set up

the foundation for the AI boom of the 2010. However the primitive models of this period often didn't achieve the expected outcomes and largely relied on handcrafted, user defined features. The lack of concrete results over competitive baselines at the time, led to a so called "AI-winter", a period in which funding and interest for AI were greatly diminished.

Deep Learning (DL)

The "Deep" part of the Deep Learning refers to the number of layers stacked one on top of the other. Each layer of a deep learning model, consists of weights and biases trained on a distinct set of features that are based on the previous layer's output. The more layers (or depth) a network has, the more complex the features it can learn to recognize, because each layer builds on the previous one to refine and combine inputs into more abstract representations. This can be thought of as a composite function:

$$F(X) = g_n(f_n(\dots g_2(f_2(g_1(f_1(X, W_1), W_2), \dots, W_{n-1})), W_n)) \tag{1.1}$$

Where:

- **Input** X: The raw input data vector to the model.
- **Layers** (f_i): Each f_i represents a transformation layer in the neural network, applying operations like weighted sums of its inputs or matrix multiplications, using the layer weights W_i.
- **Weights** (W_i): Parameters of each layer, learned during the training process. These weights define the specific transformation that the layer applies to its input.
- **Activation Functions** (g_i): Non-linear functions applied after each transformation f_i, enabling the model to learn and represent non-linear relationships. Common examples include ReLU (Rectified Linear Unit), sigmoid, and tanh functions.
- **Output** Y: The final output of the model, which can vary depending on the task, such as a single value for regression tasks, a vector of class probabilities for classification tasks, or another type of structured output.

Whilst Deep Learning research was already taking place as early as the 1970's, it started to gain traction and show competitive results only 15 years ago. Several key advances from 1980 to 2009 were critical for this, namely: The introduction of back propagation, significant improvements in optimisation algorithms based on gradient descent, the commercialisation of GPUs and the wide availability of data, particularly images.

As computational cost got cheaper another essential development was the exponential

increase in available, well curated data. The growth of the internet and large amounts of data being shared on it, allowed researchers to assemble comprehensive datasets to train these deep-learning models. One notable example of this is the ImageNet, published in 2009, which is composed of 14 million images. ImageNet was introduced to offer a standardised benchmark to the computer vision deep-learning community, and is still widely used today to benchmark image classification performance.

During the 2010's Deep-learning had a exponential evolution. One of the pivotal moments of this era was the success of deep convolutional neural networks (CNNs) in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, where a model called AlexNet significantly outperformed traditional computer vision models such as Support Vector Machines (SVMs) or decision trees, trained using image derived features such as Scale Invariant Feature Transforms (SIFT) or Histogram of Oriented Gradients (HOG).

1.1.1. Perceptron and Neural Networks

The beginnings of **Neural networks** can be traced back to 1943 when Warren McCulloch and Walter Pitts proposed the idea that biological neurons can be described as a computational device [42]. A neuron can be divided into three components: the **cell body** containing the nucleus, the **dendrites**, which receive inputs from other neurons via synapses, and an **axon**, which conducts electrical impulses away from the cell body, towards other neurons through its synaptic terminals.

Starting from the biological they proposed a mathematical formulation for an artificial neuron.

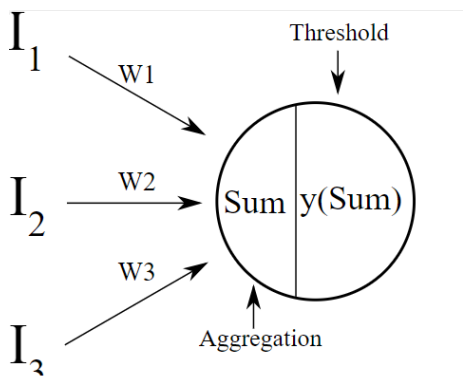


Figure 1.1: A schematic of a multiple perceptron neuron (MP-neuron), receiving inputs and performing a weighted aggregation and thresholding operation.[9]

Expanding the original definition, the output of the neuron can be defined as:

$$out = g \circ f(\overline{I} \odot \overline{W})$$

where $\overline{I} = (I_1, ..., I_n)$ is a set of inputs, $\overline{W} = (W_1, ..., W_n)$ is a set of weights, **f** is an aggregation function and **g** is an activation function. For the MP-neuron the aggregation function is the weighted sum of the inputs, which can be written in vector notation as $f = \overline{I} \cdot \overline{W}$, while the activation function is a threshold function:

$$g((f(I; W))) = \begin{cases} 1 & \text{if } I > \theta \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

McCulloch-Pitts's definition only allowed for the inputs, weights and outputs to take on binary values and the weights and threshold value had to be manually programmed to perform a task, without the use of a learning algorithm.
During the 1950s, the field of Artificial Intelligence had it's first boom, and the idea of artificial neural networks was proposed again through the 1958 Frank Rosenblatt's **Perceptron** [47]. The equation describing the perceptron is:

$$\hat{y} = \begin{cases} +1 & \text{if } (\overline{W} \cdot \overline{I}) + b > 0 \\ -1 & \text{otherwise} \end{cases}$$

While it shares with the MP-neuron the aggregation and activation function, both the input $\overline{I}$ and the weights $\overline{W}$ are real valued in the Perceptron, and an additive bias term is added to the output of the weighted sum.

Rosenblatt proposed what he defined as an error-corrective reinforcement learning procedure. Rosenblatt proved mathematically, that if a solution exists (i.e., if the data are linearly separable), the perceptron algorithm is guaranteed to converge to a set of weights that correctly classifies all training examples. However, it's important to note that this guarantee does not hold for data that is not linearly separable, which led to the development of more complex models capable of learning non-linear relationships, such as multi-layer perceptrons and other deep learning architectures.

1.1.2. Multi Layer Perceptron

The Perceptron, initially introduced as a single-layer network with one neuron, offers a foundational architecture for neural networks. By enhancing this architecture through

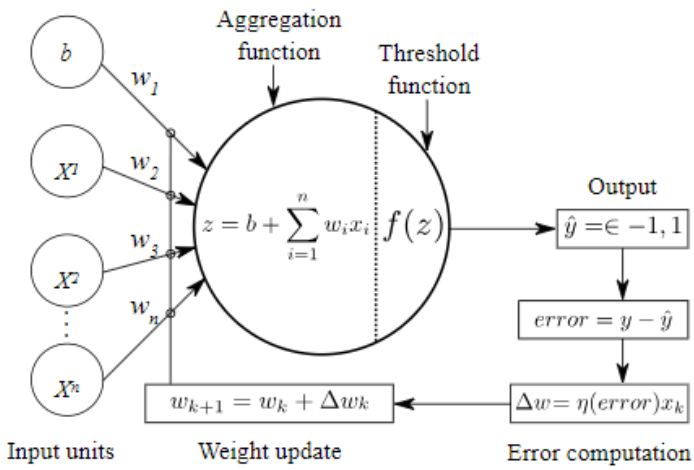


Figure 1.2: Perceptron[9]

the addition of more neurons within a layer—each learning distinct weighted sums of the inputs—or by incorporating additional layers to form a hierarchy, where the output of one layer feeds into the next, we arrive at the concept of Multi-Layer Perceptrons (MLPs). Rosenblatt, the originator of the perceptron concept, acknowledged the potential of MLPs in his early works [47], and even proposed a method for back-propagating errors to train them [46]. However, he did not succeed in developing a comprehensive algorithm for training MLPs of arbitrary depth[50]. Throughout the latter half of the 20th century, progress in MLP research continued. By 1967, Shunichi Amari demonstrated that MLPs trained via Stochastic Gradient Descent (SGD) could tackle problems that were not linearly separable[3][50]. The advent of Backpropagation in the 1970s and 1980s, culminating in the work by Rumelhart and Hinton in 1985, showcased a universal method for applying backpropagation to train MLPs effectively[49]. A pivotal moment in the understanding of neural networks came in 1989, when Hornik and Maxwell established the universal function approximation theorem. This theorem illuminated the capability of single hidden layer feedforward neural networks, equipped with non-linear activation functions, to approximate any measurable function to any desired degree of accuracy within a compact set[28]. This profound insight underlines the theoretical foundation for the versatility and power of neural networks, positioning MLPs as a central architecture in the field of deep learning for their ability to model complex and diverse functions.

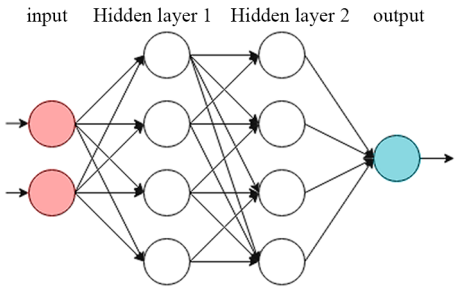


Figure 1.3: 4 layer deep MLP

Mathematically, the output of a single neuron in one layer that is part of an MLP can be described as follows:

$$\hat{h} = \phi(\sum_{i=1}^{n} w_i x_i + b)$$

The whole layer can be described with vector notation as:

$$\overline{H} = \begin{bmatrix} \phi_1(\sum_{i=1}^{n} w_{1i} x_i + b_1) \\ \vdots \\ \phi_m(\sum_{i=1}^{n} w_{mi} x_i + b_m) \end{bmatrix} = \Phi(WX + b)$$

Where n is the number of inputs, m the number of neurons, W the weight matrix and X the input matrix. ϕ is the activation function.

The output of the MLP is then:

$$\hat{y}(X|W, b) = H_m \circ H_{m-1} \circ \ldots \circ H_1(WX + b)$$

1.1.3. Activation functions

As mention before a non-linear activation function provides the flexibility for an MLP to act as a universal function approximator. The oldest function used for this purpose are the Sigmoid and the Hyperbolic tangent(TanH). These function provide an output in the range $(0, 1)$ and $(-1, 1)$ respectively. The following equation represent a Sigmoid:

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

While TanH is:

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Several other activation functions are plotted in figure 1.4

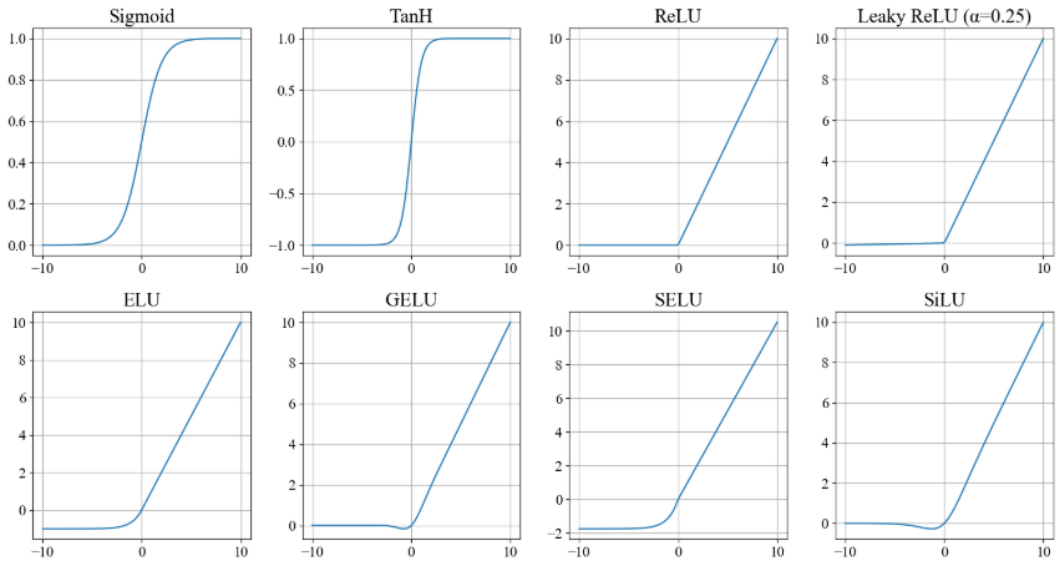


Figure 1.4: Examples of different, commonly used activation functions[14]

In 2010 Nair and Hinton [44] introduced a new activation function defined as:

$$\phi(x) = max(0, x)$$

They called this function the Rectified Linear Unit or ReLU. As of 2024 it is still the most popular activation function in the field of neural networks, with some small variations being introduced (e.g. ELU, GELU, SELU etc) to avoid the problem of "dying neurons", which just output zero. One of the primary reasons ReLUs became so popular, is because they significantly reduce the issue of vanishing gradients; for large input values the gradient of both Sigmoid and TanH becomes close to 0, whilst the ReLU's gradient will have a constant value for all input above 0. Another benefit is its computational speed, as there are no expensive exponential operations, it's much faster than both the Sigmoid or TanH operation.

1.1.4. Backpropagation

In the process of training modern neural networks, there are primarily two steps involved: forward propagation and back-propagation. During **Forward Propagation**, inputs are fed into the network, propagating through all its layers to produce the output $\hat{y}$. This output is then utilized to compute a **Loss function** J, which measures the discrepancy between the network's output and the desired output. The goal of learning is to find the

global minimum of the loss function.

To minimize the loss function, it's essential to calculate the gradient of the loss function with respect to the network's tunable parameters. This is where **Back-Propagation** comes into play, by flowing information back through the network, from the output layer to the input layer, effectively updating the parameters in a direction that reduces the loss.

The mathematical foundation of back-propagation can be summarized as follows:

1. For the output layer, calculate the error term:

$$\delta^{[L]} = \frac{\partial J}{\partial z^{[L]}} = \frac{\partial J}{\partial a^{[L]}} \odot g'^{[L]}(z^{[L]})$$

2. For each hidden layer, propagate the error backward:

$$\delta^{[l]} = \left((W^{[l+1]})^T \delta^{[l+1]}\right) \odot g'^{[l]}(z^{[l]})$$

3. Compute the gradients for weights and biases as:

$$\frac{\partial J}{\partial W^{[l]}} = \delta^{[l]}(a^{[l-1]})^T$$

$$\frac{\partial J}{\partial b^{[l]}} = \delta^{[l]}$$

where $\odot$ denotes the element-wise multiplication, and $g'^{[l]}$ is the derivative of the activation function used in layer l.

These gradients are then applied to update the network's parameters using gradient descent or any of its variants:

$$W^{[l]} = W^{[l]} - \alpha \frac{\partial J}{\partial W^{[l]}}$$

$$b^{[l]} = b^{[l]} - \alpha \frac{\partial J}{\partial b^{[l]}}$$

Here, α represents the learning rate. This iterative process of forward propagation, loss computation, and back-propagation is repeated until the network's performance is satisfactory or a predetermined number of iterations has been reached.

1.1.5. RNN and LSTM

One limitation of feed forward networks is the implicit assumption of Independence between the input's elements. To optimally handle data were this assumption doesn't hold true, for example sequential or temporal data, a new class of models were developed, named **Recurrent Neural Networks**. These networks are termed 'Recurrent' because they execute the same task for each element of a sequence. By constructing an internal 'memory', or hidden state, each element of the input sequence is influenced by those that preceded it. One key concept in RNNs is weight sharing across time steps. In a traditional feedforward network, each layer has its own set of weights, and these weights are used only once during the forward pass for a given input. However, in an RNN, the same set of weights is reused for each element of the input sequence. This is because the RNN processes the sequence step by step, and at each step, it applies the same transformation (using the same weights) to the current input and the hidden state from the previous step. Weight sharing in RNNs has several important implications:

1. Reduced Model Complexity: By reusing the same weights for each step of the input sequence, the number of parameters in an RNN is significantly reduced compared to a similar model that would use a different set of weights for each step. This makes RNNs more memory efficient and faster to train.

2. The ability to Process Variable-Length Sequences: The reuse of weights allows RNNs to process sequences of variable length using the same model. This is crucial for tasks like language modeling or speech recognition, where the input data can vary significantly in size.

3. Capturing Temporal Dependencies: The shared weights, combined with the hidden state that is passed from one step to the next, allow RNNs to learn and remember information about previous elements in the sequence. This capability enables the network to make predictions based on the context provided by the sequence history, capturing short-term and, to some extent, long-term dependencies.

Another characteristic of RNNs is their ability to generalize to any input and output sequence length. This is critically important, as many tasks (e.g. sentences in natural language or genes represented as DNA sequences), vary substantially in length.

One problem of early RNN architectures was the issue of capturing long term dependencies. In 1997 Sepp Hochreiter and Jurgen Schmidhuber proposed the Long Short-Term Memory (LSTM) Network [27]. These network, with the addition of the Gated Recurrent Unit (GRU)[17], would be the standard deep learning network for sequential data, from their development until 2017.

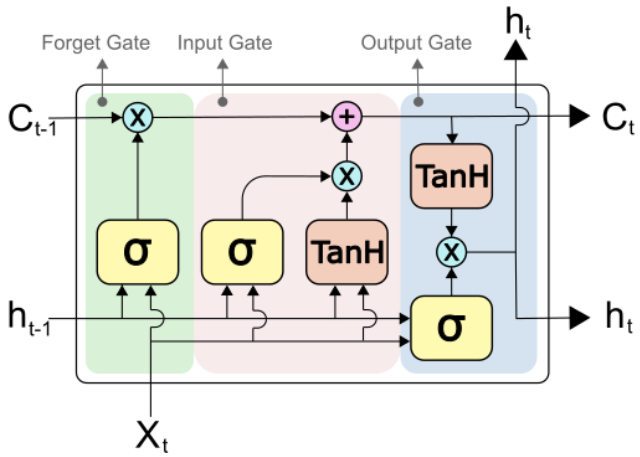


Figure 1.5: LSTM cell

LSTMs are equipped with two distinct types of memory. The first is the memory cell, which serves as the network's long-term storage, preserving information across many time steps. The second is the hidden state, which, in combination with the current input, dictates the memory's dynamics: it determines which parts of the memory should be erased, what new information needs to be stored, and which portions of the memory are relevant for computing the output at the current step as well as updating the hidden state for the next input. This intricate mechanism allows LSTMs to effectively capture and leverage temporal dependencies in sequence data, addressing the limitations of simpler recurrent neural networks (RNNs) in handling long-term dependencies.

However, a significant drawback of LSTMs, as with basic RNNs, lies in their computational model, which inherently lacks parallelism. Due to the sequential dependency of their operations—where the computation of the hidden state for a given time step must wait for the completion of the previous step—parallel processing of input elements is not feasible. This sequential processing requirement imposes limitations on the speed and efficiency of training and inference, especially for long sequences, making LSTMs less suited for applications where parallel computation is critical.

1.1.6. CNN

In 1962 Hubel and Wiesel published a major breakthrough in our understanding of the mammalian visual cortex and its neurophysiology. They proposed the existence of a hierarchical structure composed of two types of cells, **Simple cells** and **Complex cells**[30]. These discoveries would inspire the development of the Neocognitron, proposed by Kunihiko Fukushima in 1979[23]. These biologically inspired networks would reach maturity with the development, between 1989 and 1995, of the LeNet network by Yann LeCun[34], and would become the foundational model architecture of computer vision. These networks became known as **Convolutional Neural Networks**(CNN), the name derives from the convolution operation, which is used instead of matrix multiplication for the basic building block of the network. A full convolutional layer is usually composed by three components, the convolution operator, a pooling operator, and a non-linear activation function.

A convolution is an operator between to function f and g which produce a third function

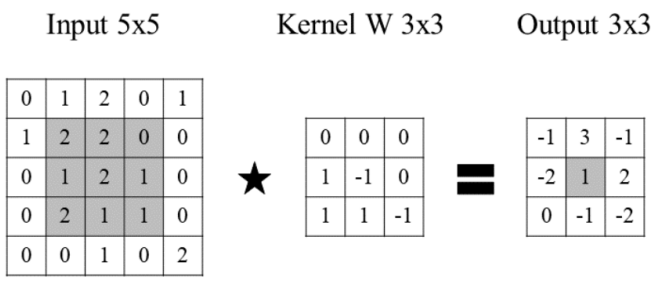


Figure 1.6: Numerical Cross-correlation[14]

$(f*g)$. It is defined as the integral of the product of the two functions after one is reflected about the y-axis and shifted.

$$s(t) = \int f(\tau)g(t-\tau)d\tau$$

if t only contains integer value a discrete convolution can be defined as:

$$s(t) = (f*g)(t) = \sum_{a=-\infty}^{\infty} f(a)g(t-a)$$

In the case of 2-dimensional function the equation becomes:

$$S(x,y) = (g*f)(x,y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(m,n)g(x-m,y-n)$$

In the context of CNNs f is the input, while g is called a **Kernel**. The output is called a feature map.

It is worth noting that the operation performed by CNNs is actually the cross-correlation function, where the kernel function g is not inverted.

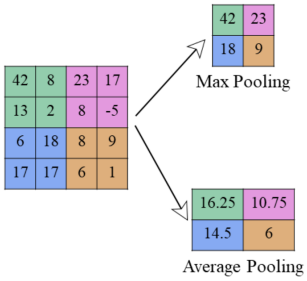
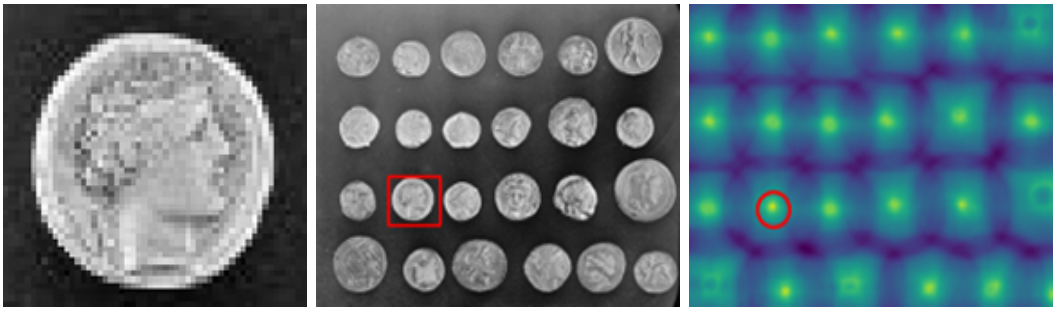


Figure 1.7: Max and Average pooling[14]

The pooling component is used to reduce the feature map spatial dimension, an aggregation function, usually either the max or average, is computed on non overlapping windows.



(a) template (b) image (c) feature map

Figure 1.8: Template matching[51]

The cross-correlation operation can be thought as comparing the kernel filter with each image patch, a high output value means high similarity between the image patch and the kernel. This approach can be used for simple image detection tasks, were the object to identify is used as the kernel. This approach is known as Template matching or Linear filtering.

Similar to the weights in MLPs and RNNs/LSTMs, the convolutional kernels in CNNs are learnt through back-propagation.

1.2. Learning Paradigms

Supervised Learning

Supervised learning is a category of machine learning where a model is trained to perform a specific task using labeled data. Labeled data is composed of pairs of input and output. Mathematically, we can represent labeled data, $\mathcal{D}$, as:

$$\mathcal{D}_{\rangle} = (x_i, y_i)$$

where x_i is the input and y_i is the corresponding output. The model is then expected to learn the relationship between these input-output pairs.

Once a model has been trained on the labeled data, it can be used to perform its task on unlabeled data. The accuracy on the unlabeled data will depend on how well the data it was trained on was representative of real-world cases and on the quality of the labels.

Supervised machine learning is generally divided into two categories:

Regression: In regression tasks, the output is a continuous variable. Mathematically, this can be represented as:

$$y \in \mathbb{R}$$

where y is the real-valued predicted output.

Classification: In classification tasks, the output is a discrete label from a pre-determined number of classes. Mathematically, this can be represented as:

$$y \in \{1, 2, ..., C\}$$

where C is the number of classes.

Unsupervised Learning

Unsupervised learning is another category of machine learning where a model is trained on unlabeled data without explicit supervision. Unlike supervised learning, there are no predefined output labels associated with the input data. The primary goal of unsupervised learning is to explore the underlying structure or patterns within the data. This can involve tasks such as clustering, dimensionality reduction, and density estimation.

One common task in unsupervised learning is dimensionality reduction, exemplified by Principal Component Analysis (PCA). PCA aims to reduce the number of features or dimensions in the data whilst preserving/maximising the explained variance. Mathematically, PCA finds a set of orthogonal vectors, called principal components, that maximize the variance of the projected data. Given a dataset X with N samples and D features, PCA computes the principal components $W = \{w_1, w_2, ..., w_k\}$ as follows: First, the data are standardised:

$$X_{\text{std}} = \frac{X - \mu}{\sigma}, \tag{1.3}$$

Where μ is the mean and σ is the standard deviation. Next we compute the covariance matrix:

$$\Sigma = \frac{1}{N} X_{\text{std}}^T X_{\text{std}} \tag{1.4}$$

We then perform an eigen-decomposition of Σ:

$$\Sigma = Q \Lambda Q^T, \tag{1.5}$$

Where Q is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. Finally, to reduce the data dimensionality, we select the top k eigenvectors:

$$W = \{w_1, w_2, \ldots, w_k\}, \tag{1.6}$$

Where w_i corresponds to the i-th largest eigenvalue. The principal components W can then be used to project the original data onto a lower-dimensional subspace, effectively reducing its dimensionality. PCA is widely used for exploratory data analysis, feature engineering, and data compression. It helps identify the most significant features in the data, uncover underlying patterns, and visualize high-dimensional datasets in a lower-dimensional space. Many modern machine learning approaches to unsupervised learning, particularly when thinking about either density estimation or dimensionality reduction, build upon methods like PCA, such as sparse factor analysis [33], Autoencoders [48] and their variational equivalents [32].

1.3. Modern-day Deep learning

The surge in computational capabilities post-2010 propelled the rapid expansion of the deep learning domain. However, within a short span, the model frameworks established over the preceding four decades began to plateau in performance despite substantial boosts

in data accessibility. Currently, Transformer architectures, introduced in 2017 through the seminal paper "Attention is All You Need" [57] have emerged as the predominant choice for numerous cutting-edge applications. A key factor driving this adoption is the exceptional scalability of transformers, evidenced by their ability to maintain accuracy even with models comprising hundreds of billions of parameters, which continue to benefit from increasing data availability.

1.3.1. Attention

"Attention"[6] is the main building block of the Transformer. Attention was developed as a differentiable retrieval algorithm. Starting from a input sequence, the elements which match with a query term are aggregated as the output. Figure 1.9 illustrates the general

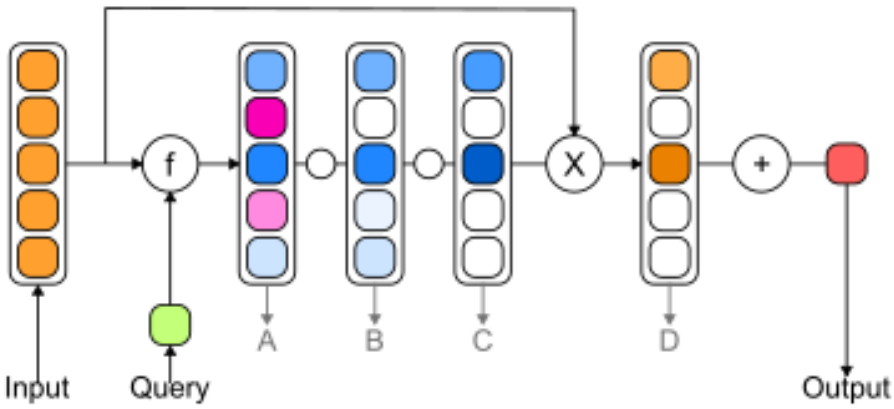


Figure 1.9: Attention general model

principles of Attention. Starting from a Input sequence and a Query term, a similarity score vector is computed by the function f. The vector (A) represents how much each element of the input sequence matches the query. These score are transformed to a distribution, with all elements of (B) have a value between 0 and 1, and the sum of the elements in (B) equalling 1. (B) is refined to form (C), which represents the **Attention weights**. The output is the weighted average of the input using the Attention Weights. In this example one query element is used, however any number of queries can be used, which are output at the same length.

Different implementations of Attention exist, one of the most common, which is used by

the transformer, is called **scaled dot-product attention** and is defined as:

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

Q is the query sequence, K and V are the learnable **Key** and **Value** weights. Q $n \times k$ matrix, K, V are $m \times k$ matrix, d_k is the dimension of the input vectors QKV. The K and V matrices are usually computed by transforming an input sequence with a linear layer. When the Q matrix is also computed using the input sequence, the attention module is referred as a **Self-Attention block**, otherwise is referred as **Cross-Attention**.

One benefit of attention is the increased interpretability of the models which employ

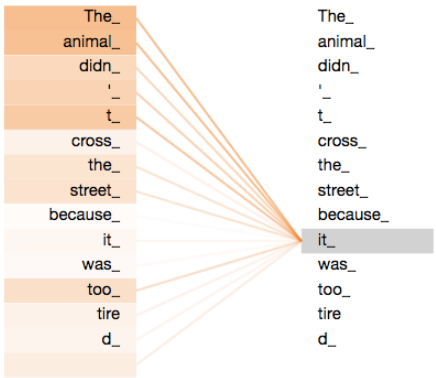


Figure 1.10: Visualization of attention weights[2]

it; by inspecting the Attention Weights of some specific inputs, it is possible to get a qualitative understanding of the information the model chooses to use for it's predictions.

1.3.2. Transformers

Attention was initially used in conjunction with existing architectures, with sequence models like LSTMs especially benefiting from it, as it improved their ability to capture long distance context. The introduction of the Transformer in 2017 showed that a model using only attention could achieve state-of-the art performance, even without the inductive biases used in models such as CNNs. Figure 1.11 shows the original Transformer architecture. It is composed of an Encoder-Decoder structure. The encoder, visible on the left, is responsible for capturing the relevant information and inner relationship of the input sequence. The encoder has two inner layers, both with a skip connection, one is a self-attention layer, and the other is MLP. The encoder is composed by a stack of these blocks, only the last block's output is fed to the decoder. The decoder shares a similar structure but each block has additional cross-attention layers, which uses the encoders

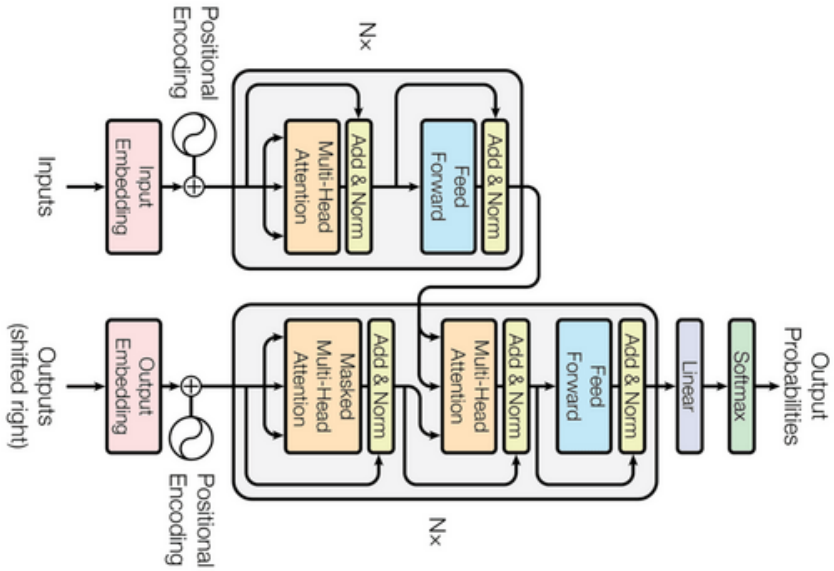


Figure 1.11: Transformer architecture[57]

output. The task of the decoder is to predict the next element of a sequence, given the previous ones. During training, this task is parallelized. This is achieved by the use of a causal attention mask, which allow the elements of the output to get information only of the ones who precede it. Whilst the original Transformer implementation employed an encoder-decoder structure, both modules can be used as stand-alone models.

Currently the biggest challenge for transformer models is that the self-attention mechanism has memory requirements that are quadratic with respect to the input sequence length. Several recent innovations have tried to ameliorate this issue[60][61][63].

One very interesting property of transformers is that, at least for NLP tasks, their performance scales approximately linearly with their parameter count and the amount of training data. This has led to the development of some huge models, hence the coined phrase, "Large Language Models" (LLMs). LLMs to date have trainable parameters in the range of hundreds of billions, and are trained on petabytes of data.

1.3.3. Vision Transformers (ViTs)

Given the overwhelming success Transformers had on NLP tasks, a lot of effort has been put to replicate their success on image data. The 2020 paper "An Image is worth 16x16 words" introduced the Vision transformer (VIT) and showed that a transformer model could outperform conventional CNNs for classification tasks[21]. VITs are based on a transformer encoder, as such they expect a 1d sequence as input. To transform an image

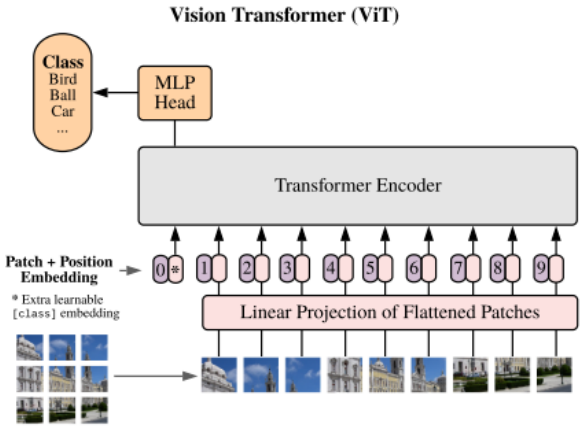


Figure 1.12: Vision transformer[21]

into a suitable shape for the encoder, it is split into non overlapping patches, each patch is flattened and transformed into a k-dimensional **Patch Embedding** with a linear layer. Like a normal transformer a positional encoding is added to the patch embeddings to keep the spatial relationship between patches. An additional learnable token is appended at the start of the embedding sequence, called a class token or CLS token, it's used to capture the global information of the image. Only the CLS token output is used to compute the loss, the other patch embeddings are typically ignored.

Overall the improvements of ViTs over CNNs were limited compared to the relative improvements Transformers had over LSTMs. ViTs are slightly more computational efficient than CNNs and can achieve higher performance, however this last property can only be appreciated once trained on enough data, with CNNs still being competitive in lower-data regimes (millions of examples).

1.3.4. Self-supervised Learning paradigms

The underlying limitation of training purely supervised models is the availability or curation of labelled datasets. Such labelled datasets (e.g. ImageNet), whilst large, are a fraction of a percent of imaging data that exists online. To overcome this bottleneck, unsupervised and self-supervised learning (SSL) have emerged as possible state of the art approaches.

SSL aim to pre-train a model using large amount of unlabeled data to discover hidden patterns, which can form provide the model a 'common sense' intuition. The learned representation can be then refined to perform more specific task, by using a small amount of annotated data.

Whilst SSL use no labels, it has been gaining an independent identity from Unsupervised Learning. One reason for such shift is that even if they don't use human curated labels, they use plenty of supervisory signal in the form of **pseudo-labels**, which are generated from the data itself trough what is called a **Pretext** task.

To understand how a pretext task can learn without labels, a popular SSL framework, the **Autoencoder**(**AE**) can be taken as an example.
AE have an encoder-decoder structure, the encoder compress the input to a latent representation. This vector is used by the decoder to reconstruct the original input. The loss is computed by comparing how well the reconstruction matches the original data.

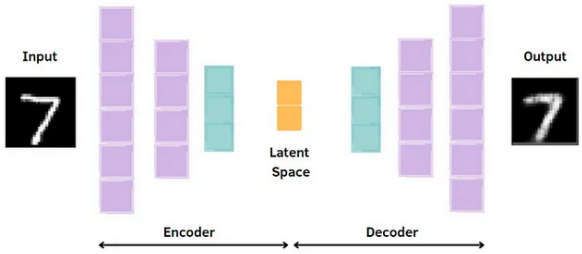


Figure 1.13: Vanilla autoencoder architecture[4]

Overall, current SSL methods can be split in two families: **Generative** architectures and **Joint-Embedding** Architectures.

Generative methods usually involve reconstructing the input starting from a corrupted version of it. Autoencoders falls into this category.

Joint-Embedding architectures are based on two encoder networks, both encoders produce a latent representation, or embedding, of the respective input, which are then compared with some loss function.
Early applications used two unrelated images as the input of the two encoders and tried to maximize the difference between the embeddings. This is often referred as **Contrastive Learning**. more recently, a lot of work has focused on developing non-contrastive approaches, where the two encoders receive two views of the same image, and the loss function tries to minimize the difference between the two embeddings.

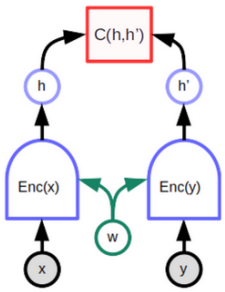
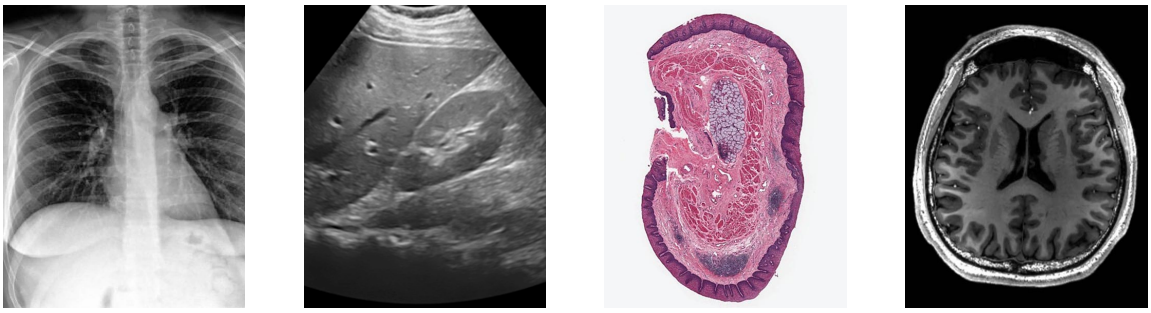


Figure 1.14: Joint-embedding architecture[35]



(a) Xray (b) Ultrasound (c) WSI (d) MRI

Figure 1.15: Biomedical images[65][18]

1.4. Applications of Machine Learning to biomedical imaging

One field of major interest for the application of Machine learning is Biomedical Imaging. Figure 1.15 show some common acquisition modalities. These techniques play a crucial role for the healthcare field, they can be used to diagnose disease, monitor treatment success as well as disease progression.

Deep learning has shown great promise in aiding and automating our understanding of complex, high resolution biomedical images. For example, histology is used routinely to diagnose patients that have undergone biopsy or surgery. Glass slides containing small amounts of thinly sliced (4um) tissue are stained and analysed under a microscope for diagnostic purposes. However, recent technological advances is taking pathology digital. Diagnostic glass slides can now be scanned into gigapixel, high resolution 40x images, called Whole Slide Images (see Figure 1.15c). Deep learning algorithms have been successfully applied to WSI to perform cell identification [56], cell instance segmentation [29], tissue classification [16] and even the prediction of molecular readouts, such as gene expression [18], microsatellite instability (MSI) [13] and breast cancer HER2 positivity

[59].

Other imaging modalities, such as full-body MRI, have been utilised alongside deep learning algorithms, to quantify organ and fat volume variability within populations [37] as well as the characterisation of subregional structural variation of the human brain [22]. One primary concern when applying machine learning algorithms to medical images are their predictive interpretability. Recent advances, such as vision transformers (discussed previously), have built in attention mechanisms that guide the user in how any given prediction was made. However, there is a large field of explainable AI that is attempting to make machine learning models applied to such data, as interpreitable and as transparent as possible. A substantial problem of utilising supervised learning in the medical imaging domain is the need for complex, time-consuming and expert-driven annotations. For example, annotating 3D organ structures requires hundreds of 2D-slice annotations requires hours of consultant/specialist time. An alternative and increasingly promising approach is the application of unsupervised or self-supervised representation learning algorithms to biomedical datasets ([15], [54], [62]). These methods offer two possible advantages:

1. They require little to no annotation
2. They have the ability to extract unknown or unused diagnostic information from images that would otherwise be ignored

However, whilst such approaches are now competitive with supervised methods when applied to large natural image datasets, medical images are typically more scarce, requiring access from larger integrated healthcare systems or national biobanking initiatives (see section: UK Biobank).

1.4.1. Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a non-invasive imaging technique used to capture internal images of the body. MRI is a pivotal diagnostic tool within the realm of medical physics, renowned for its versatility and depth of detail in imaging soft tissues. Its applications span a broad spectrum of medical fields, each leveraging the technique's capacity to produce high-resolution images without the use of ionising radiation, a significant advantage over traditional X-rays and CT scans.

How does it work: MR uses the electromagnetic properties of the hydrogen atom's nucleus to acquire images. To understand how MR acquires images, an understanding of sub-atomic particle physics is necessary.

One key property is the particle **Spin** (S), which denotes its intrinsic angular momentum. The name Spin derives from the particles' behaviour under the influence of a static magnetic field, which is analogous to a rotating object under the influence of gravitational force. However, it must be noted that the spin (S) is a quantum mechanical property and not a literal spinning motion, which affects the particle's magnetic moment (μ).

Particles with a non-zero spin exhibit a **Magnetic Moment** (μ). Spin and Magnetic Moment are connected by a constant γ called the gyromagnetic ratio with the formula:

$$\mu = \gamma S$$

The value of the gyromagnetic ratio (γ) is unique to each atom, and in MR fields it is commonly measured as [MHz/Tesla].

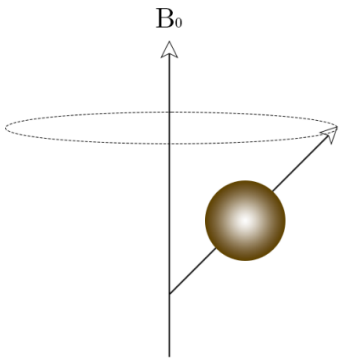


Figure 1.16: Precession of a particle along a magnetic field B_0[14]

When a particle with non-zero Spin is placed in a magnetic field, it will **Precess** around it. When placed in a magnetic field, particles with a non-zero spin precess around the magnetic field (B_0) at a frequency (f_0) determined by one of the most important equations in MR, the **Larmor equation**, given by:

$$f_0 = \gamma B_0$$

During an MRI acquisition the signal from individual nuclei is not relevant, instead the magnetic properties of billions of nuclei are aggregated to form a **Net Magnetisation** At equilibrium the net magnetisation will align to the magnetic field, with no precession, as the individual nuclei contribution cancel each other out.

The main main field (B_0), produced by an MRI machine, is in the order of tesla, while the

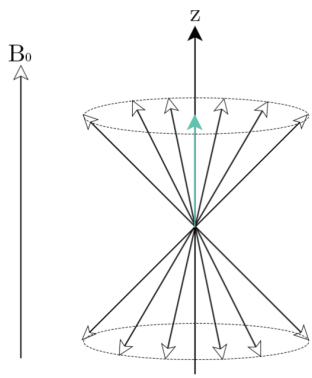


Figure 1.17: Average of many nuclei produce the net magnetisation[14]

net magnetisation of a body is in the microtesla range. The magnitude difference makes it unfeasible to measure the net magnetisation along this axis. MRI machines apply a second magnetic field, called the Radio Frequency (RF) pulse (B_1), which acts perpendicularly to the main magnetic field B_0. This second field will induce the net magnetisation to precess along the z-axis with an ever increasing angle as long as it is active.
The B_1 field is not static but does rotate with a frequency extremely close to the hydrogen proton's Larmor frequency. This will induce a resonance behaviour, which will cause the net magnetizstion to rotate toward the transverse axis. The RF field B_1 rotates at or near the Larmor frequency of the hydrogen protons, inducing resonance. This synchrony maximises the energy absorption by the protons, effectively tipping the net magnetisation towards the transverse plane. If the B_1 field were to rotate at a significantly different frequency, it would not effectively synchronise with the spins, leading to a negligible net magnetisation in the transverse plane.

The MR signal is formed by the precession of net magnetisation in the transverse plane, induced by the moving magnetic field (B_1). This signal is detected by coils aligned perpendicularly to the main magnetic field (B_0) following Lenz's law. Once the RF field is turned off, the system will decay to the equilibrium state. The time it takes the system to return to equilibrium is known as the relaxation time. However, this decay is not uniform, but different tissues will exhibit different decay patterns. This makes it possible to distinguish among different types of tissue.

1.4.2. DIXON MRI

Among various types of MRI, the DIXON imaging technique is the one that allows a clear distinction between the water and fat present in a particular tissue. This technique has been developed by extending the chemical shift imaging method using the in-phase/out-

of-phase cycling of fat and water proposed by Dixon in 1984 [20]. He demonstrated that the signal intensity of pixels from the in-phase image (IP) and the signal intensity from the out-of-phase or opposed-phase image (OP) are related to water (W) and fat (F) as:

$$\begin{aligned} IP &= W + F \\ OP &= W - F \end{aligned} \tag{1.7}$$

By calculating the average of the complex sum and the complex difference of these images, Water-only and fat-only images can thus be derived using the following equations:

$$\begin{aligned} 1/2[IP + OP] &= 1/2[(W + F) + (W - F)] = 1/2[2W] = W \\ 1/2[IP - OP] &= 1/2[(W + F) - (W - F)] = 1/2[2F] = F \end{aligned} \tag{1.8}$$

1.5. Complex Traits Genetics

A disease, or Trait, is defined as complex when it is the result of intricate interplay of of genetic and environmental factors [41]. In the realm of genetics and biomedical research, the study of disease inheritance patterns has long been a focal point. One of the fundamental distinctions lies between monogenic diseases, caused by mutations in a single gene, and complex diseases or traits, which result from the intricate interplay of multiple genetic and environmental factors [7]. This demarcation forms the basis of understanding the genetic architecture of various disorders and conditions afflicting human populations. While monogenic diseases often exhibit straightforward inheritance patterns, complex traits such as diabetes, cardiovascular diseases, and psychiatric disorders present a daunting challenge due to their multifaceted nature. In recent years, advancements in genomic technologies and computational methodologies have enabled researchers to delve deeper into the genetic underpinnings of complex diseases, shedding light on the intricate mechanisms governing their manifestation [58]. This has primarily been guided by large biobanking initiatives, such as the UK Biobank, in which volunteers are both densely phenotyped and genotyped [8]. These biobanking initiatives have enabled researchers to conduct large scale genetic studies, known as genome-wide association studies (GWAS). In GWAS, researchers assess the association between specific genetic variants and a given trait or disease of interest. Performed genome-wide across millions of typed or imputed variants, a GWAS provides a list of genetic variants associated with the trait. Over the last 15 years, thousands of genetic variants across hundreds of diseases and traits have been described [1]. For complex diseases and traits, hundreds of small effect genetic variants can contribute to the traits variability in a population. These are known as polygenic traits.

Polygenicity and the role of genetic variation in influencing traits is highly variable, and often a quantity that is estimated. The proportion of variation explained by genetics is referred to as narrow-sense heritability:

$$h^2 = \frac{V_g}{V_g + V_e}$$

where V_g is the proportion of variance attributed to the additive genetic component and V_e is the proportion of variance attributed to the environmental component (ignoring an gene-by environment interaction terms). A complex trait can therefore be understood as a ratio of both genetic and environmental effects.

1.6. The UK Biobank

The UK Biobank is a large-scale biomedical database and research resource, collecting data from participants aged between 40-69 years from across the UK. UK Biobank contains in-depth genetic and health information from half a million UK participants. The Biobank was established to improve the prevention, diagnosis, and treatment of a wide range of serious and life-threatening illnesses – including cancer, heart diseases, stroke, diabetes, arthritis, osteoporosis, eye disorders, depression, and forms of dementia.

The data from the UK Biobank has been utilized extensively in the understanding of the epidemiology of diseases and the genetic basis of complex diseases and traits through genome-wide association studies (GWAS). These studies have identified numerous genetic variants associated with diseases and traits, offering insights into their biology and potential therapeutic drug targets.

- **Disease Understanding**: The broad range of data allows researchers to study the multifactorial causes of diseases, including genetic predispositions, environmental exposures, and lifestyle factors.
- **Genetic Basis of Complex Diseases and Traits**: GWAS utilizing UK Biobank data have uncovered genetic variants linked to a plethora of conditions and traits, enhancing our understanding of their heritability and biology.
- **Intermediate Traits**: The imaging data from the cohort enables the study of intermediate traits, such as organ size, liver fat content, and brain structure variations. These traits can help in understanding the pathophysiology of complex diseases by acting as biomarkers or revealing early disease processes.

The imaging cohort within the UK Biobank aims to collect imaging data from 100,000

of its participants. This effort includes a variety of imaging types, such as MRI scans of the brain, heart, and body, as well as DXA (Dual-energy X-ray Absorptiometry) scans for assessing bone density and body composition, and ultrasound scans of the carotid arteries. The objective is to provide a comprehensive imaging dataset that can be used in conjunction with genetic, lifestyle, and health record data to study the underlying causes of many different diseases in detail.

1.6.1. The UKBB DIXON Full Body MRI

The DIXON MRI technique in the UK Biobank imaging study is able to distinctly separate and visualize fat and water tissue content within the body. This technique is based on the chemical shift property of hydrogen atoms in fat and water, which allows for the acquisition of images that can be processed to produce separate signals for fat and water. This separation is critical for accurately assessing body composition, including visceral and subcutaneous fat volumes, muscle mass, and even detecting ectopic fat deposits that are associated with various metabolic and cardiovascular diseases.

The DIXON full body MRI data from the UK Biobank imaging cohort has numerous applications, including:

- **Disease Mechanisms**: Understanding how body fat distribution and composition are related to the risk of diseases such as diabetes, cardiovascular disease, and certain cancers.
- **Intermediate Traits**: Full body MRI enables the study of intermediate traits like liver fat content, muscle mass, and abdominal fat distribution. These traits can act as markers for understanding the progression of metabolic diseases and the effectiveness of specific interventions.
- **Phenotypic Associations**: Researchers can use the data to explore associations between genetic variants and phenotypic traits derived from the imaging data, as previously described using GWAS.

While the DIXON full body MRI provides invaluable data, its implementation on such a large scale also presents challenges, including the standardization of imaging protocols across different scanning sites, ensuring the high quality of the data, and developing sophisticated image analysis tools capable of handling the complexity and volume of the data generated. This dataset will be the primary focus of my thesis.

1.7. Related Work

The study of Liu *et al* [38] has been of the utmost relevance for this work.

They analyzed over 38000 subject from the UK Biobank across multiple imaging datasets. By using a supervised deep learning model, they were able to quantify 11 traits related to overall volume, fat, and iron content in multiple organs and tissues. From these traits, also termed Image Derived Phenotypes (IDPs), they were able to identify multiple health-related outcomes and by performing a Genome Wide Association Study (GWAS) they found 4 novel associations with liver volume out of a total of 93 genome wide significant associations.

Of these 11 IDPs, 6 are extracted from the DIXON dataset: volume of the liver, kidneys, spleen, lungs, Visceral adipose tissue, and abdominal subcutaneous adipose tissue, which will be discussed later in the results section.

2 Methods

2.1. Data

2.1.1. UKBiobank

The dataset used in this thesis was approved by the UK Biobank Research Ethics Committee (reference number 11/NW/0382) and the use of the UK Biobank as part of the application 82779. It was approved by a local institutional review board.

2.1.2. Clinical Data

Hospitalization events

UKBB's category 2002 is the source of the hospitalization diagnoses. This record use the ICD9 and ICD10 clinical ontologies.

The records don't have a regular structure and it was neccessary to refactor them into a dataframe with a defined schema (*key, date, id, icd_code, icd_version, diagnoses_type)*. The *key* is the unique entry identifier, *icd_version* specifies if the code is from ICD9 or ICD10, *diagnoses_type* specifies if the entry is the reason for the hospitalization, or it's an underlying / secondary condition.

GP clinical event

The UKBB Record Table 1060 is the source for the primary healthcare information. This record uses the READv2 and READv3 clinical ontologies.

These codes provide an extremely detailed encoding, not only of clinical information of diagnoses, laboratory results, symptoms, etc. But also ethnicity, occupation and religion. The table contains more than $1e^8$ entries, making it quite burdensome to extract meaningful information without large amounts of manual curation.

The READ codes were converted to ICD-10 codes using the UKBB Resource 592, which

provides mappings between different ontologies.

READv2 were converted only if they mapped to a unique code.

READv3 were converted if the corresponding ICD code was a one-to-one match (*mapping_status* = E,G), or if a default mapping existed among multiple alternatives (*mapping_status* = D).

All READ codes which were not converted were filtered out.

GP medications

The UKBB's Record Table 1062 is the source for the primary care prescription data. This table uses three different ontologies: READ_2, BNF, and DMD.

Read_2 and BNF cover more than 99% of the table entries. Since there is no mapping between DMD and BNF, DMD entries were removed.

BNF provides the most practical encoding, with a clear hierarchical structure. UKBB Resource 592 provides the mapping between READv2 and BNF. After the conversion, any entry without a BNF code was removed.

Verbal interview

UKBB data-Field 20002(20003) contains the self reported non-cancer illness (medications) obtained during the verbal interview at the UKBB assessment centre visits.

The records don't have a regular structure, instead each subject has an associated list containing all the illness (medication) codes applicable to them. This data was refactored in a dataframe with the following schema: *(key, id, code, date)*. *key* is a unique entry key, *id* is a subject identification code, *code* is the illness (medication) code and *date* refers to the first recorded occurrence.

2.1.3. Whole body MRI

MRI Acquisition

As described in [36] all scans are performed using a Siemens 1.5 Tesla MAGNETOM Aera scanner (Siemens Healthineers, Erlangen, Germany) with VD13A software and employing relevant elements of spine and body matrix coils. No pharmacological stress or contrast agent was used. The scans were acquired with 6 stages and localisation was performed relative to the jugular notch, which is the centre position of the first stage of the Dixon

imaging.

Preprocessing

An automated pipeline from [38] was used to preprocess all the raw imaging data from DICOM format. The pipeline merged the six acquired stages into a unique three dimensional, "neck-to-knee" full body volume and apply a bias-field correction. The volumes are resampled to a single dimension and resolution (size = [224, 174, 370], each voxel corresponding to $2.232 \times 2.232 \times 3.0$ mm). Three additional channels are computed by the pipeline, the "Water Percent", "Fat Percent", and a mask. These correspond to overall water percentage, fat percentage, and a binary mask that separates background from body. These processed data represent approximately 55TB of stored MRI volumes.

2.1.4. 2D training

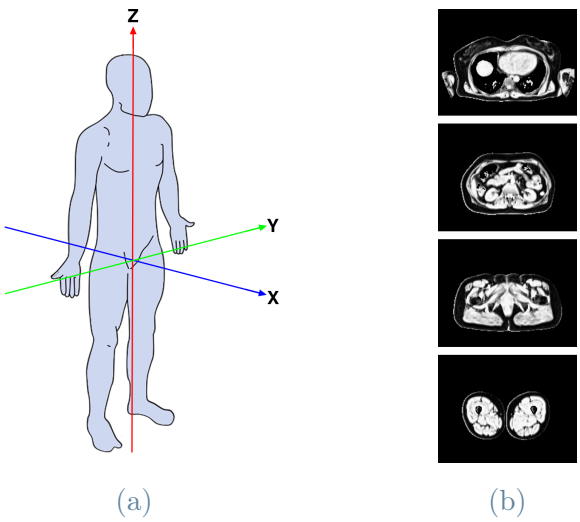


Figure 2.1: (a)**Anatomical body axis**: X - Saggittal axes, Y - Frontal axis, Z - Longitudinal axis. (b) Example's longitudinal slices

The large size of the 3D volumes makes training models extremely computationally demanding. To achieve this given limitations in our HPC environment, a specific training regiment was implemented. Each 3D volume was subdivided along its longitudinal axis to obtain 2D slices (Figure 2.1). These slices are uniformly sampled to form a batch, regardless of the volume[1].

The model trained with this procedure computes a latent factor for each slice. After inference, all of the latent factors from the overall volume are aggregated in a single sequence.

[1]This uniform sampling avoids any memory bottle-necking issues

During the model's training, the slices are treated independently. In this case the full dataset dimension is: $52773 \times 370 = 1.776e7$

2.2. Image embeddings

2.2.1. Factor VAE

Variational Autoencoders (VAEs) are models in the Autoencoder family. One limitation of the standard Autoencoder is their latent space structure, which can be described as discontinuous or none-regularized. The learnt representation only uses a limited portion of the latent space, selecting a point outside these areas will generate a random output. As standard autoencoders are not generative, this limits the interpretability/downstream application of the learnt representation, outside of performing compression.

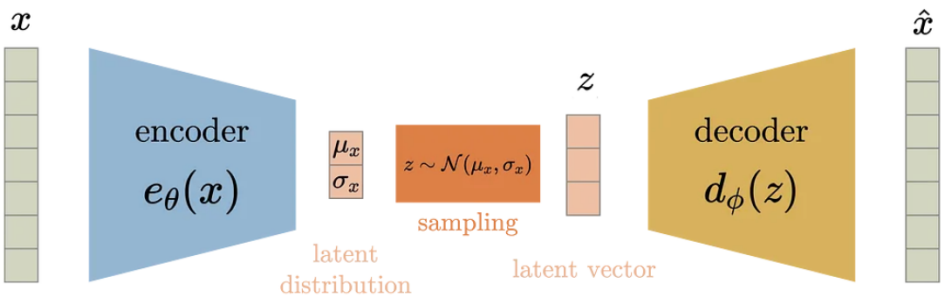


Figure 2.2: Variational Autoencoder architecture. source[5]

VAEs are able overcome this issue by replacing the deterministic latent factor with a probabilistic one.
The networks encoder produces two vectors μ_z and σ_z, that are used as parameters of a standard normal distribution, respectively as the mean and standard deviation. A vector z is sampled from this unit normal distribution and decoded to produce the reconstruction.

More formally, VAEs learn a stochastic mapping between the input, described by a complex and unknown data generating distribution $p(x)$, and the latent space z, usually described by a normal distribution $p(z)$. The model learns the joint distribution $p(x, z)$ which can by factorized with the product rule as:

$$p(x,z) = p(x) \underbrace{p_\phi(z|x)}_{\text{Encoder Distribution}} = p(z) \underbrace{p_\theta(x|z)}_{\text{Decoder Distribution}}$$

The encoder aims is to learn the $p_\phi(z|x)$ distribution, the decoder learns $p_\theta(x|z)$. To map from the data generating distribution $p(x)$, to a normally distributed latent factor, and then back to a reconstructed version of the input, both the encoder and the decoder must learn a complex mapping function. Inferring the true joint distribution is an intractable problem, with no direct analytical solution. Instead an approximate variational distribution $q_\phi(z|x)$, can be learnt using techniques from variational inference.

The optimization objective of the VAE is the evidence lower bound(ELBO), which is a lower bound on the data log likelihood $\mathbb{E}_{p(x)}[\log p(x))$:

$$\mathcal{L}_{VAE}(\phi,\theta) = \underbrace{\mathbb{E}_{q_\phi(z|x)}(\log p_\theta(x|z))}_{\text{Reconstruction error}} + \underbrace{D_{KL}(q_\phi(z|x) \parallel p_\theta(z))}_{\text{KL divergence}}$$

where ϕ and θ are the trainable parameters of the encoder and the decoder respectively. D_{KL} represents the Kullback-Leibler divergence that measures the difference between the two probability distributions. Therefore the VAEs loss function, $\mathcal{L}_{VAE(\phi,\theta)}$, is a trade off between reconstruction error and regularisation of the latent factors to be approximately normal.

Factor VAE[31], is a method used to improve the disentanglement of the latent space z without the trade off in reconstruction quality necessary in other VAE models such as the β-VAE[25]. FactorVAE modifies the VAE objective with a term that directly encourages independence in the latent factors, arriving at the following objective:

$$\frac{1}{N}\sum_{i=1}^{N}[\mathbb{E}_{q_\phi(z|x^{(i)})}(\log p_\theta(x^{(i)}|z)) + D_{KL}(q_\phi(z|x^{(i)}) \parallel p_\theta(z))] - \gamma D_{KL}(q(z) \parallel \bar{q}(z))$$

where $\bar{q}(z) := \prod_{j=1}^{d} q(z_j)$; the term $D_{KL}(q(z) \parallel \bar{q}(z))$ is known as *Total correlation*(TC) A discriminator network is trained to classify between samples from $q(z)$ and $q(\bar{z})$, this task allows it to approximate the density ratio needed for estimating TC.

2.2.2. DINO

DINO(DIstillation with NO labels)[11], is a self-supervised learning framework.

It's based on a student-teacher architecture. Each of the two branches have two components: an encoder network, which creates a latent representation of an image, and an MLP head, also called a projection head, which expands the latent vector to K dimensions. A probability vector P is obtained by normalizing the output of the network with a Softmax function.

More precisely, given an input image x and the branch network g, P is computed as:

$$P(x)^{(i)} = \frac{\exp\left(g(x)^{(i)})/\tau\right)}{\sum_{k=1}^{K} \exp\left(g(x)^{(k)})/\tau\right)}$$

τ is a temperature parameter, and controls the sharpness of the output distribution.

DINOs learning procedure consists as follows: Given an image, a set V of different views is obtained with random transformations. This set contains two global views, x_1^g and x_2^g and several local views with a smaller resolution. All crops are passed through the student while only the global views are passed through the teacher. The loss that is minimized is as follows:

$$L = \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} -(P_t(x) log(P_s(x'))$$

where the term $-P_t log(P_s)$ is the Cross-entropy loss.

Only the student branch parameters are updated by stochastic gradient descent. The teacher branch use an exponential moving average(ema) of the student networks parameters, computed as:

$$T_t = S_t * m + (T_{t-1} * (1 - m))$$

the m parameter is called **Teacher momentum**.

The choice of hyperparameters is critical to avoid the latent representation collapsing to a trivial solution, this phenomenon would manifest either with the latent having extreme entropy values (approximating a uniform distribution), or a low entropy solution, where one dimension dominates the others.

Two phenomenon, **centering** and **sharpening** has to be balanced. Sharpening decreases the entropy of the latent while centering increases it.

The centering operation also adds a bias term c to the teachers output before applying the softmax normalization. the center c is updated with a e.m.a. with the formula:

$$c = mc + (1 - m)\frac{1}{B}\sum_{i=1}^{B} g_t(x_i)$$

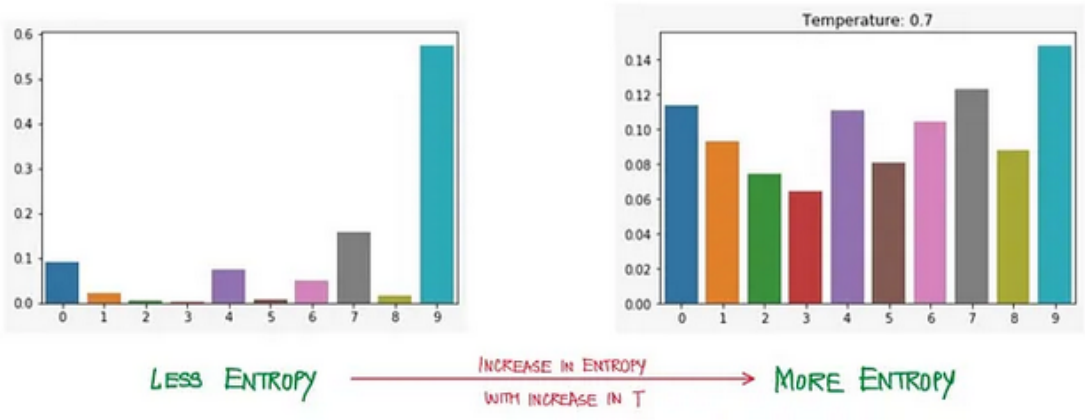


Figure 2.3: Softmax sharpening. Reproduced from the following source[52]

DINO is agnostic on the encoder network choice, however vision transformers have demonstrated state of the art performance.

After the model is trained, the projection heads are discarded, and the teacher network is kept as the final model.

2.2.3. DINOv2

DINOv2 builds upon the original paper by combining it with methodologies from two previous papers, SwAV [10] and iBOT [64]. A major focus of this work was towards stabilizing and accelerating SSL training when scaling model and data sizes.

Notable changes are the incorporation of the loss from iBOT and the centering algorithm suggested by SwAv.

The iBOT loss introduce a patch-level objective. The student input images have a random amount of masked patches. The teacher receives the complete images. The student masked token are retrieved from the VIT output and are processed analogously as the default DINO token. The same process is applied to the teachers (visible) patch tokens which corresponding to the ones masked in the student. The iBOT loss term is obtained as:

$$L_{iBOT} = -\sum_{i} P_{Ti} log P_{Si}$$

where i are patch indices for masked tokens.

The encoder for DINOv2 must be a VIT, this is due to the iBOT loss term which uses the seq2seq characteristics of the model to concurrently obtain the image and patch representations.

Pretraining

The training procedure described in 2.1.4 was used. The input channels were *Water Percent, Fat Percent and In-Phase*. The 2D slice were padded to 224×182 to match the patch size constraint.

The pretrained model used is the DINOv2 VIT_base/14. it has a patch size of 14^2 pixel, and an embedding d of 768. The VIT has 18 layers, each layer has 12 attention heads.

Model Distillation and Fine tuning

Distillation [26] aims at reproducing the output of a large model with a smaller one by minimizing some distance between both outputs for a set of given inputs. Since DINO's loss function is a form of distillation itself, the default training loop can be used for this task with minimal modifications. The larger model to distill is used as the Teacher network, with all it's parameters frozen. The Student's parameters EMA is not used to update the teacher, but is stored to be used as the final version of the model. This technique was used to distill the pre-trained VIT-base into a smaller model. The network used for the student is a VIT with 12 layers, each layer with 8 attention heads. The embedding size d is 256.

After distillation, Fine tuning is performed: the distilled Student's EMA weights are used as a starting point for both the student and the teacher network, and the default DINOv2 training procedure is run.

In both stages, the global crops (local crops) are extracted with a crop factor between 0.4-0.9 (0.15-0.4). The aspect ratio is kept constant and the major length of the crop is resized to 224(98). A warm-up period of 5 epochs effects the *teacher temperature* and *learning rate*, which rise from 0.2->0.4 and $1e^{-6}$->$1e^{-3}$ respectively. The batch size used was 90 per GPU. Other hyperparameters are set to the DINOv2 default values. The input data are the same as the pretrained model.

Custom

The main aim of the custom implementation is to force the model to ignore the background region. This is achieved with two solutions. The first is to avoid crops containing only background, this is achieved by forcing the center point of the crop to on a body region; since the background pixels have a value near zero, selecting all the pixels above a threshold provide a mask on which the crops can be centered. To be used as the center of a crop, a pixel must have at least a value of 0.1(pixel value in the range [0,1]) in one

channel.

This solution avoid selecting crops without any semantic meaning, but a lot of background was still being captured, especially in the global crops.

The second solution modify the attention mechanism to avoid putting any focus on background patches still captured by the crops. Attention mask are commonly used, for example in the transformer Decoder, however the default approach assume that all inputs use the same mask. In our case each element require a different mask, store all these mask would a term with a memory complexity that scales quadratically to the input sequence length. An alternative approach was devised: all patches with a maximum value below 0.1(on a range of value in [0,1]), are considered background patches. The indices of these patches are then passed to the attention mechanism, where they are used to select the QKV tokens corresponding to the backgrounds area, which are than multiplied them by 0. Supplementary figure 5.1 show the attention masking effect.

The main model architecture is the same as the fine-tuned model, VIT with 12 layers, each with 8 attention heads, and an embedding size of 256. The patch size was set to 11.

To improve the computational efficiency the projection head output size was reduced from 131072 to 5000, while is a major reduction, the ablation studies performed in the original DINO paper[11] showed a limited impact on performance. Another computational optimization was to use a two step training. During the first step the model is trained at a lower resolution until the loss converged. During the second step the resolution is increased for a few epochs. DINOv2 used this strategy, as it achieve comparable results to a model trained with high resolution from scratch, at a fraction of the computational cost.

Following the findings of [19], 4 learnable parameters called registration tokens are added to the patch-embeddings sequence like the CLS token. These tokens should reduce artifacts in low-informative regions.

The training procedure described in 2.1.4 was used to load the data, however to reduce the training time the bottom 100 slice were discarded. Instead of 2D slice small 3D subsection of the longitudinal axis were extracted, having a shape of 224×172×3 with 2 channel. These 3D section overlap with each other and are extracted with stride 1.

the keep the 2D architecture, the third dimension was stacked with the channels, producing a 2D image with 6 channels. The image was than padded to be a square aspect ration, with size 224×224.

the global crops(local crops) are extracted with a crop factor between 0.4-0.9(0.2-0.4).

the The aspect ratio is kept constant and the major side of the crop is resized to 132(66) during the low resolution step and to 220(99) during the High(low) resolution step. A different set of augmentation was used(Table 5.1).

A 5 epoch warmup rate was used for the Learning rate, which increase from $1e^{-6}$ to $5e^{-4}$. A fixed teacher temperature of 0.04 is used. Other hyperparameters use DINOv2 default value. The batch size was 420(170) during the low(high) resolution step.

2.3. Encoder evaluation

Global PCA

For the global PCA experiments, embedding sequences from all subjects were converted to a 1D vector, these were obtained by simply concatenating all the subject's slice embeddings. PCA was then utilised to select the first 50 components from each subjects concatenated representation.

Slice PCA

For the slice based PCA experiments, 100 randomly sampled subjects were selected. PCA was used to compute the first 50 components from each slice embedding of the selected subjects.

Patch Embeddings PCA

The patch tokens were selected from the VIT's output. The top 50 PCs of the tokens were computed. The token were rearranged to form an image, where each token represents one pixel, with the PC as the channel dimension. The pixel at i, j therefore has the value of the token generated by the input patch at i, j.

Pseudo-Segmentation

The patch tokens obtained in 2.3 were clustered with hierarchical clustering. The cluster labels were used to form an image, allowing one to see the spatial distribution of the embedding clusters. Meaningful clusters should have structure, for example specific clusters that represent organs, or specific localisation within a given image. Clusters scattered across the entire image will highlight the presence of noise in the patch embeddings.

2.4. embedding analysis

embedding transformer

A Transformer encoder with linear prediction head is used to aggregate the slice embedding and predict global traits. The encoder has 12 layers with 8 attention heads each, the embedding size is 256, its outputs token are aggregated with an Global Average pooling layer, with is processed by a linear layer to produce the final output. By modifying the output dimension the architecture can be used for both single and multi-traits prediction task.

2.5. GWAS

Genome-wide Association studies (GWAS) were performed using REGENIE [40] (version 3.2.7 - nf-pipeline-regenie).

All Autosomal variants were tested for association whilst adjusting for: Gender, age at imaging, age at imaging squared, imaging date (year), genotype batch, imaging centre.

REGENIE performs genome-wide association studies in two steps: A whole genome regression model is fit that first estimates a polygenic risk score. For computational efficiency, variants are split into many different blocks and ridge regressions are performed in parallel. These separate predictors are then used in a second ridge regression step and a shrinkage parameter is estimated based on cross-validation. This polygenic risk score, or phenotype genetic prediction is then estimated using a "Leave One Chromosome Out" methodology. All baseline covariates are first regressed out of both the phenotype of interest and the genotypes. For this estimate, it is important to use variants that are not in LD and therefore, we used 417k SNPs as in [53]. Variants a subjects were retained using the following thresholds:

1. Minor allele frequency (MAF) > 1
2. Genotype call rate $\geq 99\%$
3. Minor allele frequency (MAF) $\geq 1\%$
4. Hardy–Weinberg equilibrium (HWE) $P \geq 10^{-15}$
5. LD pruning performed with PLINK, version 2.0 [12]).
6. Subject missingness $\leq 10\%$

LD pruning was performed using PLINK, version 2.0 [12]).

After this step, all genome-wide imputed variants are tested for association with the trait of interest using linear regression, conditioning on the polygenic predictor estimated in step 1. 42,795,932 SNPs were tested for association in total using the following criteria for inclusion:

1. call rate $\geq 95\%$
2. MAF $\geq 1\%$
3. HWE $P \geq 10^{-30}$
4. imputation score ≥ 0.7

Human reference genome build GRCh37 was used for all genetic analyses.

POP-GWAS

As mentioned previously, the UK Biobank (UKBB) offers a range of image-driven abdominal phenotypes derived from various imaging modalities available within the dataset. The phenotypes related to organ volumes were acquired from DIXON images as part of the study conducted by Liu et al. [38]. That study utilised the DIXON imaging cohort available at that point, encompassing 32,860 subjects. Currently, DIXON images for 69,296 subjects are available that provides the opportunity to expand the cohort size for the GWAS on these traits. This thesis attempts to predict the phenotype values of the missing subjects and utilises these predictions to impute the phenotypes for conducting GWAS on an enlarged cohort. Given that mixing "real" and imputed phenotypes may introduce bias into the analysis, POP-GWAS was employed to mitigate this issue.

It has been demonstrated recently that machine learning inferred phenotypes (using any methodology) can lead to increased false discovery rates (FDR). POP-GWAS implements a means to account for such a bias, and has theoretical guarantees that a GWAS performed with additional predicted/imputed phenotypes should at least match, if not exceed the power of the original GWAS[43].It combines three sets of GWAS estimates: the effect of SNPs on imputed phenotypes in unlabelled samples, the effect on observed phenotypes in labelled samples, and the effect on imputed phenotypes in labelled samples. The framework uses the difference between the estimated SNP effects on observed and imputed phenotypes in labelled data to correct biases in ML-assisted GWAS. POP-GWAS does not rely on assumptions about the imputation procedure, making it robust even with 'black box' ML algorithms, and works directly with the summary statistics obtained from the three types of GWAS mentioned earlier.

As of October 2023, UK Biobank provided abdominal phenotype values for 33,039 individuals (i.e. labelled samples), while the remaining 25,245 subjects with DIXON MRIs did not have any phenotype values available (i.e. unlabelled samples). In this thesis, these two cohorts of samples were utilised for the three GWAS estimates as explained earlier.

3 | Results

3.1. Overview

self-supervised learning has shown promise in the ability to show emergent properties when trained on large collections of natural images. These emergent properties, learnt in the form of a latent factor or embedding, capture semantic concepts in the image, such as location, classification and size of objects. Here, I applied a state of the art self-supervised learning method, DINOv2, to a large dataset of whole body MRI volumes from the UKBB. My aim was to see whether embeddings learnt from such scans could identify semantic concepts from the MRI that are relevant to a subjects health, disease status, or phenotypes that can be quantified from the images. These phenotypes, if predictable, can then be used in downstream genetic analyses to add power to previous results that had been conducted on a smaller sample size cohort. I describe both the qualitative and quantitative results of applying DINOv2, as well as detailing methodological considerations I made, including optimisations to the original model and its training.

DIXON dataset size

At the start of the project, 52,773 MRI scans, with 47,898 unique subjects, were available. To train the models, this data was split into Train/Val/Test sets with proportions of [0.70, 0.15, 0.15]. An additional set of scans with 21,665 new subjects became available only at later stages of the project; only the GWAS analysis used inferred traits on these scans.

3.2. Variational Autoencoders fail to capture whole body information

The large size of the UKB Whole body MRI dataset ($\geq$ 55TB) requires particular considerations to be taken when conducting any analyses. Training a model on the 3D scans requires trade-offs in terms of model complexity, to enable us to exploit the use of GPUs. These limitations are not only restricted to the overall model architecture size

or its parameter count, but also to hyper-parameters such as the batch size used and learning rate, leading to long training times. These issues are particularly problematic in a Self-Supervised framework, were a simple model might only converge to learn trivial representations.

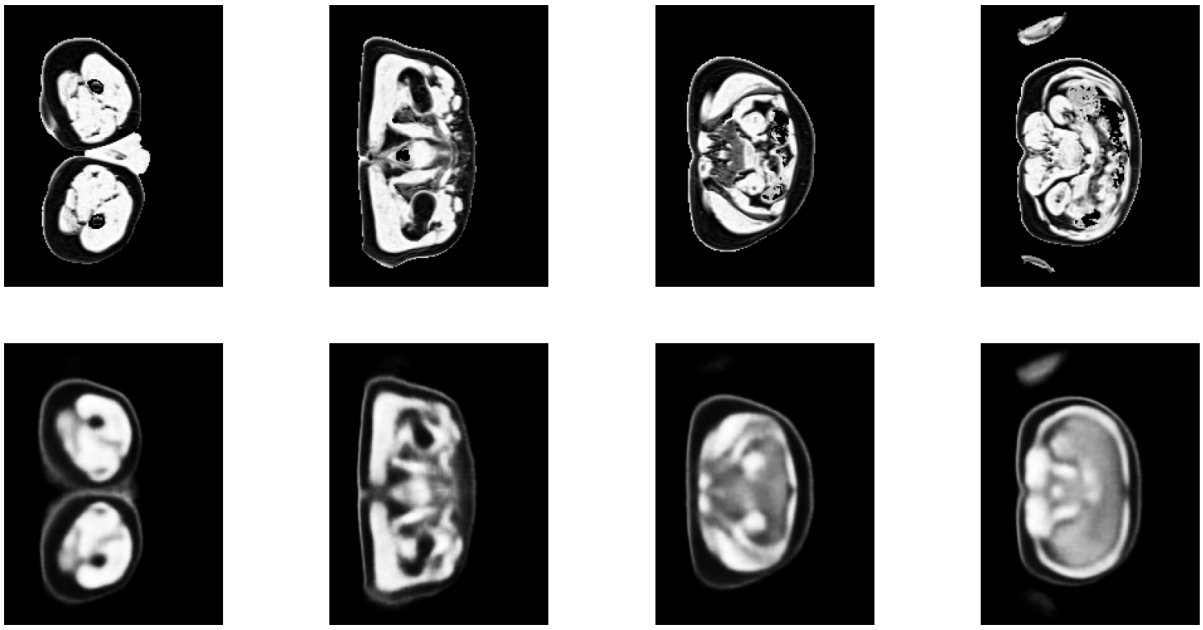


Figure 3.1: $\text{VAE}^{(10)}_{WFP}$ reconstructions. (top row) Representative slices are shown from the original MRI of an example participant. (bottom row) The corresponding reconstructions from $\text{VAE}^{(10)}_{WFP}$. model inference are displayed

To overcome this issue, a divide-and-conquer approach was used. Each 3D volume was considered as a set of sub-views (i.e. 370 slices) and the model is trained on these sub-views / slices. The whole volume representation of the subject is then obtained by aggregating over all of the sub-view / slice representations it is composed of.

The first model architecture that was trained on the data was a factor VAE (2.2.1), following the train procedure described in 2.1.4. Three models was trained, each using one of the channel pairs, *In-Phase/Opposed-phase*, *Water/Fat*, *Water Percent/Fat Percent*. The latent dimension was set to 10; this small dimension size had two objectives: 1. To provide a very simple baseline, and 2. To avoid having a large dimensional subject-wise representation. Since a 3D-scan has 370 slices, even with a small slice-wise latent size the subject will be represented by a 370×10 vector. To evaluate the models, we compared the structured similarity index metric (SSIM). Of the three models evaluated, the *Water Percent/Fat Percent* model achieved the best reconstruction score, with a SSIM of 85.5%, *Water/Fat* and *In-phase/Opp-phase* had an SSIM of 85% and 83%, respectively.

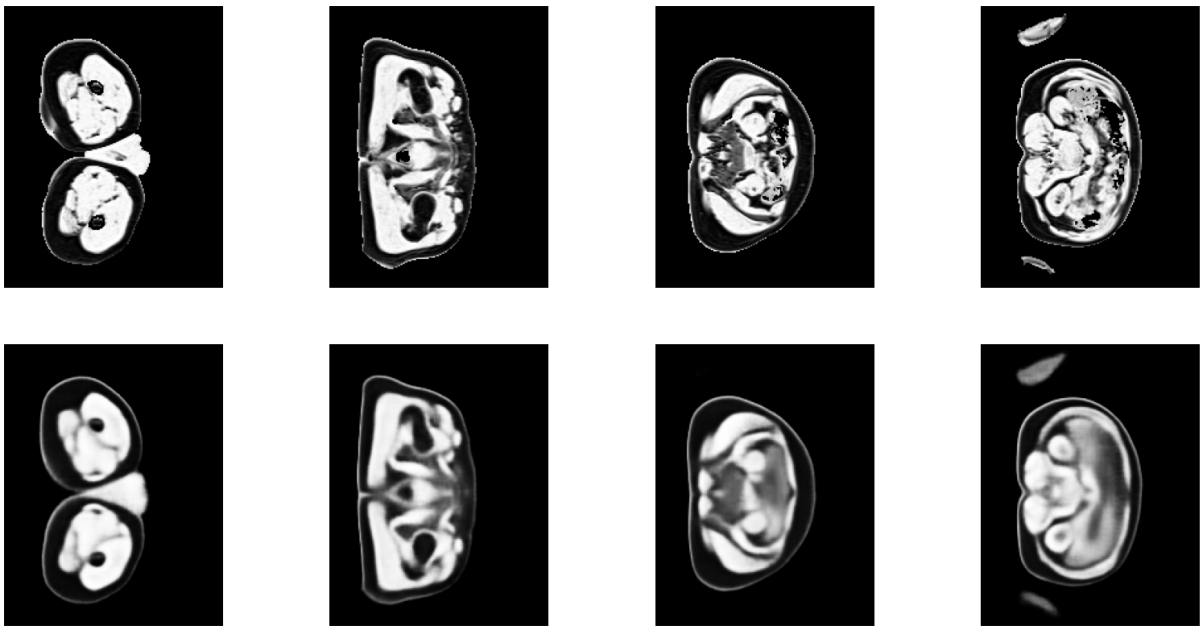


Figure 3.2: $\text{VAE}^{(64)}_{WFP}$ reconstructions. (top row) Representative slices are shown from the original MRI of an example participant. (bottom row) The corresponding reconstructions from $\text{VAE}^{(64)}_{WFP}$. model inference are displayed. These reconstructions demonstrate the additional capacity when going from a 10-d to 64-d latent factor.

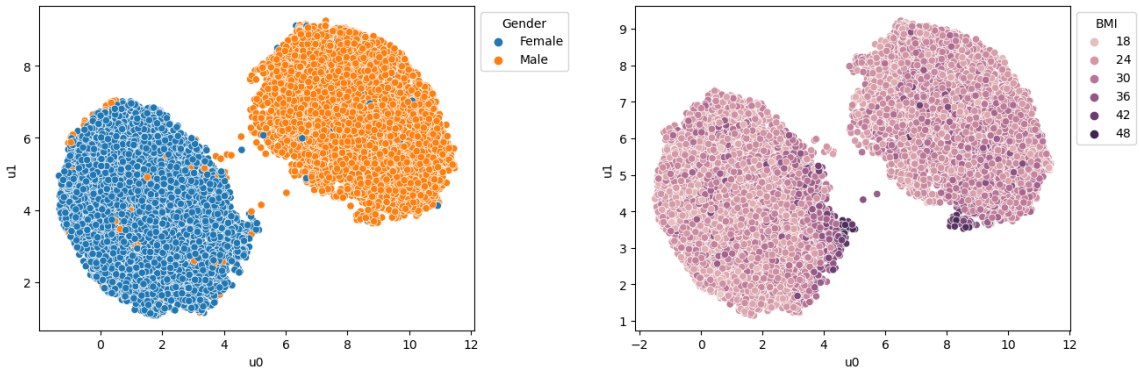
The reconstruction, while achieving a high score, clearly lacked any fine details, only the muscles were well defined.

A second set of model were trained, increasing the latent dimension to 64. The model trained on Water Fat Percent (WFP) still achieved the best performance[1], with a SSIM of 89.5%. Figure 3.2 shows the reconstructions of this model.

While the model achieved an SSIM of nearly 90%, the reconstructions still lacked any fine details, especially in the abdominal cavity. This can be explained from the large amount of background present in each slice. At this stage, a proper evaluation procedure for the embeddings was not yet defined; any label would only apply to the whole 3D volume, to be used the slice embeddings must be aggregated. Multiple architecture can be used for this task, like weakly supervised model or Multiple instance learning, but also simple model for sequential data. However all this approach would be useless if the slice embedding are not capturing meaningful information, as such, a qualitative analysis is warranted.

In the case of both VAE models, the embeddings showed similar structure. Figure 3.3

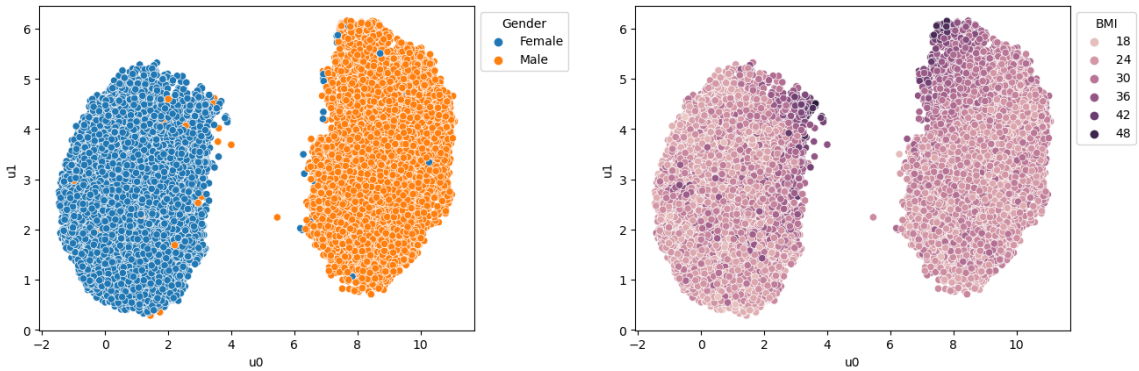
[1]Only the WFP model was trained to convergence due to technical issues that interrupted all three models during training. The loss of all three models resembled the loss observed during the 10-d latent factor VAE training. As it was clear that the WFP channel would still achieve the best performance it was the only model in which training was resumed



(a) Genetic sex (b) BMI

Figure 3.3: $\text{VAE}^{(10)}_{WFP}$ embeddings umap: (a) shows clusters labeled by genetic sex , (b) cluster are labeled by BMI. A clear separation amongst sex is visible, but also a trajectory related to increasing BMI is also present for both sexes.

shows the umap plot of the $\text{VAE}^{(10)}_{WFP}$ embedding's global representation, obtained as described in 2.3. While the genetic sex is clearly identifiable, other major body composition information, such as BMI, show more subtle differences in which for both males and females, there is a BMI trajectory.(5.2) Comparable results are achieved for the $\text{VAE}^{(64)}_{WFP}$(Figure 3.4).



(a) Genetic sex (b) BMI

Figure 3.4: $\text{VAE}^{(64)}_{WFP}$ embeddings umap: (a) shows clusters labeled by genetic sex , (b) cluster are labeled by BMI. Whilst the same seperation by sex is present, a more nuanced trajectory for BMI is observed as compared to the $\text{VAE}^{(10)}_{WFP}$ latent factors.

3.3. self-supervised learning results in improved latent representation quality

We wanted to assess whether increasing model capacity and changing model architecture could lead to more powerful representations. The three networks in which the FactorVAE is composed of: an encoder, decoder and discriminator, are extremely simple Convolutional Neural Networks (CNNs); even the FactorVAE has been superseeded by additional architectures, which improves both on the latent representation and the reconstruction quality achievable. Examples of such models are the Masked Autoencoder (MAE) [24] or Diffusion-based models such as DiffAE [45]. Despite this, training a FactorVAE model on the Whole Body scan's sub-view dataset take around 180 GPU hours[2]. This hinders the ability to iterate and test many different model architectures and hyperparameters.

Instead of focusing on incremental improvements to these models, we decided to assess the current[3] state-of-the art self supervised model, DINOv2[4].

DINOv2 was trained on 142 million natural images and then fine-tuned on the ImageNet dataset. This provides an excellent baseline set of parameters to fine-tune from. Fine-tuning is essential, due to the difference between natural images and MRI. To assess this quantitatively, we used the pre-trained DINOv2 model to obtain embeddings, similar to how the VAE was used in previous sections. Figure 3.5 demonstrates that even without fine-tuning, compared to the VAE models, the Global PCA for DINOv2 CLS tokens shows better overall structure. Whilst DINOv2 still captures the separation in sex, it also more clearly delineates differences in BMI. Nevertheless, looking at subfigure 3.5b it's clear that the pretrained DINOv2 struggles with the specific features of MRI; with many PCs focusing on background features rather than the body in the foreground.

The pretrained model used in DINOv2 was VIT-base. However, as the computational requirements were too demanding to fine-tune two VIT-base architectures simultaneously a strategy to simplify the architecture and therefore its training was required. A custom VIT, matching the architecture of the DINOv2 VIT small, but with a smaller embedding size, was obtained by following the distillation procedure described in 2.2.3.

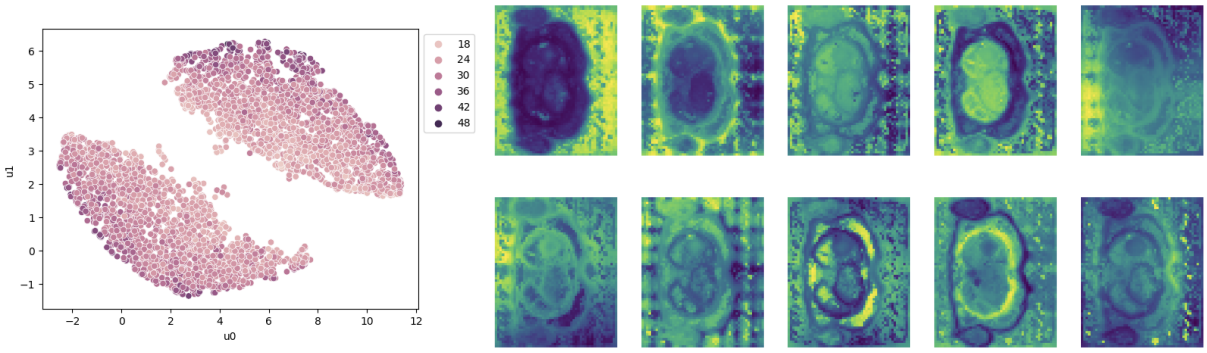
Since it was not feasible to train three different versions of the model to assess differences between channel pairs, only *Water Percent/Fat Percent*, which achieved the best loss scores on both VAE models was used. However, since the model expects three channels

[2]Model training performed with a 32GB Nvidia V100

[3]as of April 2023

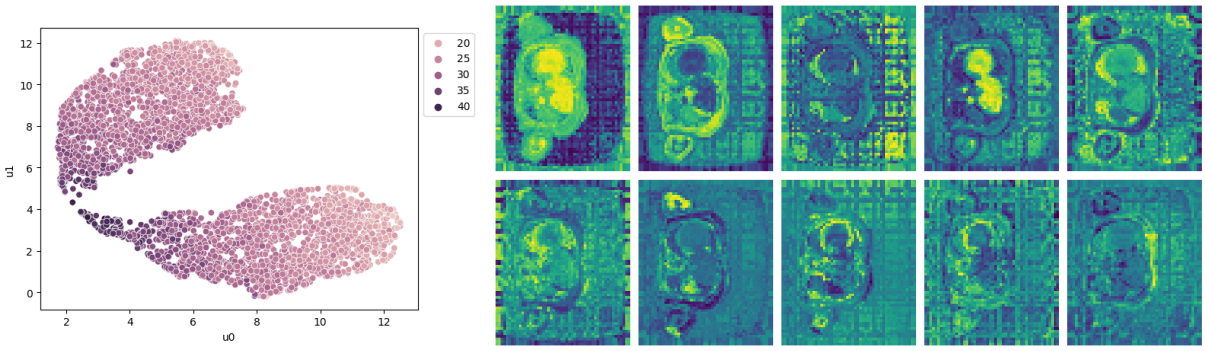
[4]At the time of it's publication, DINOv2 achieved state-of-the art performance on ImageNet top-1 classification compared with other SSL models

as input, the *In phase* channel was also added as an additional channel.



(a) global PCA (b) patch embeddings PCs

Figure 3.5: Pretrained DINO qualitative plots: (a) show the Global embedding PCA labeled by BMI, the two clusters show separation by Genetic Sex. (b) patch embeddings Principal Components(2.3)



(a) Global Embeddings (b) patch embeddings PCs

Figure 3.6: Fine-tuned DINO qualitative plots: (a) umap plot of the Global embedding PCA labeled by BMI, the two clusters show separation by genetic sex. (b) Patch embeddings Principal Components(2.3)

Using the distilled model as a base model, fine-tuning was performed, as described in 2.2.3. Figure 3.6 show the qualitative results. Compared to the pretrained DINOv2 model, the umap has very similar characteristics. The patch-embedding PC demonstrates an improvement, having an increased focus on the body rather than background MRI features, however a lot of artifacts were still present on the background area.

One possible explanation for this behavior is the DINO crop-based objective, which can be thought of as learning a crop-invariant representation. Since the local crops can be very small and are sampled randomly in the image, many of them will contain only back-

ground. The multi-crop with random scale strategy probably avoids the representation to completely collapse to a trivial solution, but the representation of the background remains dominant due to its abundance.

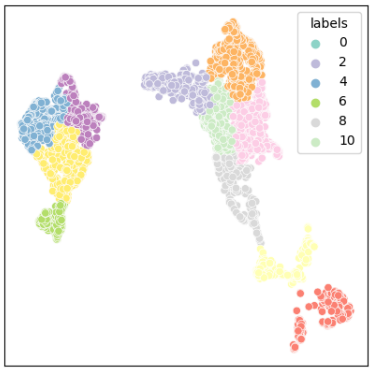


Figure 3.7: Patch-embeddings umap

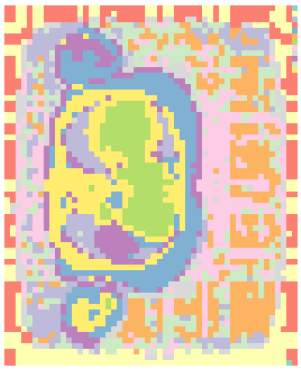


Figure 3.8: patch embeddings PCs

Figure 3.9: UMAP of patch embeddings and visual representation of patch embedding PCs displayed as an image. Patch embeddings clearly form clusters, and clusters loosely represent key semantic concepts, like organs (green, lilac), fat (blue/purple) and muscle (yellow).

To better understand what the model is capturing, the patch embeddings can be clustered together, and see where the clusters map to on the original image slice (2.3). In the umap plot in 3.7 it is clearly visible the separation of the background area, represented by the big cluster on the right side, from the body-specific embeddings, in the cluster to the left. 3.8 shows a coherent mapping between the patch embeddings, clusters loosely represent key semantic concepts, like organs (green, lilac), fat (blue/purple) and muscle (yellow).

The patch-embedding noise is problematic as the main slice embedding, or CLS token, is analogous to an average of all the patches that make up the slice. Figure 3.10 shows the 8 attention heads from the CLS token. While it is less noisy than the patch tokens previously described, there is still a large amount of focus on the background, and overall, the signal from the body is quite low.

3.4. Custom implementation of DINOv2 to improve representations

To overcome the issue of the fine-tuned version, a custom implementation of DINO was used. The main change was to force the model to completely ignore the background area.

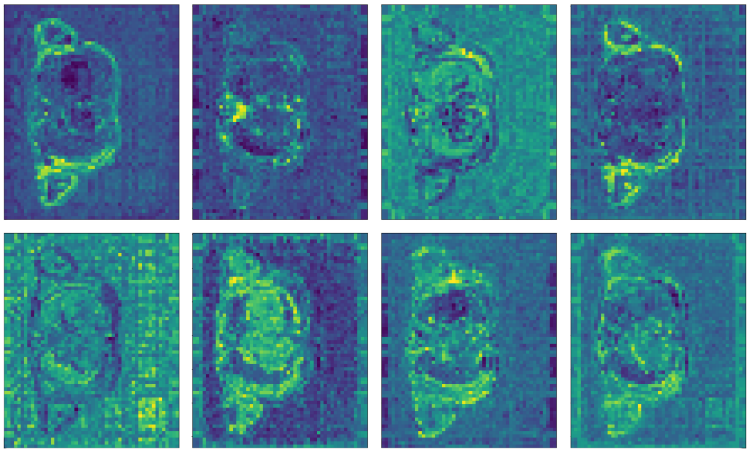
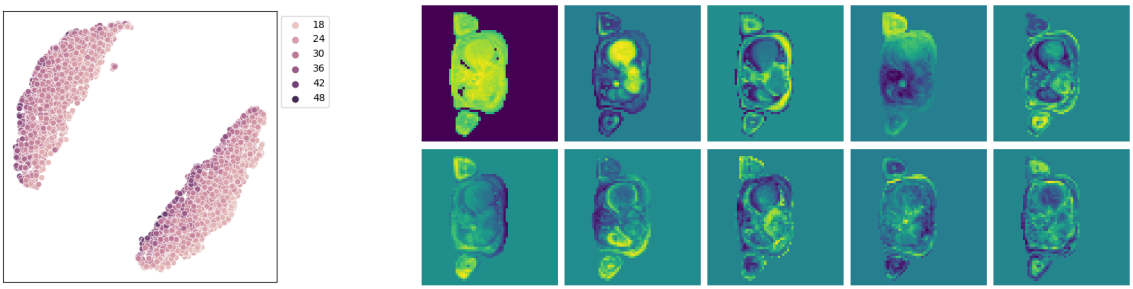


Figure 3.10: CLS token attention heads. Significant amounts of attention are focused on the background noise of the MRI.



(a) Global Embeddings (b) patch embedding PCs

Figure 3.11: Custom DINO qualitative plots: (a) umap plot of the global embedding PCA labeled by BMI, the two cluster show the divison by Genetic Sex. (b) patch embeddings Principal Components(2.3)

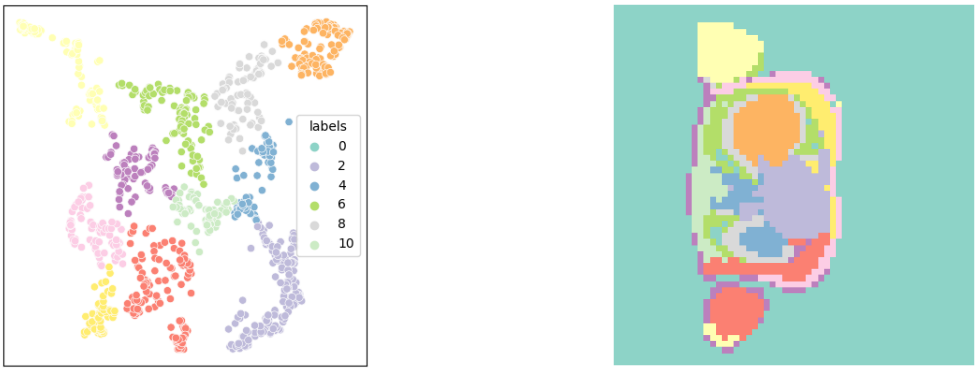
The channels used as input were changed to the *Water/Fat*. This was motivated by a loss of the absolute signal intensity in the *Water Percent/Fat Percent*. Two tissues, both having 100% water composition might have different signals in the *WF* channel, but would appear the same in the *WFP* channel.

Other modifications involved increasing the input slice thickness; instead of using an individual 2d slice, three slices were used and the slices were stacked on the channel dimension to keep the original 2D architecture of the model.

Moreover the lower section of the 3D scan was ignored when extracting the slices. This section should always contains only the legs region, the cutoff point as choose conservatively to have a large margin to the pelvis. This reduction has two reason, first it reduce by a third the training time, but the main reason is that avoid a potential bias toward

a leg centered representation, having both a simpler and more uniform stucture than the abdominal section, the model might neglect the latter.

These changes rendered the model incompatible with the DINOv2 published pretrained weights. These changes, together with other technical modifications, significantly improved the required training time, allowing faster iteration and better tuning of the hyper-parameters (see methods 2.2.3).



(a) Patch-embeddings umap (b) patch embeddings PCs

Figure 3.12

Repeating the analysis conducted on the fine-tuned DINOv2 model previously described in section 3.3 with the customised model does indeed reveal cleaner results. By forcing the model to ignore the background, a much stronger signal was obtained from the body area, this was visible for both the patch-embeddings (Figure 3.11b) but more importantly in the CLS token attention maps (Figure 3.13). The effect of this noise reduction can be appreciated in Figure 3.12b, the latent space is fully utilized, and the patch embeddings umap clustering demonstrate a cleaner representation, where both the heart and liver are clearly delineated.

3.5. Whole body aggregated embeddings are predictive of relevant traits

The analysis performed in the previous section focus primarily on qualitative results, but do not provide any quantitative measure that assesses the quality of the inferred embeddings. To assess the representational capacity of the embeddings, it is typical that a linear layer is added on top of the CLS token, and the model fine-tuned for a supervised regression or classification task. However, in this case, we have a many-CLS token to

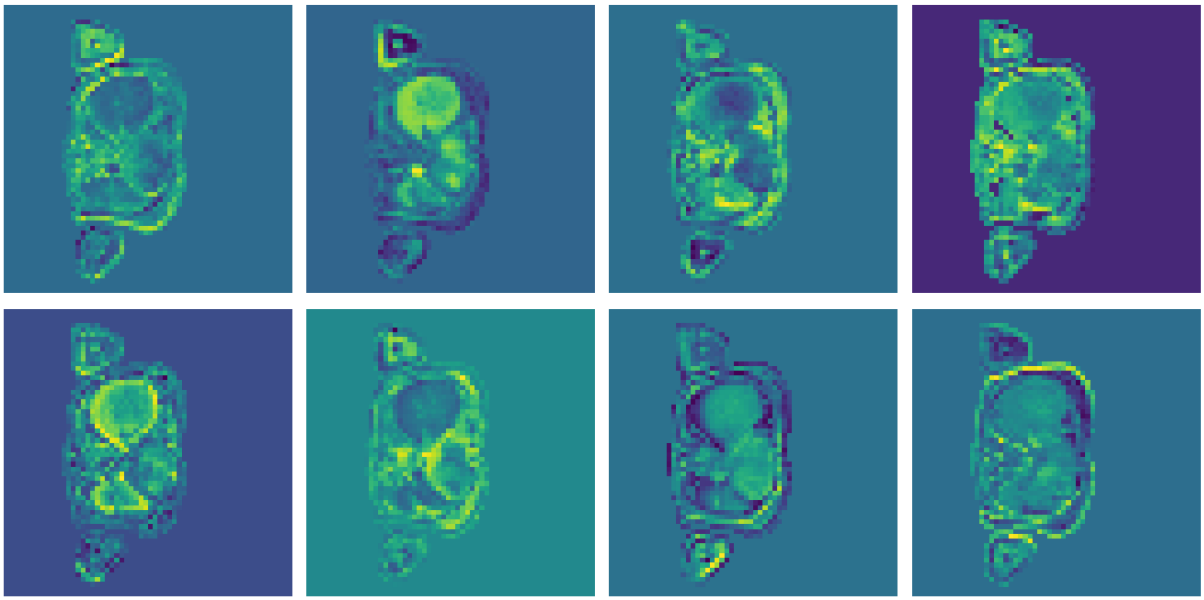
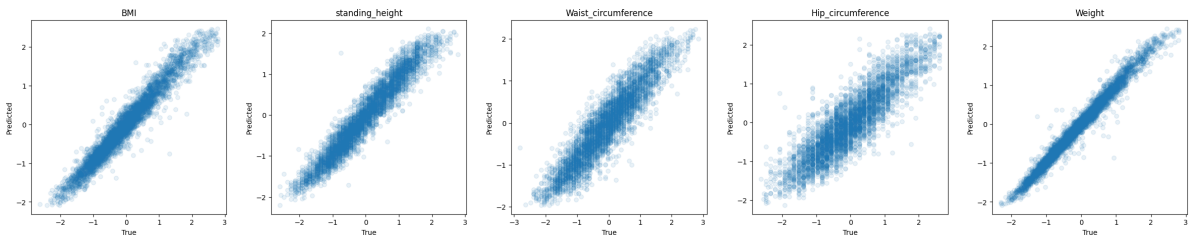


Figure 3.13: CLS token attention heads

outcome mapping: Each subjects MRI is represented by a sequence of CLS tokens. A more complex model therefore was trained using all of a 3D scan's sub-view representations as input to predict a given trait of the individual. Two limitations of this approach is that it's then difficult to understand whether it is the embedding not capturing the relevant trait, or if the CLS aggregation strategy is performing poorly.

A selection of body measurements: BMI (kg/m^2), height (m), waist circumference (cm), hip circumference (cm), and overall weight (kg), were used to benchmark the predictive capabilities of the embeddings.

The aggregation model, described in 2.4 was used to perform a multi-trait regression on these traits, achieving an R^2 score of 0.94, 0.92, 0.85, 0.77, 0.97 respectively.



(a) (b) (c) (d) (e)

Figure 3.14: Body measurement regression: (a) BMI, (b) Height, (c) Waist circumference, (d) Hip circumference, (e) Weight

DXA Traits

A source of more fine grained measurement to test the learned embeddings are the DXA (Dual-energy X-ray absorptiometry) measurements[5]. The measurement of the first(repeated) DXA visit were matched to the first(repeated) MRI. Two multi-trait regressions (2.4) were performed on body fat content and bone mineral content, for a total of 45 traits, achieving an average R^2 of 0.969 and 0.82. The complete results are shown in the supplementary figure 5.3, 5.4 While it achieved good performance, the insights it provides on the embedding quality is limited, as all the traits are very correlated with each other, with an average absolute Pearson correlation of 0.59, as shown in figure 3.15.

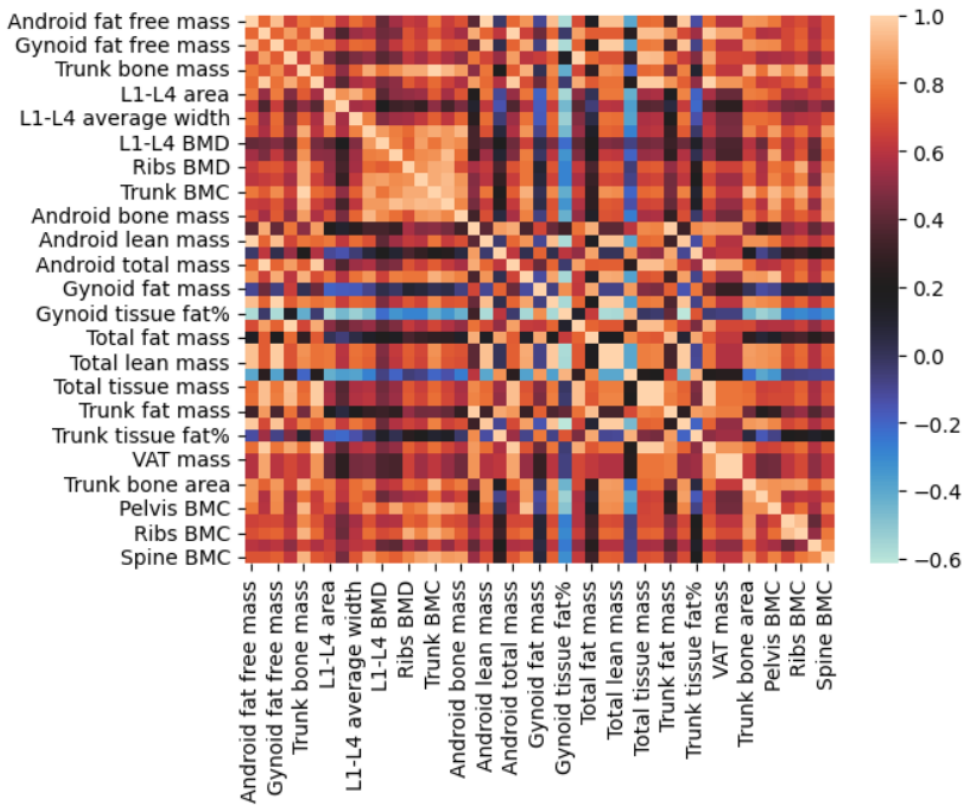


Figure 3.15: DXA body measurement correlation

Abdominal Traits

The Image derived Phenotypes from Liu et al.[38] provided the best test targets to probe the predictive power of the learnt embeddings. As previously mentioned, these IDPs are composed of organs and tissue volumes and composition measurements, of the 11 define traits, 9 were taken into consideration: left and right kidney, spleen, lung, visceral fat,

[5]UKBB Category 124

subcutaneous fat, pancreas volumes(l) and liver's proton density fat fraction(PDFF). Of these traits, the pancreas and PDFF were not derived from the DIXON dataset directly, but specific MR acquisitions of the pancreas. With an average absolute Pearson correlation of 0.36, they are more independent than the DXA traits, providing a more challenging task when used as a prediction target.

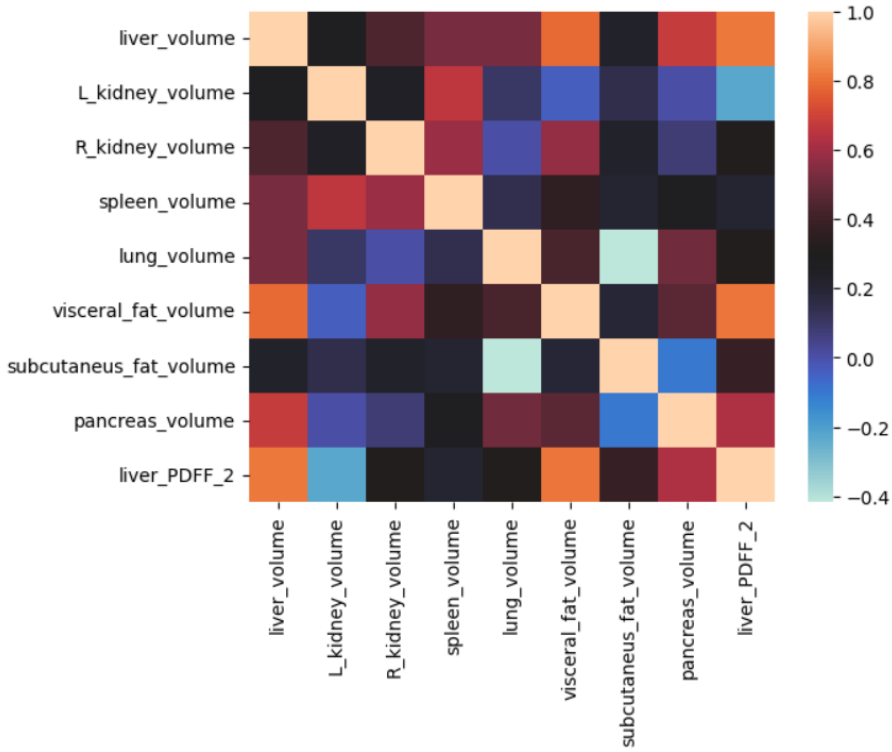
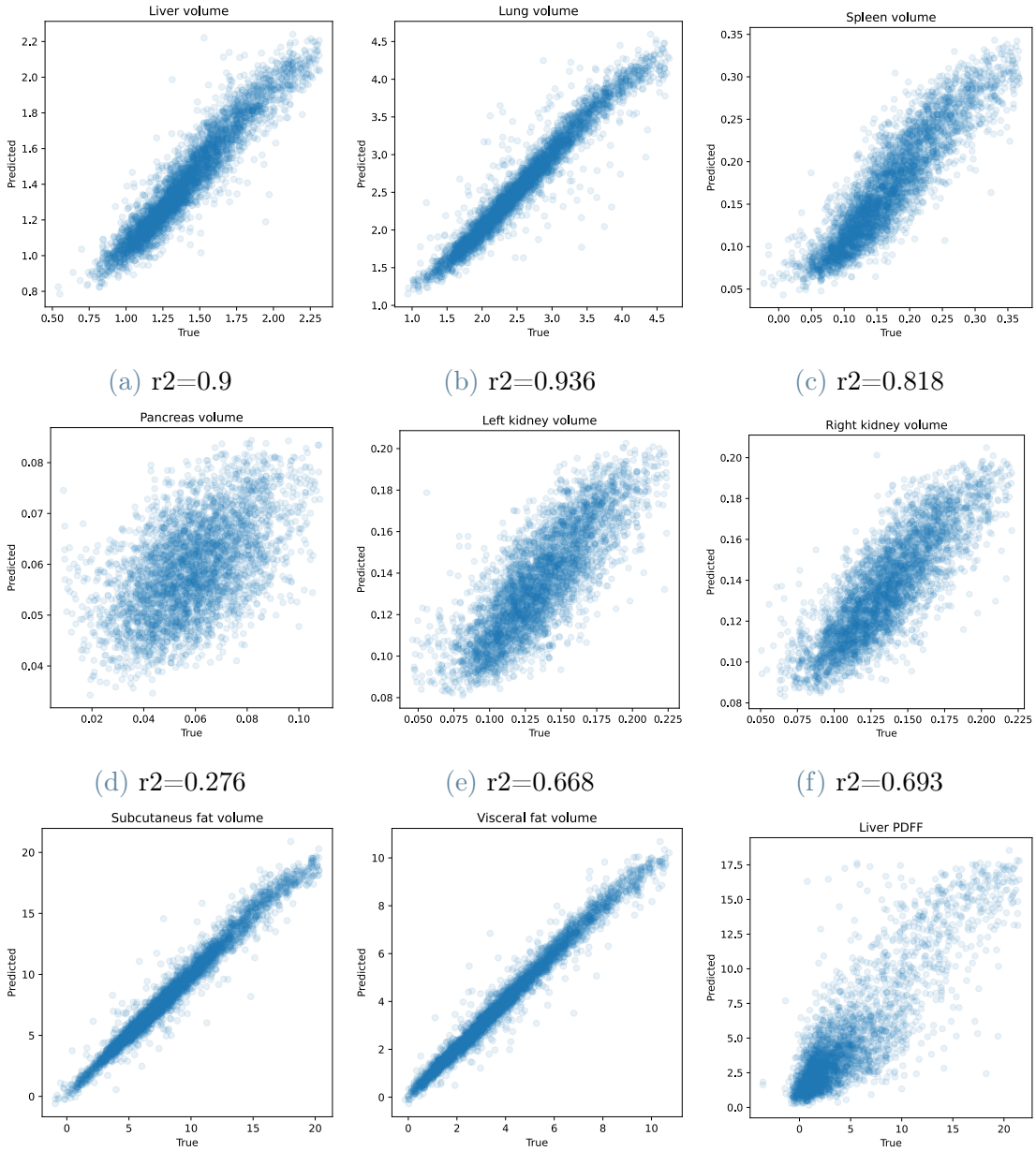


Figure 3.16: Abdominal composition trait's correlation

A multi-trait regression was run with these traits, figure 3.17 shows the result for each of the traits.

The most telling results came from the organ volumes, which proves that 3D structure can be inferred from the representation of 2D inputs. This also highlights a limitation on the dimension of the captured phenotypes; While the dimensions of larger organs such as the lungs and liver can be predicted with high accuracy, as organs become smaller, the predictions become increasingly noisy. The results of the two adipose tissues align with those from body measurements and DXA, where fat-related traits are the most easily predicted. While the PDFF seems to contradict these conclusion, it's worth noting that it was not a trait derived from the DIXON data originally.



(a) r2=0.9 (b) r2=0.936 (c) r2=0.818

(d) r2=0.276 (e) r2=0.668 (f) r2=0.693

(g) r2=0.971 (h) r2=0.978 (i) r2=0.738

Figure 3.17: Abdominal trait regression results

UKBB clinical data

The combination of a large imaging cohort with an extensive clinical and lifestyle data collection provided by the UK Biobank (UKBB) lays the groundwork for gaining new insights into the physiological effects of diseases and health conditions impacting the human body.

Utilizing this abundant collection of health-related data was a primary objective for ap-

plying the learnt embeddings.

The clinical data originates from three primary sources: the subject's self-reported (SR) medical history gathered during the initial UKBB visit, records from General Practitioners (GPs), and Hospital (HI) admission records. Initial data processing was conducted on these tables (see methods 2.1.2) to extract the most prevalent conditions.

However, it became apparent that while these data sources offer an excellent retrospective view of an individual's health history, they are not well-suited for integration with MRI scans. Even if an individual has been diagnosed with a certain condition in the past, by the time they undergo imaging, they may have fully recovered, and the corresponding physical phenotype may have disappeared. Since most diagnoses were recorded years before imaging, it's unclear how to define which subjects still carry the disease trait. Trying to limit the time span between diagnosis and scan could eliminate this discrepancy, however this will drastically reduce the sample size per disease. Tables 5.2,5.3,5.4 demonstrated the loss of data by using a 10 year interval. The same consideration can be applied to prognostic tasks, with the added complication that the SR records aren't usable, as they are only collected during the UKBB verbal interview stage, before imaging.

While chronic or more serious diseases might remain detectable for an extended period of time after diagnosis, they are relatively uncommon in the UKBB, whose subjects were recruited from healthy volunteers. Moreover, results from abdominal composition prediction indicate that detecting subtle physical changes is challenging, limiting the diseases that can be considered.

For these reasons, the use of these clinical conditions as a direct prediction target was not further explored in this study.

3.6. GWAS on Abdominal Traits

Baseline GWAS (see Sec. 2.5) involving 33,039 subjects (excluding those not passing the various QCs) were conducted using phenotype values available from the UKBB, in order to replicate the analyses performed in the original study [38]. This resulted in the identification of the following number of genome-wide significant loci (i.e. $p < 5 \times 10^{-8}$) - subcutaneous fat volume: 5, visceral fat volume: 16, lung volume: 7, left kidney volume: 20, right kidney volume: 12, spleen volume: 190, and liver volume: 46. Among these, the following number of loci were also deemed study-wide significant (i.e. $p < (5 \times 10^{-8}/7) = 7 \times 10^{-9}$): subcutaneous fat volume: 3, visceral fat volume: 6, lung volume: 6, left kidney volume: 12, right kidney volume: 6, spleen volume: 149, and liver volume: 39. It is worth

mentioning that the original study combined the left and right kidney volumes in their analyses, while this thesis treated them separately.

For the execution of POP-GWAS, two additional sets of GWAS were conducted utilising the predicted phenotype values: one with the baseline cohort and another with the cohort for which the actual phenotype values were unavailable. These two sets of GWAS, along with the baseline GWAS, were utilised to perform POP-GWAS, yielding the final GWAS results involving an effective cohort size of 58,284 subjects (excluding those not passing the various QCs) - introducing 25,245 additional subjects. POP-GWAS identified the following number of genome-wide significant loci - subcutaneous fat volume: 23, visceral fat volume: 32, lung volume: 64, left kidney volume: 32, right kidney volume: 22, spleen volume: 346, and liver volume: 102, and study-wide significant loci - subcutaneous fat volume: 11, visceral fat volume: 24, lung volume: 32, left kidney volume: 21, right kidney volume: 12, spleen volume: 258, and liver volume: 76. The associations identified by POP-GWAS for each of the seven traits are illustrated in Figures 3.18 through 3.24 with the help of Manhattan plots where the genome-wide significant variants are annotated with the names of the closest genes using Ensembl [39], along with the corresponding quantile-quantile (QQ) plots. Furthermore, lead SNPs for the study-wide significant loci identified for each trait, together with their Effect Allele (EA), Frequency, Effect Size (β), Standard Error and P value are detailed in the appendix (Tables 5.5 through 5.11).

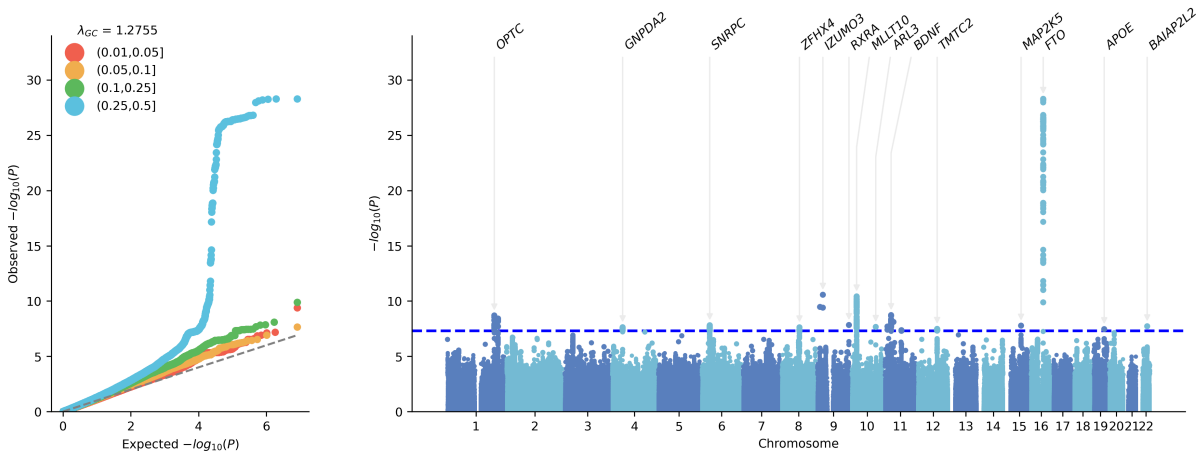


Figure 3.18: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **subcutaneous fat volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

It is evident that POP-GWAS yielded a significantly greater number of significant loci compared to the baseline GWAS. Table 3.1 shows the number of significant loci identified

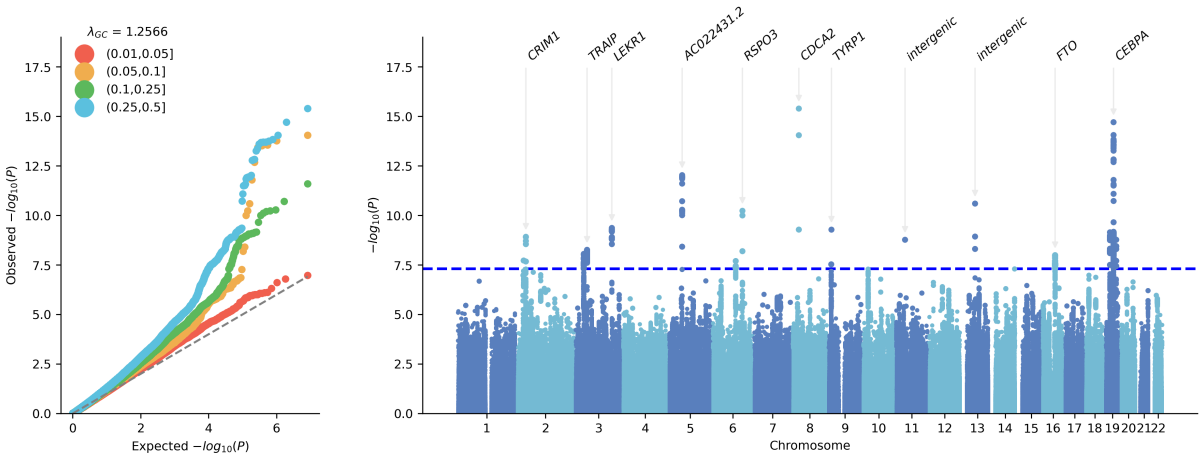


Figure 3.19: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **visceral fat volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

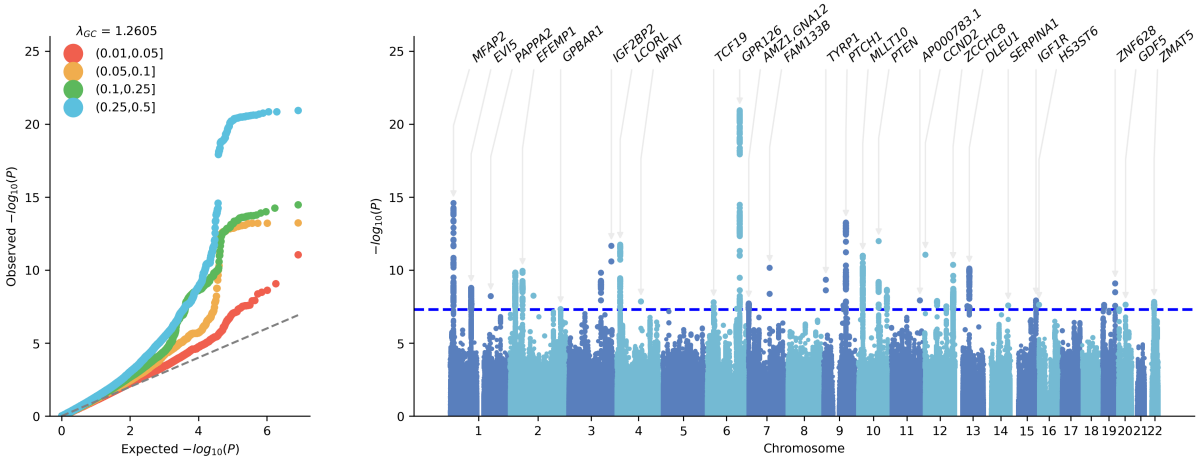


Figure 3.20: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **lung volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

by POP-GWAS, indicating whether they were also identified by the baseline GWAS or not. These results demonstrate the advantage of the increased statistical power in conducting GWAS, facilitating the discovery of additional loci. Moreover, a further analysis was conducted to ascertain whether loci identified by the baseline GWAS were also identified by POP-GWAS - the results are presented in Table 3.2. It was observed that, although the majority of loci identified by the baseline GWAS were also identified by POP-GWAS,

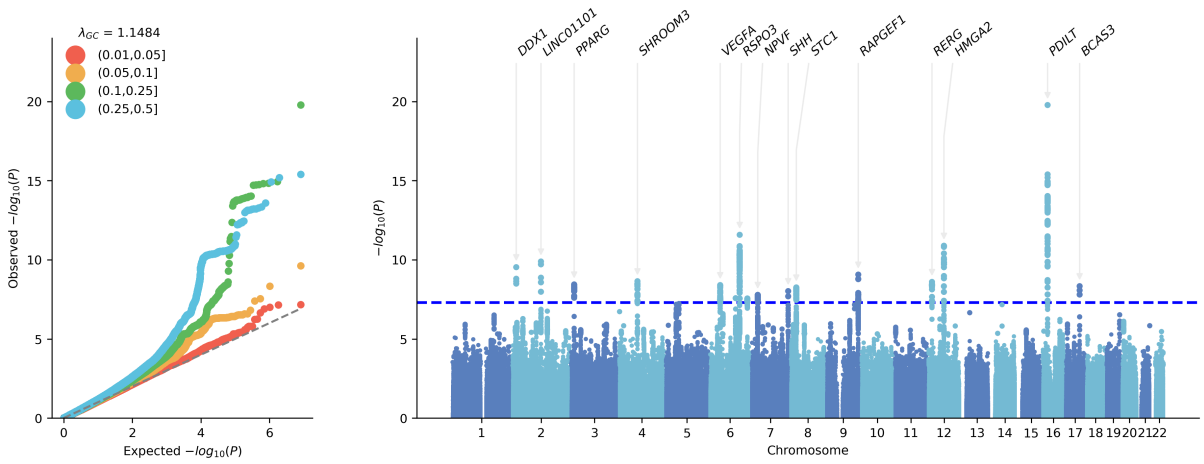


Figure 3.21: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **left kidney volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

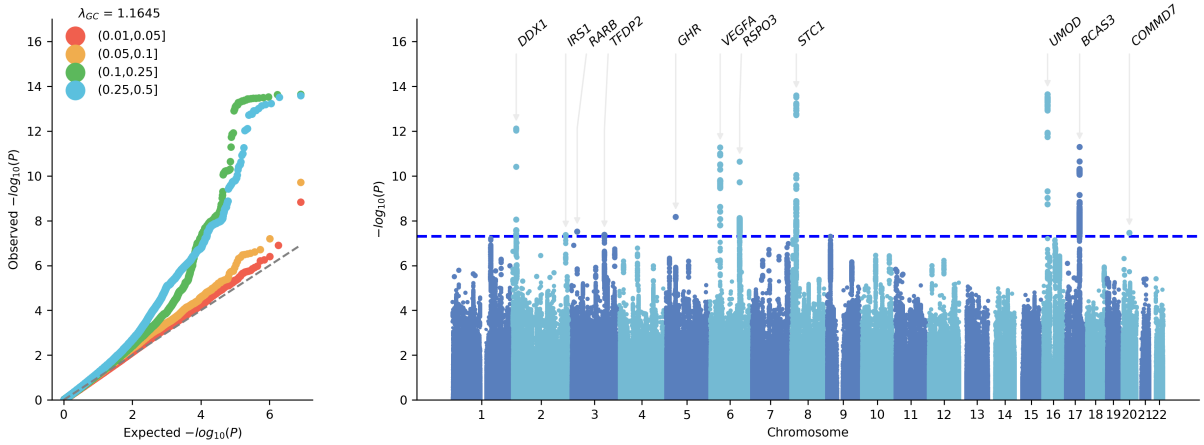


Figure 3.22: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **right kidney volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

a small number of them remained unidentified.

Additional experiments were conducted to evaluate the contribution of POP-GWAS. To this end, two supplementary GWAS were undertaken: one utilising solely predicted phenotype values, and the other employing a combination of real phenotype values (where available) and predicted values (where not). For both instances, the entirety of the available cohort was utilised, consistent with the approach of POP-GWAS. Table 3.3 presents

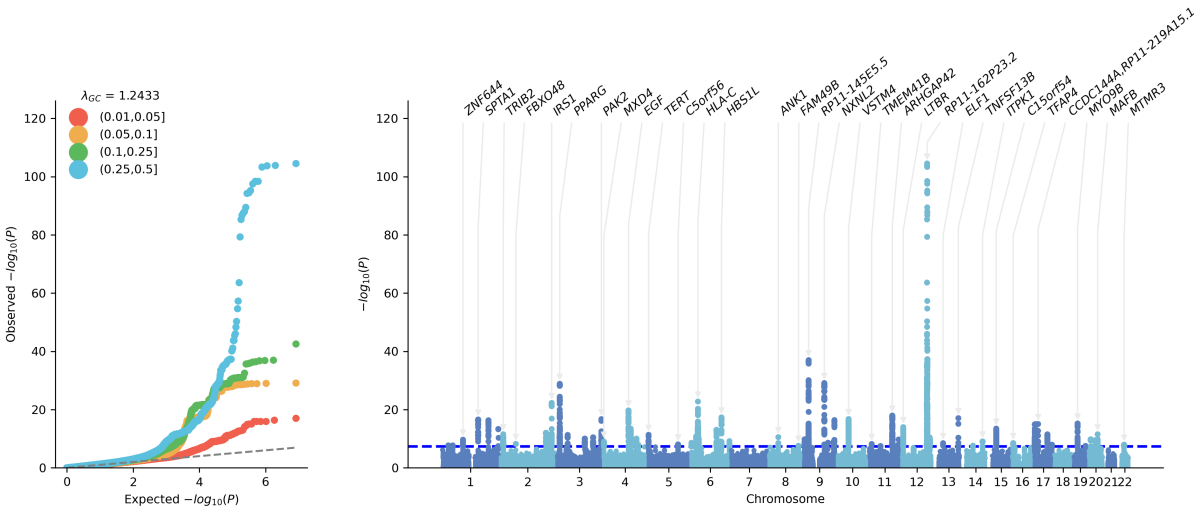


Figure 3.23: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **spleen volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

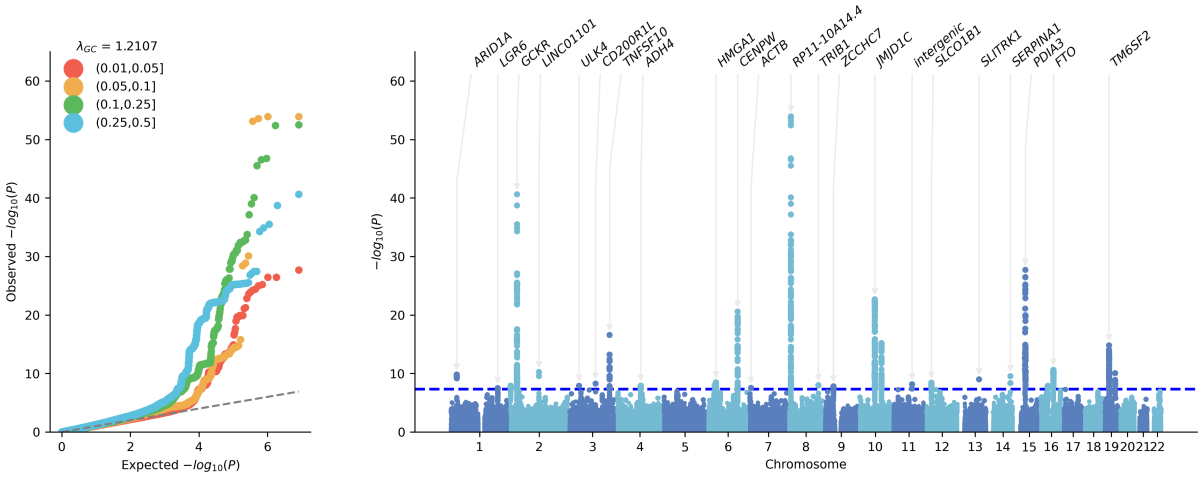


Figure 3.24: On the right, a Manhattan plot illustrating the genome-wide significant associations discovered by POP-GWAS for **liver volume**, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes. On the left, the corresponding Quantile-Quantile (QQ) plot.

the number of significant loci identified by POP-GWAS, which were not recognised as significant in the GWAS conducted with predicted phenotype values alone and in the GWAS incorporating both real and predicted phenotype values. It was observed that a significant number of loci were absent from both these GWAS. This can be ascribed to the inherent imperfections in the predicted values. Consequently, when GWAS were

Table 3.1: The number of loci found significant following the execution of POP-GWAS on the entire available cohort with the imputed phenotypes that were either present or absent from the set of significant loci identified after conducting GWAS on the available phenotypes (baseline, according to [38])

	Genome-wide significant ($P < 5 \times 10^{-8}$)		Study-wide significant ($P < 7 \times 10^{-9}$)	
	Present	Absent	Present	Absent
Subcutaneous fat volume	5	18	3	8
Visceral fat volume	16	16	9	15
Lung volume	8	56	7	25
Left kidney volume	17	15	12	9
Right kidney volume	9	13	4	8
Spleen volume	210	136	159	99
Liver volume	55	47	45	31

executed using solely predicted values, this "noise" detrimentally impacted the results. When GWAS were performed utilising a combination of real and predicted phenotypes, the phenotype values for some subjects were "noise-free", whereas others were affected by "noise". This approach resulted in fewer missing loci compared to POP-GWAS. However, it is imperative to acknowledge that employing a direct amalgamation of real and predicted phenotype values introduces a bias into the dataset - a distinction between the subjects with real versus predicted values.

GWAS on Learnt Embedding

The obtained CLS tokens were further utilised to derive 64 endophenotypes or latent phenotypes by supplying the tokens to an autoencoder (see Sec. X). These latent phenotypes are not constrained by the abdominal traits, thus enabling the potential capture of a broad spectrum of interesting phenotypes. GWAS utilising these latent phenotypes were conducted and a total of 426 genome-wide significant loci were identified across 64 latent phenotypes, of which 399 possessed more than one SNP in LD. The associations (considering only the loci with more than one SNP in LD) identified by the GWAS across 64 latent phenotypes are illustrated in Figure 3.25 with the help of a Manhattan plot where the genome-wide significant variants are annotated with the names of the closest genes using Ensembl [39]. Finally, Table 3.4 presents the counts of unique SNPs associated with specific traits (obtained using to LDtrait with $R^2 >= 0.5$), derived from the genome-wide significant loci. It was noted that various traits corresponding to different body parts were

Table 3.2: The number of loci found significant after conducting GWAS on the available phenotypes (baseline, according to [38]) that were either present or absent from the set of significant loci identified following the execution of POP-GWAS on the entire available cohort with the imputed phenotypes.

	Genome-wide significant ($P < 5 \times 10^{-8}$)		Study-wide significant ($P < 7 \times 10^{-9}$)	
	Present	Absent	Present	Absent
Subcutaneous fat volume	3	2	3	1
Visceral fat volume	13	3	6	0
Lung volume	7	0	6	0
Left kidney volume	17	3	11	1
Right kidney volume	9	3	4	2
Spleen volume	183	7	143	6
Liver volume	45	1	38	1

associated with the significant loci, with the highest number of associations observed for body weights and measures, followed by bone measurements. These findings are in-line with the fact that the input DIXON images were "full body" images, exhibiting a range of different organs. Further investigation is warranted to elucidate the implications of the significant loci identified.

Table 3.3: The number of loci found significant following the execution of POP-GWAS on the entire available cohort with the imputed phenotypes that were either present or absent in the GWAS performed using predicted and real + predicted phenotypes

	Predicted phenotype				Real+predicted phenotype			
	Genome-wide significant ($P < 5 \times 10^{-8}$)		Study-wide significant ($P < 7 \times 10^{-9}$)		Genome-wide significant ($P < 5 \times 10^{-8}$)		Study-wide significant ($P < 7 \times 10^{-9}$)	
	Present	Absent	Present	Absent	Present	Absent	Present	Absent
Subcutaneous fat volume	12	11	6	5	12	11	7	4
Visceral fat volume	23	9	16	8	27	5	17	7
Lung volume	46	18	22	10	46	18	23	9
Left kidney volume	12	20	10	11	23	9	14	7
Right kidney volume	4	18	1	11	14	8	7	5
Spleen volume	257	89	198	60	308	38	237	21
Liver volume	61	41	47	29	83	19	61	15

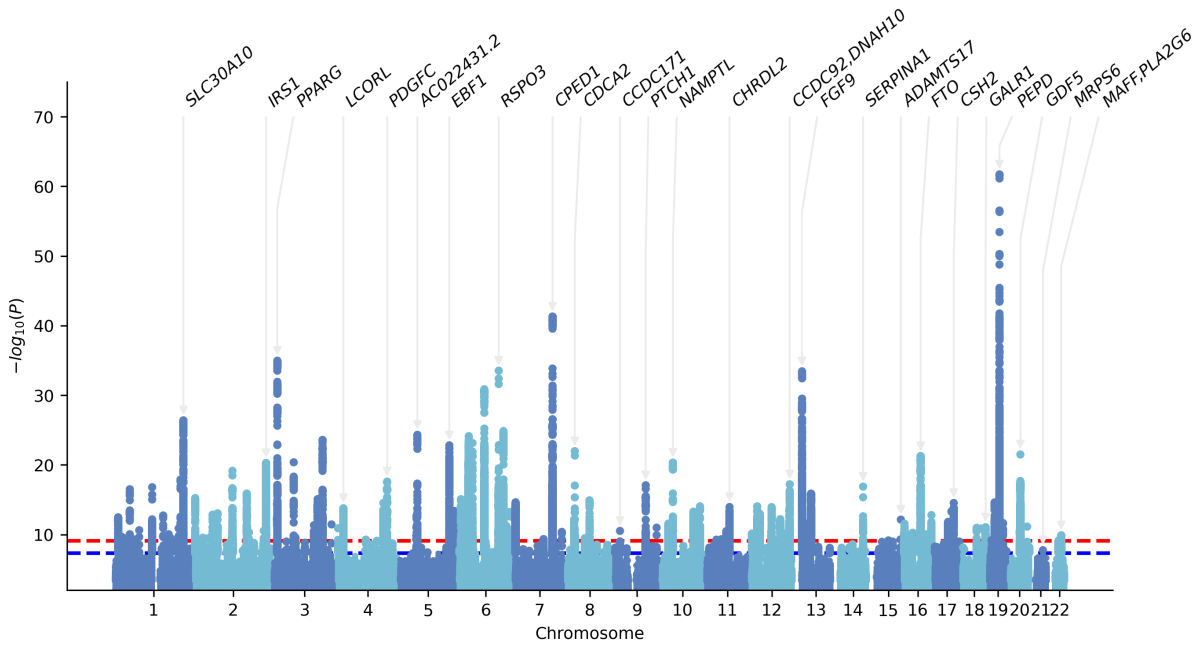


Figure 3.25: Manhattan plot illustrating the genome-wide significant associations discovered by GWAS across 64 latent phenotypes, variants selected using a sliding window with a window size of 50Mb, annotated with the names of the nearest genes.

Table 3.4: LDtrait analysis: number of genomic associations of the 64 latent phenotypes obtained by applying an autoencoder on the CLS tokens

Ontology tags	Trait	# SNPs
EFO_0004324	Body weights and measures	305
EFO_0004512	Bone measurement	220
EFO_0004503	Haematological measurement	190
EFO_0005105	Lipid or lipoprotein measurement	166
EFO_0004298	Cardiovascular measurement	150
EFO_0003892	Pulmonary function measurement	117
EFO_0004340	Body mass index (BMI)	115
EFO_0000319	Cardiovascular disease	92
EFO_0004464	Brain measurement	90
EFO_0004303	Vital signs	88
EFO_0002461	Skeletal system disease	73
EFO_0003923	Bone density	72
EFO_0004260	Bone disease	61
EFO_0004764	Visceral adipose tissue volumes	56
MONDO_0005148	Type 2 diabetes mellitus	56
EFO_0003777	Heart disease	50
EFO_0001645	Coronary artery disease	32
EFO_0000275	Atrial fibrillation	14
EFO_0000537	Hypertension	12
EFO_0000612	Myocardial infarction	7
EFO_0003931	Bone fracture	3
HP_0001627	Abnormal heart morphology	2
EFO_0009531	Aortic valve disease	1

4 | Conclusion

The primary focus of this thesis has been the application and extension of the DINOv2 self-supervision framework and what we can learn from full-body MRI scans at a population level. The primary goal of this thesis was to assess whether, like with natural images, it is possible to learn rich, multi-use case representations that can be utilised for downstream clustering, segmentation and prediction tasks.

In this thesis, I have demonstrated that VAEs lack sufficient complexity to learn embeddings of 3D MRI, and whilst they can separate subjects based on coarse phenotypic characteristics like sex and BMI, self-supervised learning (e.g DINOv2) demonstrated superior performance. Additionally, I have shown that it is necessary to fine-tune self-supervised learning algorithms, as out of the box, non-fine tuned models lacked sufficient representational capacity. This is likely due to MRI images being out of domain with respect to what these models were originally trained on, e.g. natural images. Additional considerations should be taken when training self-supervised models on 3D MRI data. I have shown that by modelling multiple slices as additional channels, improves the CLS embeddings obtained with respect to single-slice models. This is likely due to the model being able to better capture local correlative structure through space and therefore learn to segment different semantic features present in the human body, such as organs, fat and muscle content. This suggests that as GPU power scales, representing entire volumes would be a promising approach to better capture semantic and medically relevant concepts from whole-body MRI data.

Through additional changes to the original model, for example limiting the self-attention to focus only on areas of the body, guided by use of a mask, I was able to reduce noise and improve the learnt embeddings for use in several downstream tasks. I went on to demonstrate that these embeddings learnt spatially sensible clusters, are predictive of several MRI and DXA derived traits, and by utilising these predictions through GWAS, one can perform better powered genetic analyses. Future work will focus on interpreting these novel GWAS loci discovered and seeking suitable replication cohorts.

5 | Appendix

Custom DINO implementation

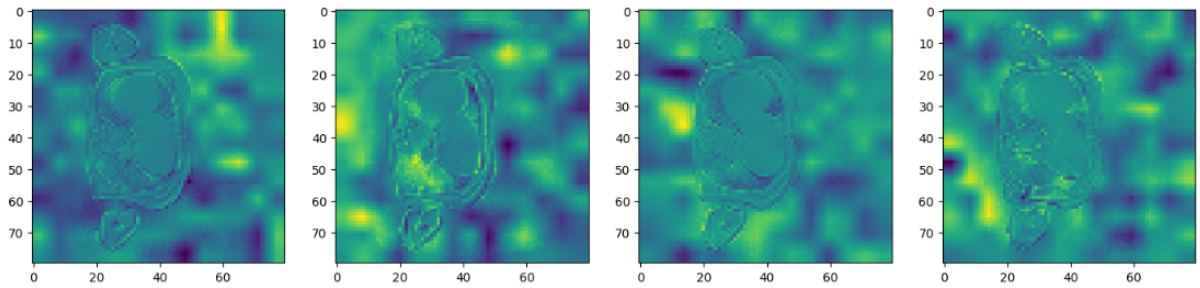
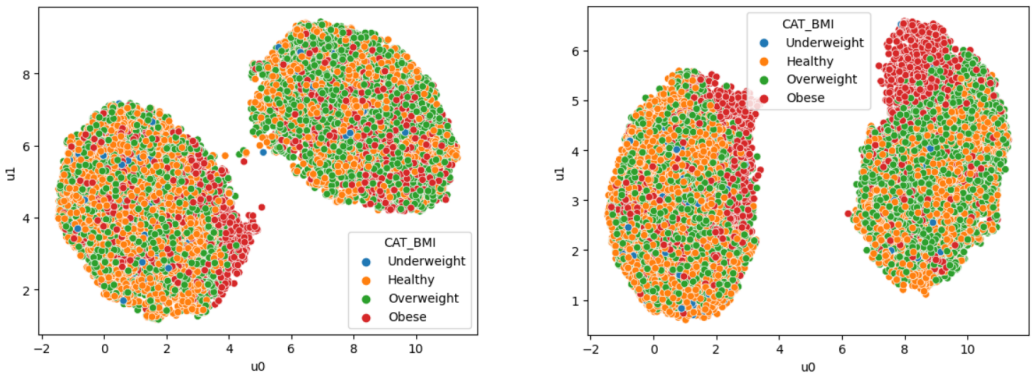


Figure 5.1: Patch embedding PCA: when trained on crops centered on body region but without an explicit attention mask, DINOv2 rely on embeddings generated in the background area

	Global	Local
Rotation	0.3	0
Horizontal flip	0.2	0.2
Vertical flip	0.2	0.2
Blur/Sharpening	0.25	0.2
Solarize/Equalize/Sobel	0.25	0.2
Flip channels	0.1	0.1

Table 5.1: DINOv2 augmentations, the table shows the probability that a transformation is applied. Transformations in the same row are mutually exclusive.

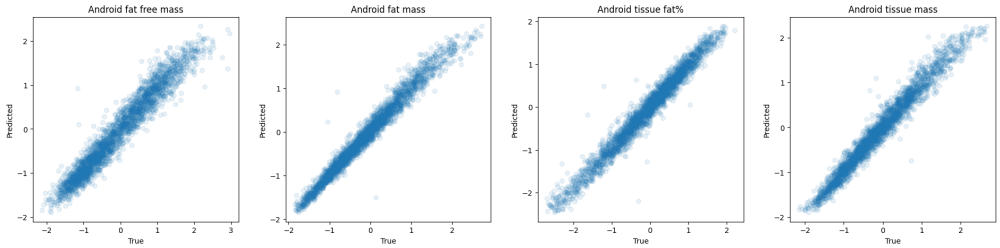
Additional Results: VAE



(a) VAE_{10} BMI (b) VAE_{64} BMI

Figure 5.2: VAEs global PCA umap: using a categorical version of BMI highlight the model's ability to identify the obese subjects. However the other categories are extremely mixed.

Additional Results: DXA Traits

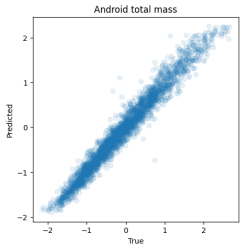


(a) Andr. fat free mass

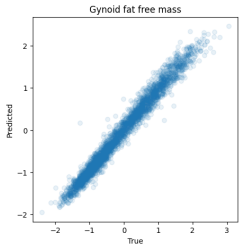
(b) Andr. fat mass

(c) Andr. tissue fat

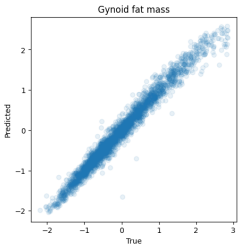
(d) Andr. tissue mass



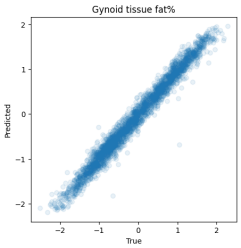
(e) Andr. total mass



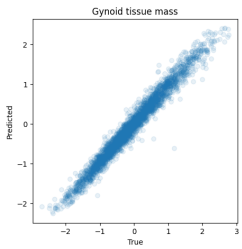
(f) Gynoid fat free mass



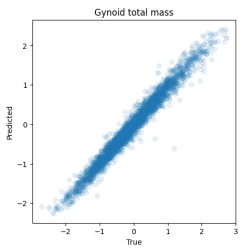
(g) Gynoid fat mass



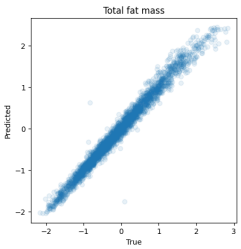
(h) Gynoid tissue fat



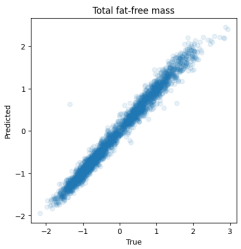
(i) Gynoid tissue mass



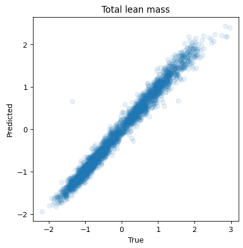
(j) Gynoid total mass



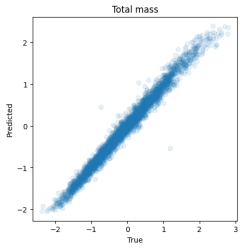
(k) Total fat mass



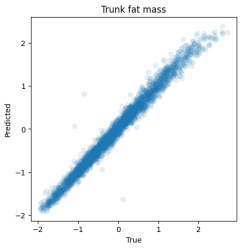
(l) Total fat-free mass



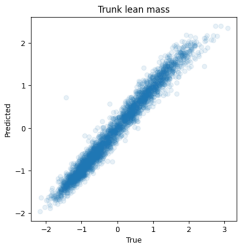
(m) Total lean mass



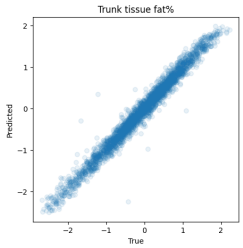
(n) Total mass



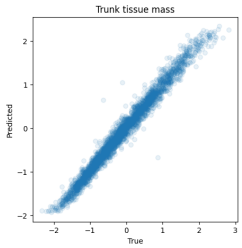
(o) Trunk fat mass



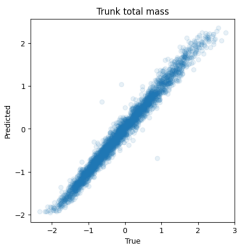
(p) Trunk lean mass



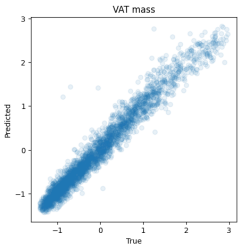
(q) Trunk tissue fat



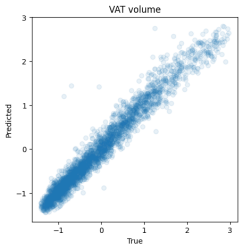
(r) Trunk tissue mass



(s) Trunk total mass

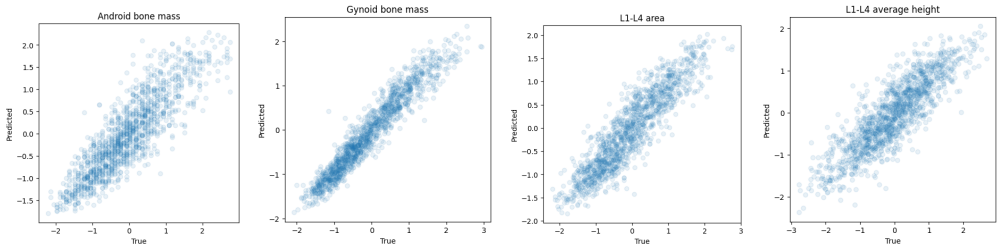


(t) VAT mass

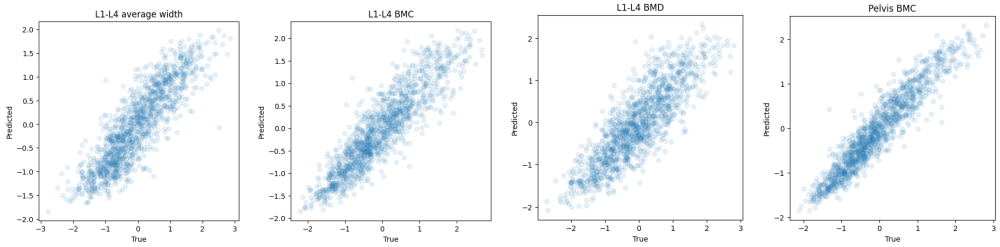


(u) VAT volume

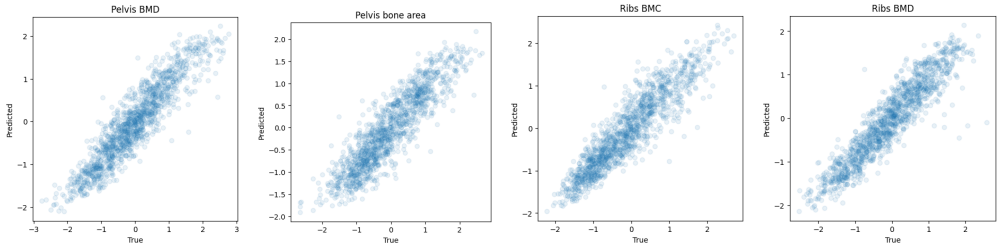
Figure 5.3: DXA fat-related trait's regression results



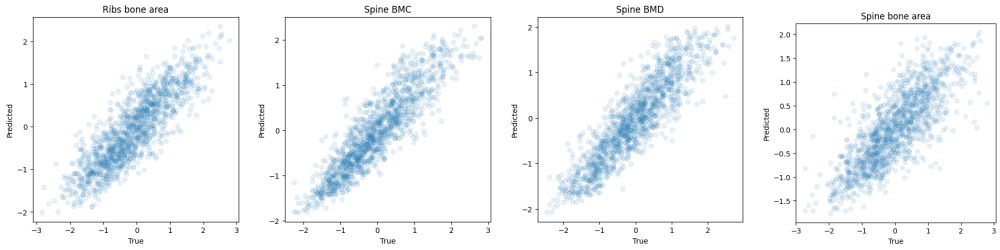
(a) Android bone mass (b) Gynoid bone mass (c) L1-L4 area (d) L1-L4 avg. height



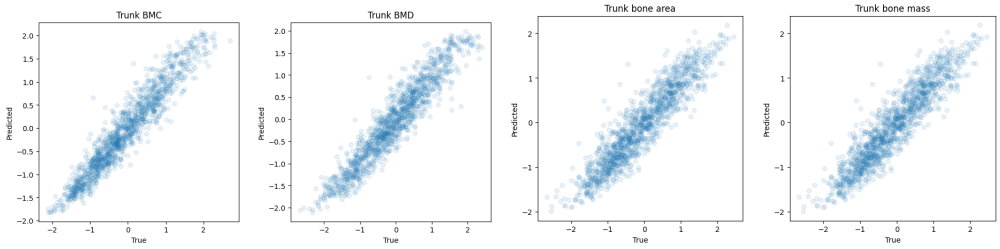
(e) L1-L4 avg. width (f) L1-L4 BMC (g) L1-L4 BMD (h) Pelvis BMC



(i) Pelvis BMD (j) Pelvis bone area (k) Ribs BMC (l) Ribs BMD



(m) Ribs bone area (n) Spine BMC (o) Spine BMD (p) Spine bone area



(q) Trunk BMC (r) Trunk BMD (s) Trunk bone area (t) Trunk bone mass

Figure 5.4: DXA bones-related trait's regression results. BMC(BMD) refer to Bone Mineral Content(Density)

Processed Clinical Data

Table 5.2: General Practitioner event

mapping	all	<10
Pain in joint	7584	5145
Family history of diseases of the circulatory system	6907	3761
Personal history of psychoactive substance abuse	6735	5982
"Radiological examination, not elsewhere classified"	5483	4697
"Acute upper respiratory infection, unspecified"	5248	3392
Pain in limb	4925	3232
Tobacco abuse counselling	4745	4061
Low back pain	4370	2286
"Counselling, unspecified"	3501	2304
"Disorder of skin and subcutaneous tissue, unspecified"	3387	2274
Family history of stroke	3117	1459
"Acute pharyngitis, unspecified"	2949	1417
Impacted cerumen	2774	1935
"Dermatitis, unspecified"	2640	1397
"Urinary tract infection, site not specified"	2569	1460
Family history of diabetes mellitus	2494	1386
Other and unspecified abdominal pain	2445	1298
"Dorsalgia, unspecified"	2421	1048
Cervicalgia	2370	1209
"Chest pain, unspecified"	2204	1104
"Arthrosis, unspecified"	2171	1399
Gastroenteritis and colitis of unspecified origin	2047	1123
Lateral epicondylitis	1995	914
"Asthma, unspecified"	1890	1058
"Haemorrhoids, unspecified"	1863	884
Dependence syndrome	1798	1002
Menopausal and female climacteric states	1743	757
"Depressive episode, unspecified"	1738	721
"Otitis externa, unspecified"	1691	1118
Other specified counselling	1683	1194

Continued on next page

Table 5.2 – continued from previous page

mapping	all	<10
"Hearing loss, unspecified"	1679	976
Sciatica	1672	790
Tinea unguium	1656	955
Irritable bowel syndrome without diarrhoea	1640	582
Other chest pain	1635	832
"Elevated blood-pressure reading, without diagnosis of hypertension"	1632	834
Actinic keratosis	1598	1218
"Acute conjunctivitis, unspecified"	1593	550
"Acute sinusitis, unspecified"	1582	722
Gastro-oesophageal reflux disease without oesophagitis	1581	1115
"Melanocytic naevi, unspecified"	1572	790
Need for immunisation against influenza	1560	1362
"Migraine, unspecified"	1539	568
"Obesity, unspecified"	1518	842
Pain localised to upper abdomen	1482	786
"Chronic sinusitis, unspecified"	1457	940
Excessive and frequent menstruation with regular cycle	1442	415
"Allergic rhinitis, unspecified"	1393	384
Otalgia	1379	735
Pure hypercholesterolaemia	1374	802

Table 5.3: Hospitalization event

mapping	all	<10
Diverticular disease of large intestine without perforation or abscess	2187	1380
"Chest pain, unspecified"	2019	1003
"Unilateral or unspecified inguinal hernia"	1857	1032
"Cataract, unspecified"	1782	1121
Polyp of colon	1477	945
Atherosclerotic heart disease	1431	752
Unknown and unspecified causes of morbidity	1410	285
Special screening examination for neoplasm of intestinal tract	1378	894
Unspecified haematuria	1319	717

Continued on next page

Table 5.3 – continued from previous page

mapping	all	<10
Change in bowel habit	1289	818
Diaphragmatic hernia without obstruction or gangrene	1246	702
Other and unspecified abdominal pain	1244	547
Senile nuclear cataract	1172	657
Skin of other and unspecified parts of face	1133	661
Varicose veins of lower extremities without ulcer or inflammation	1080	383
"Gastritis, unspecified"	1036	655
Haemorrhage of anus and rectum	1026	453
Pain localised to upper abdomen	995	493
Dyspepsia	956	556
"Gonarthrosis, unspecified"	948	516
Carpal tunnel syndrome	923	475
Malignant neoplasm of prostate	917	519
Gastro-oesophageal reflux disease with oesophagitis	908	446
Other chest pain	901	468
"Non-infective gastro-enteritis and colitis, unspecified"	899	464
Sterilisation	873	89
Hyperplasia of prostate	866	515
Gastro-oesophageal reflux disease without oesophagitis	817	450
Unspecified haemorrhoids without complication	790	424
"Breast, unspecified"	726	350
Excessive and frequent menstruation with regular cycle	725	196
Polyp of corpus uteri	720	345
Pain localised to other parts of lower abdomen	709	351
"Urinary tract infection, site not specified"	705	332
Sigmoid colon	702	390
Oesophagitis	701	376
Postmenopausal bleeding	699	274
Leiomyoma of uterus, unspecified	663	233
Calculus of gallbladder without cholecystitis	648	353
"Gastro-intestinal haemorrhage, unspecified"	627	460
Fitting and adjustment of urinary device	598	343
Syncope and collapse	595	308
"Disorder of skin and subcutaneous tissue, unspecified"	593	270

Continued on next page

Table 5.3 – continued from previous page

mapping	all	<10
Gastroenteritis and colitis of unspecified origin	586	402
Headache	578	276
Trichilemmal cyst	577	216
Calculus of gallbladder with other cholecystitis	577	334
Observation for other suspected diseases and conditions	576	325
Atrial fibrillation and flutter	570	391
Dysphagia	564	319

Table 5.4: Self reported disease

mapping	all	<10
hypertension	12617	5051
chickenpox	9870	21
high cholesterol	9171	5193
measles / morbillivirus	7401	2
hayfever/allergic rhinitis	5956	653
asthma	5935	825
unclassifiable	5368	2145
osteoarthritis	5000	2337
mumps / epidemic parotitis	4712	5
gastro-oesophageal reflux (gord) / gastric reflux	4584	2449
depression	4497	1232
tonsiltis	3886	47
cataract	3296	2641
eczema/dermatitis	3080	665
rubella / german measles	2892	1
migraine	2867	321
hypothyroidism/myxoedema	2795	1080
arthritis (nos)	2354	1444
irritable bowel syndrome	2136	604
enlarged prostate	2014	1334
diabetes	1906	880
anxiety/panic attacks	1781	671

Continued on next page

Table 5.4 – continued from previous page

mapping	all	<10
appendicitis	1703	73
hiatus hernia	1689	802
back pain	1629	522
pneumonia	1550	329
allergy or anaphylactic reaction to drug	1468	191
essential hypertension	1365	664
prolapsed disc/slipped disc	1285	267
shingles	1283	499
whooping cough / pertussis	1283	28
sciatica	1234	511
angina	1233	494
osteoporosis	1171	715
uterine fibroids	1154	207
cholelithiasis/gall stones	1117	430
inguinal hernia	1081	467
urinary frequency / incontinence	1070	703
type 2 diabetes	1069	617
diverticular disease/diverticulitis	1066	668
gout	1039	451
heart attack/myocardial infarction	1000	478
psoriasis	985	173
glaucoma	978	599
bronchitis	975	106
tinnitus / tiniitis	947	452
back problem	944	125
varicose veins	930	202
atrial fibrillation	905	612
chronic sinusitis	869	146

Study-wide significant loci identified by GWAS on abdominal traits

Table 5.10: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **spleen volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) ± the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	β ± SE	P value
rs79898419_A_G	1	91587561	G	0.243840	-0.045 ± 0.007	2.26e-10
rs857720_T_C	1	158611797	C	0.266020	-0.059 ± 0.007	2.61e-17
rs35457431_G_A	1	204266749	A	0.272710	0.058 ± 0.007	4.84e-17
rs9787298_A_C	1	204276040	C	0.214290	0.046 ± 0.008	8.97e-10
rs56043070_G_A	1	247719769	A	0.069930	0.091 ± 0.012	4.69e-14
rs4335411_G_A	1	249191706	A	0.762440	0.047 ± 0.007	5.37e-11
rs10185404_A_G	2	12870393	G	0.657730	-0.039 ± 0.006	2.79e-09
rs35330522_G_A	2	12877060	A	0.475850	-0.043 ± 0.006	3.08e-12
rs34352314_A_G	2	68655836	G	0.046750	0.085 ± 0.015	6.49e-09
rs700635_C_A	2	202153225	A	0.733980	-0.05 ± 0.007	9.72e-13
rs13020901_A_G	2	227254695	G	0.343370	-0.045 ± 0.006	2.54e-12
rs11676298_C_G	2	227291731	G	0.190680	-0.077 ± 0.008	5.34e-23
rs574716965_T_C	2	227299531	C	0.059860	-0.078 ± 0.013	2.24e-09
rs454615_T_C	3	12077010	C	0.829990	0.055 ± 0.008	1.70e-11
rs13098031_G_T	3	12239931	T	0.269170	0.063 ± 0.007	1.73e-19
rs13089989_T_A	3	12259089	A	0.239200	0.059 ± 0.007	4.92e-16
rs310747_T_G	3	12259545	G	0.626800	0.043 ± 0.006	1.34e-11
rs12491937_A_G	3	12268244	G	0.418010	0.07 ± 0.006	1.42e-29
rs310752_G_A	3	12272683	A	0.442150	-0.037 ± 0.006	2.11e-09
rs9850825_G_A	3	12286720	A	0.444780	0.044 ± 0.006	2.09e-12
rs9875338_G_A	3	12296469	A	0.398080	0.049 ± 0.006	7.22e-15
rs2920500_G_A	3	12323413	A	0.527100	-0.042 ± 0.006	1.44e-11
rs6442310_T_A	3	12358230	A	0.566460	0.038 ± 0.006	6.31e-10
rs2028759_C_T	3	12418612	T	0.474100	-0.037 ± 0.006	3.12e-09
rs1175540_C_A	3	12465243	A	0.345570	0.04 ± 0.006	5.14e-10
rs2305637_C_T	3	47045846	T	0.177710	0.054 ± 0.008	1.51e-11
rs62248908_A_G	3	47297060	G	0.176270	0.056 ± 0.008	5.24e-12

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs56185965_T_A	3	47580158	A	0.163830	0.049 ± 0.008	4.40e-09
rs62262391_C_T	3	123105119	T	0.222410	0.048 ± 0.007	9.34e-11
rs4680559_C_G	3	159833750	G	0.133740	0.055 ± 0.009	9.49e-10
rs4679874_C_T	3	159930789	T	0.285600	0.045 ± 0.007	3.41e-11
rs55667203_C_T	3	159950798	T	0.171670	0.048 ± 0.008	2.81e-09
rs7612937_G_C	3	196490162	C	0.602830	0.039 ± 0.006	6.11e-10
rs13092376_A_C	3	196516288	C	0.419000	0.053 ± 0.006	1.93e-17
rs7649045_T_C	3	196519878	C	0.593500	0.043 ± 0.006	6.33e-12
rs2237000_T_C	4	2257917	C	0.134060	-0.055 ± 0.009	1.04e-09
rs4444903_A_G	4	110834110	G	0.403480	0.058 ± 0.006	1.97e-20
rs17320827_G_T	4	110948011	T	0.349730	0.044 ± 0.006	6.84e-12
rs13129661_C_G	4	120152909	G	0.324690	0.046 ± 0.007	3.88e-12
rs10026625_T_C	4	120280136	C	0.438070	0.04 ± 0.006	9.73e-11
rs71595380_C_T	4	120329294	T	0.255380	0.042 ± 0.007	2.31e-09
rs569705290_A_C	4	120333521	C	0.163830	0.053 ± 0.008	1.76e-10
rs7692994_C_A	4	120427489	A	0.332930	0.049 ± 0.007	3.41e-14
rs2389875_G_T	4	120557014	T	0.463290	-0.039 ± 0.006	1.76e-10
rs13148635_T_C	4	146172749	C	0.837310	0.054 ± 0.008	9.43e-11
rs7734992_T_C	5	1280128	C	0.415890	0.043 ± 0.006	4.80e-12
rs2736100_C_A	5	1286516	A	0.502140	-0.038 ± 0.006	8.71e-10
rs7710703_T_C	5	1287505	C	0.868090	-0.056 ± 0.009	9.49e-10
rs318481_G_A	6	2888213	A	0.535570	-0.037 ± 0.006	2.78e-09
rs199088_T_C	6	23536807	C	0.520690	0.042 ± 0.006	1.58e-11
rs2517910_A_C	6	29688501	C	0.547250	-0.039 ± 0.006	4.15e-10
rs1611719_G_A	6	29829636	A	0.212920	0.051 ± 0.008	1.17e-11
rs1611719_G_T	6	29829636	T	0.120560	0.058 ± 0.009	7.06e-10
rs1611581_T_C	6	29850827	C	0.680910	-0.042 ± 0.007	1.37e-10
rs2523882_C_T	6	31042217	T	0.242040	-0.044 ± 0.007	6.83e-10
rs2535305_A_G	6	31054381	G	0.393990	-0.046 ± 0.006	2.14e-13
rs35944195_C_T	6	31074829	T	0.100150	0.067 ± 0.01	5.48e-11
rs28749551_G_C	6	31163972	C	0.196880	0.048 ± 0.008	4.08e-10
rs4455663_T_C	6	31193164	C	0.317710	-0.042 ± 0.007	1.68e-10
rs3134783_G_T	6	31197263	T	0.104130	-0.071 ± 0.01	1.45e-12
rs3130533_A_T	6	31208788	T	0.458930	0.042 ± 0.006	7.70e-12

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs2844623_C_T	6	31232543	T	0.187300	-0.05 ± 0.008	3.01e-10
rs2395470_G_A	6	31235261	A	0.325150	-0.05 ± 0.007	2.71e-14
rs1130559_T_G	6	31236652	G	0.228060	-0.046 ± 0.007	3.28e-10
rs1130586_G_A	6	31236680	A	0.275840	-0.048 ± 0.007	4.33e-12
rs1050686_G_A	6	31238909	A	0.152800	0.05 ± 0.009	1.03e-14
rs1050686_G_A	6	31238909	A	0.152800	0.066 ± 0.009	1.03e-14
rs17879195_C_T	6	31239305	T	0.173400	0.056 ± 0.008	4.69e-12
rs2308557_C_T	6	31239417	T	0.346410	-0.038 ± 0.006	4.76e-09
rs2524084_A_G	6	31241639	G	0.362530	0.053 ± 0.006	1.36e-16
rs3132487_G_A	6	31243008	A	0.256210	0.051 ± 0.007	1.80e-23
rs3132487_G_A	6	31243008	A	0.256210	0.07 ± 0.007	1.80e-23
rs3132487_G_T	6	31243008	T	0.350460	0.048 ± 0.006	1.54e-13
rs9264942_T_C	6	31274380	C	0.347410	-0.042 ± 0.006	9.72e-11
rs1634791_A_G	6	31276777	G	0.535660	-0.058 ± 0.006	5.45e-21
rs9265548_T_C	6	31298218	C	0.242890	0.05 ± 0.007	3.18e-12
rs9265620_T_C	6	31299685	C	0.376960	0.04 ± 0.006	2.05e-10
rs9266057_G_A	6	31319176	A	0.144560	0.077 ± 0.009	8.77e-19
rs9266057_G_T	6	31319176	T	0.654190	-0.05 ± 0.006	1.21e-14
rs36057735_C_G	6	31319923	G	0.206630	0.054 ± 0.008	1.25e-12
rs9266092_A_G	6	31320578	G	0.388870	-0.042 ± 0.006	2.66e-11
rs2596501_C_T	6	31321211	T	0.462920	-0.039 ± 0.006	1.92e-10
rs12196740_C_T	6	31321407	T	0.085120	0.066 ± 0.011	1.72e-09
rs41551018_C_A	6	31324051	A	0.052400	0.084 ± 0.014	1.17e-09
rs9266271_T_C	6	31326082	C	0.352080	-0.04 ± 0.006	7.68e-10
rs2523591_G_A	6	31326960	A	0.391840	-0.047 ± 0.006	9.80e-14
rs2844551_A_G	6	31342781	G	0.266700	-0.045 ± 0.007	8.00e-11
rs6929845_T_C	6	31358545	C	0.426250	-0.039 ± 0.006	4.68e-10
rs2523497_T_C	6	31376928	C	0.328870	-0.043 ± 0.007	4.37e-11
rs55957433_A_G	6	31402383	G	0.195140	-0.05 ± 0.008	8.61e-11
rs72844762_G_A	6	31463380	A	0.032290	-0.109 ± 0.017	4.50e-10
rs4713470_T_C	6	31472821	C	0.305800	-0.045 ± 0.007	1.61e-11
rs4713471_A_G	6	31472823	G	0.153490	-0.064 ± 0.009	5.23e-14
rs169738_A_G	6	33537546	G	0.600330	0.039 ± 0.006	6.16e-10
rs76376137_T_G	6	34173330	G	0.049050	0.093 ± 0.014	6.79e-11

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs2797963_C_T	6	34185749	T	0.920250	-0.067 ± 0.011	4.42e-09
rs190927521_C_A	6	34237201	A	0.023830	0.119 ± 0.02	4.02e-09
rs2502405_C_T	6	113446635	T	0.380820	0.047 ± 0.006	9.65e-14
rs113898003_T_C	6	130341235	C	0.262550	-0.043 ± 0.007	8.86e-10
rs7775698_C_T	6	135418635	T	0.263540	0.06 ± 0.007	4.97e-18
rs6934903_T_A	6	135451564	A	0.189560	0.046 ± 0.008	4.18e-09
rs118069386_T_C	6	135603460	C	0.010010	-0.179 ± 0.031	6.70e-09
rs4710707_A_G	6	170481351	G	0.839260	-0.052 ± 0.008	7.11e-10
rs138449099_A_C	6	170519417	C	0.053530	0.08 ± 0.014	5.63e-09
rs9348303_G_C	6	170568621	C	0.187360	-0.046 ± 0.008	7.04e-09
rs4737010_G_A	8	41630447	A	0.227270	-0.049 ± 0.007	3.04e-11
rs6476798_A_G	9	4048636	G	0.736150	0.045 ± 0.007	1.66e-10
rs12005199_G_A	9	4763491	A	0.316390	0.038 ± 0.007	6.90e-09
rs2027939_G_A	9	21956492	A	0.142600	0.065 ± 0.009	1.20e-13
rs3731211_T_A	9	21986847	A	0.720880	-0.079 ± 0.007	8.32e-31
rs3731201_C_T	9	21988896	T	0.865610	-0.073 ± 0.009	6.86e-16
rs61271866_T_A	9	21997015	A	0.287700	-0.04 ± 0.007	3.60e-09
rs115574830_T_A	9	22059061	A	0.088180	0.063 ± 0.011	6.05e-09
rs2065504_G_T	9	22141552	T	0.517080	-0.042 ± 0.006	5.16e-12
rs10811665_T_C	9	22146604	C	0.157610	0.108 ± 0.008	1.06e-37
rs6475612_G_A	9	22152580	A	0.376110	-0.045 ± 0.006	1.26e-12
rs554379017_C_T	9	22154665	T	0.064300	0.115 ± 0.013	6.57e-20
rs10811668_C_A	9	22164991	A	0.205530	0.054 ± 0.008	1.54e-12
rs10965280_A_G	9	22181079	G	0.244740	0.056 ± 0.007	3.21e-15
rs2779746_T_A	9	22193242	A	0.353390	-0.041 ± 0.006	1.16e-10
rs7035278_G_A	9	22203026	A	0.477300	0.04 ± 0.006	8.14e-11
rs62549693_A_G	9	91373903	G	0.017710	0.166 ± 0.023	1.20e-12
rs9410202_T_C	9	91401495	C	0.063970	0.143 ± 0.013	7.69e-30
rs9410204_G_C	9	91401773	C	0.138900	0.067 ± 0.009	7.08e-14
rs9410407_A_T	9	91525989	T	0.051990	0.118 ± 0.014	2.19e-17
rs4743779_C_T	9	107732421	T	0.358520	0.038 ± 0.006	2.47e-09
rs1533968_G_C	9	132287406	C	0.358180	0.039 ± 0.006	1.25e-09
rs12378537_C_T	9	136065526	T	0.078520	0.072 ± 0.011	2.69e-10
rs635634_C_T	9	136155000	T	0.186690	-0.066 ± 0.008	4.20e-17

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs883017_T_C	10	50217438	C	0.565210	-0.041 ± 0.006	2.83e-11
rs2250602_G_A	10	50247251	A	0.591340	-0.041 ± 0.006	9.58e-11
rs61850682_C_T	10	50252258	T	0.234810	0.062 ± 0.007	1.82e-17
rs4240499_T_C	10	50310251	C	0.390150	0.047 ± 0.006	5.13e-14
rs111209284_G_A	10	50313377	A	0.124440	0.056 ± 0.009	1.59e-09
rs7483960_G_A	11	9365378	A	0.679420	-0.038 ± 0.007	6.90e-09
rs744036_C_T	11	100444523	T	0.309010	0.045 ± 0.007	9.43e-12
rs72996119_A_G	11	100457812	G	0.095740	0.092 ± 0.01	1.12e-18
rs3016199_T_A	11	100458965	A	0.415350	0.04 ± 0.006	9.65e-11
rs11224356_G_A	11	100520617	A	0.684270	-0.047 ± 0.007	7.36e-13
rs11224393_A_G	11	100569021	G	0.062820	0.086 ± 0.013	1.44e-11
rs73573308_A_G	11	100571580	G	0.164560	0.054 ± 0.008	6.21e-11
rs8705_G_A	11	128328913	A	0.314880	0.042 ± 0.007	1.96e-10
rs10849446_A_C	12	6479292	C	0.565020	-0.041 ± 0.006	3.39e-11
rs10849447_T_C	12	6484709	C	0.531040	0.044 ± 0.006	1.53e-12
rs4285931_T_C	12	6487161	C	0.310890	0.041 ± 0.007	9.28e-10
rs3759327_G_A	12	6490879	A	0.574570	0.037 ± 0.006	4.35e-09
rs28999107_G_T	12	6493100	T	0.451640	0.048 ± 0.006	1.12e-14
rs10849448_A_G	12	6493351	G	0.753610	0.053 ± 0.007	1.14e-13
rs2238147_C_G	12	111280604	G	0.606880	-0.047 ± 0.006	4.83e-14
rs2339637_C_A	12	111302295	A	0.908100	-0.062 ± 0.011	6.92e-09
rs11065755_G_C	12	111335752	C	0.767890	-0.058 ± 0.007	1.10e-15
rs876312_C_A	12	111366750	A	0.507960	-0.056 ± 0.006	1.02e-19
rs7968960_C_A	12	111426615	A	0.657450	-0.08 ± 0.006	2.83e-35
rs2157876_G_T	12	111592380	T	0.078880	-0.081 ± 0.011	1.69e-12
rs11832104_T_C	12	111619440	C	0.157360	-0.066 ± 0.008	7.07e-15
rs3809293_G_C	12	111639098	C	0.162060	-0.074 ± 0.008	1.15e-18
rs4766559_C_T	12	111666984	T	0.149270	-0.067 ± 0.009	7.22e-15
rs2339718_A_G	12	111702646	G	0.246810	-0.076 ± 0.007	8.08e-27
rs1265564_A_C	12	111708458	C	0.443820	0.122 ± 0.006	5.27e-86
rs2301658_C_T	12	111733274	T	0.168760	-0.078 ± 0.008	1.62e-21
rs7973949_T_A	12	111756402	A	0.154300	-0.058 ± 0.009	1.32e-11
rs150481160_C_T	12	111761895	T	0.033190	-0.118 ± 0.017	7.54e-12
rs112158477_C_T	12	111766446	T	0.146630	-0.062 ± 0.009	1.00e-12

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs3809272_G_A	12	111800258	A	0.302230	-0.096 ± 0.007	9.78e-47
rs117829767_C_T	12	111811474	T	0.033280	0.116 ± 0.017	1.14e-11
rs565050797_T_A	12	111813920	A	0.085060	-0.085 ± 0.011	9.72e-15
rs10774621_G_A	12	111817367	A	0.205170	-0.081 ± 0.008	1.95e-26
rs530225620_T_A	12	111855296	A	0.116660	-0.087 ± 0.01	1.29e-19
rs2239194_C_T	12	111883779	T	0.075810	-0.095 ± 0.012	2.57e-16
rs73201772_C_T	12	111905245	T	0.117520	-0.064 ± 0.01	2.20e-11
rs111673663_A_G	12	111920229	G	0.038250	-0.11 ± 0.016	6.81e-12
rs17805591_G_A	12	111923455	A	0.039280	-0.118 ± 0.016	1.15e-13
rs76604970_A_T	12	111967155	T	0.093910	-0.081 ± 0.011	1.24e-14
rs61944391_A_C	12	112013937	C	0.704490	0.099 ± 0.007	4.23e-49
rs142170016_C_T	12	112027067	T	0.087860	0.087 ± 0.011	9.18e-16
rs112867136_C_T	12	112032145	T	0.019230	0.133 ± 0.022	2.90e-09
rs117532831_G_A	12	112037450	A	0.092400	-0.114 ± 0.011	5.22e-27
rs11613633_C_A	12	112040904	A	0.165330	-0.078 ± 0.008	7.71e-21
rs10744774_C_A	12	112090022	A	0.830390	0.113 ± 0.008	2.58e-43
rs56750693_A_T	12	112144586	T	0.275690	-0.103 ± 0.007	5.22e-51
rs6490291_T_A	12	112177775	A	0.962940	-0.095 ± 0.016	4.69e-09
rs2013002_T_C	12	112200150	C	0.587240	-0.136 ± 0.006	3.10e-105
rs75480510_G_A	12	112240192	A	0.094080	-0.093 ± 0.011	1.58e-18
rs111481433_A_G	12	112252052	G	0.037500	-0.118 ± 0.016	3.10e-13
rs35450384_A_G	12	112259837	G	0.119460	0.088 ± 0.009	1.09e-20
rs111752063_G_T	12	112297450	T	0.172180	-0.08 ± 0.008	1.31e-22
rs11066091_A_G	12	112367364	G	0.117520	-0.098 ± 0.01	1.60e-24
rs79423041_T_C	12	112413025	C	0.047400	-0.099 ± 0.014	6.90e-12
rs7135514_T_C	12	112448485	C	0.867480	0.106 ± 0.009	2.54e-31
rs117102609_T_C	12	112495256	C	0.057910	0.119 ± 0.013	1.65e-19
rs537498778_T_C	12	112511091	C	0.046890	0.113 ± 0.015	8.95e-15
rs11066152_A_C	12	112519764	C	0.119030	0.089 ± 0.01	5.80e-21
rs112693912_T_C	12	112536600	C	0.031280	0.124 ± 0.018	2.64e-12
rs113473789_A_G	12	112547337	G	0.037750	-0.119 ± 0.016	1.87e-13
rs7965087_G_A	12	112548562	A	0.094870	-0.092 ± 0.01	1.58e-18
rs558464174_T_A	12	112573648	A	0.085310	0.067 ± 0.011	1.01e-09
rs17630235_G_A	12	112591686	A	0.407730	0.132 ± 0.006	3.53e-99

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs77215829_A_C	12	112618346	C	0.131210	-0.09 ± 0.009	7.03e-23
rs11610922_G_A	12	112708540	A	0.060790	-0.104 ± 0.013	7.24e-16
rs547587127_C_T	12	112761777	T	0.393980	-0.099 ± 0.006	2.06e-55
rs28551113_A_G	12	112814745	G	0.094970	-0.09 ± 0.01	6.99e-18
rs7398263_A_T	12	112828643	T	0.243650	-0.053 ± 0.007	2.15e-13
rs11066309_G_A	12	112883476	A	0.408110	0.13 ± 0.006	6.18e-96
rs188838656_C_T	12	112886935	T	0.026200	-0.117 ± 0.019	1.10e-09
rs11614544_A_G	12	112904955	G	0.140130	-0.104 ± 0.009	1.08e-31
rs113499824_T_G	12	112917667	G	0.037990	-0.115 ± 0.016	1.02e-12
rs117179247_G_A	12	112942856	A	0.031160	0.12 ± 0.018	1.22e-11
rs10774653_A_T	12	112954926	T	0.071650	-0.081 ± 0.012	9.47e-12
rs7138688_G_T	12	112978204	T	0.879960	-0.09 ± 0.009	1.24e-21
rs11066353_G_A	12	113012293	A	0.158410	-0.064 ± 0.008	2.49e-14
rs233724_G_A	12	113030227	A	0.516180	-0.071 ± 0.006	3.96e-31
rs117529015_G_A	12	113030631	A	0.027490	0.119 ± 0.019	2.18e-10
rs55847700_C_A	12	113045801	A	0.423160	-0.06 ± 0.006	7.76e-22
rs233709_A_T	12	113058814	T	0.336310	-0.04 ± 0.007	5.96e-10
rs10492017_G_A	12	113075294	A	0.216640	-0.049 ± 0.007	6.30e-11
rs1013556_T_A	12	113097291	A	0.171500	-0.051 ± 0.008	3.99e-10
rs232924_T_C	12	113104903	C	0.429880	0.059 ± 0.006	1.65e-21
rs232935_T_C	12	113115628	C	0.566880	0.043 ± 0.006	4.20e-12
rs4767948_G_A	12	113116000	A	0.269810	-0.053 ± 0.007	3.23e-14
rs755340265_A_G	12	113116128	G	0.135280	0.07 ± 0.009	4.66e-15
rs7298438_G_C	12	113132490	C	0.325480	-0.051 ± 0.007	1.00e-14
rs2384049_A_G	12	113135531	G	0.715490	-0.092 ± 0.007	4.51e-41
rs556180462_T_C	12	113150742	C	0.028300	0.129 ± 0.019	3.86e-12
rs34560882_C_A	12	113164710	A	0.320670	-0.042 ± 0.007	1.27e-10
rs6489855_G_A	12	113164767	A	0.554540	-0.052 ± 0.006	3.67e-17
rs7313341_A_T	12	113167134	T	0.563350	-0.071 ± 0.006	2.70e-30
rs7132088_C_T	12	113235279	T	0.442400	-0.038 ± 0.006	7.86e-10
rs7297436_T_G	12	113263569	G	0.341100	0.042 ± 0.006	6.01e-11
rs56959712_G_T	12	123188475	T	0.213840	-0.05 ± 0.008	3.79e-11
rs10846663_C_T	12	124860830	T	0.754800	-0.044 ± 0.007	8.03e-10
rs12367444_G_A	12	124861864	A	0.560040	-0.04 ± 0.006	1.17e-10

Continued on next page

Table 5.10 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs61963226_T_A	13	41535409	A	0.194210	0.046 ± 0.008	3.48e-09
rs9520836_A_G	13	108957063	G	0.231210	0.043 ± 0.007	2.62e-09
rs374039502_T_A	13	108960385	A	0.021400	0.182 ± 0.021	9.13e-18
rs12874404_A_G	13	108993494	G	0.053520	0.09 ± 0.014	4.83e-11
rs66800452_T_C	14	93505385	C	0.110390	-0.06 ± 0.01	9.13e-10
rs8032320_C_G	15	39305379	G	0.385260	0.041 ± 0.006	1.22e-10
rs12913015_C_T	15	39305443	T	0.433700	-0.047 ± 0.006	3.48e-14
rs1157997_C_T	15	39575166	T	0.552070	-0.036 ± 0.006	3.97e-09
rs62002309_T_C	15	39611101	C	0.409070	-0.038 ± 0.006	8.88e-10
rs112642422_C_T	16	4306785	T	0.306120	0.039 ± 0.007	3.29e-09
rs4455005_G_A	17	1958536	A	0.637990	0.051 ± 0.006	1.21e-15
rs187187134_G_A	17	2135194	A	0.016170	-0.143 ± 0.024	4.98e-09
rs7213347_G_C	17	2153257	C	0.700410	0.051 ± 0.007	3.46e-14
rs11658587_C_T	17	4901982	T	0.037200	0.131 ± 0.016	9.69e-16
rs12325805_T_G	17	6912219	G	0.074880	-0.08 ± 0.012	8.54e-12
rs1050541_T_G	17	7560835	G	0.551510	0.037 ± 0.006	2.43e-09
rs146207734_A_T	17	16472850	T	0.016700	0.164 ± 0.024	9.06e-12
rs572423904_G_A	17	16686286	A	0.011850	0.228 ± 0.028	9.58e-16
rs561203872_C_T	17	16749391	T	0.026310	0.118 ± 0.019	9.62e-10
rs2060779_G_A	17	57506092	A	0.222020	0.045 ± 0.007	1.41e-09
rs8073039_C_A	17	57521208	A	0.366300	0.039 ± 0.006	1.50e-09
rs2645479_A_G	17	57920532	G	0.443300	0.043 ± 0.006	3.36e-12
rs11086054_A_T	19	17246737	T	0.297990	0.043 ± 0.007	1.45e-10
rs59616136_G_A	19	17252041	A	0.091660	0.083 ± 0.011	9.77e-15
rs35365035_T_C	19	17252151	C	0.388720	-0.051 ± 0.006	5.97e-16
rs59360945_G_A	19	17256032	A	0.203450	-0.056 ± 0.008	2.62e-13
rs6055955_C_T	20	8604181	T	0.507630	0.039 ± 0.006	1.57e-10
rs6579239_C_T	20	30223509	T	0.157090	-0.053 ± 0.008	3.20e-10
rs34753522_T_C	20	39278391	C	0.353760	-0.045 ± 0.006	3.16e-12

Table 5.5: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **subcutaneous fat volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) ± the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs2802773_A_C	1	203526353	C	0.304070	0.037 ± 0.006	2.00e-09
rs2785998_C_A	1	219734998	A	0.315950	0.036 ± 0.006	3.82e-09
rs10756360_T_G	9	12288829	G	0.475300	-0.036 ± 0.006	3.40e-10
rs2772572_C_T	9	24528103	T	0.286130	-0.042 ± 0.006	2.70e-11
rs2772572_C_A	9	24528103	A	0.042260	-0.089 ± 0.014	3.96e-10
rs10828250_C_G	10	21847178	G	0.304270	0.041 ± 0.006	3.79e-11
rs806774_A_T	10	22114427	T	0.720760	-0.037 ± 0.006	5.34e-09
rs2353487_C_T	11	27690399	T	0.664630	-0.036 ± 0.006	1.86e-09
rs62048402_G_A	16	53803223	A	0.399710	0.065 ± 0.006	5.12e-29
rs2042031_G_A	16	53823718	A	0.613720	0.046 ± 0.006	2.31e-15
rs7187961_T_C	16	53826034	C	0.786460	0.045 ± 0.007	1.31e-10

Table 5.11: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **liver volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) ± the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs193084249_A_G	1	26987646	G	0.023740	0.126 ± 0.02	1.50e-10
rs182050989_C_T	1	27262545	T	0.028320	0.113 ± 0.018	3.36e-10
rs34730887_G_A	2	27379371	A	0.500060	0.036 ± 0.006	2.39e-09
rs74860790_G_A	2	27388447	A	0.056190	-0.081 ± 0.013	5.01e-10
rs1659689_A_G	2	27393030	G	0.696960	0.042 ± 0.007	7.15e-11
rs9678013_A_G	2	27582035	G	0.447220	-0.051 ± 0.006	2.84e-17
rs67086575_A_G	2	27686480	G	0.172660	0.066 ± 0.008	4.52e-17
rs1260326_T_C	2	27730940	C	0.608560	-0.083 ± 0.006	2.38e-41
rs3817588_T_C	2	27731212	C	0.195400	-0.05 ± 0.008	4.86e-11
rs4665985_A_C	2	27753878	C	0.270290	0.074 ± 0.007	7.69e-28
rs202049562_T_A	2	27779565	A	0.176670	-0.048 ± 0.008	9.29e-10
rs2141371_G_A	2	27860258	A	0.695380	0.044 ± 0.006	9.03e-12
rs4616435_C_T	2	27933642	T	0.202420	0.065 ± 0.007	2.79e-18

Continued on next page

Table 5.11 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs17050272_G_A	2	121306440	A	0.404630	-0.04 ± 0.006	5.58e-11
rs7650472_A_G	3	112483087	G	0.034730	0.095 ± 0.016	5.44e-09
rs17600346_T_C	3	172223982	C	0.040240	0.111 ± 0.015	3.70e-13
rs113565595_G_A	3	172253647	A	0.047250	0.096 ± 0.014	8.05e-12
rs13101221_G_T	3	172262865	T	0.116500	0.059 ± 0.009	2.51e-10
rs532782761_A_C	3	172276502	C	0.014280	0.181 ± 0.025	7.63e-13
rs79287178_G_A	3	172294500	A	0.030820	0.147 ± 0.017	2.61e-17
rs144100226_C_T	6	34180297	T	0.044980	0.085 ± 0.014	3.28e-09
rs1490384_C_T	6	126851160	T	0.496960	0.057 ± 0.006	2.64e-21
rs9388498_G_T	6	126873423	T	0.185330	0.046 ± 0.008	3.10e-09
rs853974_T_C	6	127068983	C	0.737810	-0.046 ± 0.007	1.83e-11
rs9491675_G_A	6	127251021	A	0.141640	0.055 ± 0.009	1.56e-10
rs118146420_A_G	8	9157497	G	0.039920	0.09 ± 0.015	4.03e-09
rs34648177_G_A	8	9161125	A	0.036960	0.096 ± 0.016	1.45e-09
rs370720_C_T	8	9168146	T	0.064060	0.101 ± 0.012	1.53e-16
rs7012814_G_A	8	9173358	A	0.472810	-0.066 ± 0.006	3.26e-28
rs12546815_G_A	8	9184099	A	0.244590	-0.044 ± 0.007	2.68e-10
rs4240624_G_A	8	9184231	A	0.906930	-0.16 ± 0.01	1.20e-54
rs34290760_C_G	8	9185179	G	0.031220	0.166 ± 0.017	5.54e-22
rs28494016_A_C	8	9187961	C	0.793100	-0.08 ± 0.007	4.48e-27
rs732839_G_A	8	9194172	A	0.227530	-0.052 ± 0.007	4.24e-13
rs10102352_A_G	8	9206069	G	0.602100	-0.062 ± 0.006	5.87e-24
rs10095103_T_C	8	9226095	C	0.567290	0.041 ± 0.006	8.53e-12
rs12546198_A_G	8	9231661	G	0.101930	0.092 ± 0.01	1.89e-20
rs78913998_A_C	8	9328024	C	0.053010	-0.078 ± 0.013	4.79e-09
rs11249925_A_G	8	9343978	G	0.357580	-0.037 ± 0.006	2.13e-09
rs111622501_C_T	8	9358850	T	0.026870	0.111 ± 0.019	1.85e-09
rs1350559_C_G	8	9367743	G	0.405620	-0.037 ± 0.006	1.98e-09
rs2034983_A_G	8	9372329	G	0.080680	-0.064 ± 0.011	4.85e-09
rs10822120_T_C	10	64829314	C	0.399530	-0.04 ± 0.006	8.00e-11
rs4363488_T_G	10	64862314	G	0.794590	-0.046 ± 0.007	4.58e-10
rs116510698_T_A	10	64916031	A	0.129980	-0.055 ± 0.009	5.56e-10
rs10733789_T_C	10	64948684	C	0.313020	-0.049 ± 0.006	4.90e-14
rs10465990_G_A	10	64960536	A	0.786260	-0.059 ± 0.007	5.38e-16

Continued on next page

Table 5.11 – continued from previous page

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs12358789_A_G	10	65121485	G	0.439540	-0.06 ± 0.006	2.07e-23
rs2893923_C_T	10	65261184	T	0.313230	-0.048 ± 0.006	7.63e-14
rs117212080_T_C	10	65275214	C	0.053050	0.078 ± 0.013	5.55e-09
rs7358191_A_C	10	65278788	C	0.816210	-0.054 ± 0.008	2.21e-12
rs3847327_C_T	10	65382328	T	0.534920	-0.042 ± 0.006	1.74e-12
rs1044563_C_T	10	93390149	T	0.062000	0.074 ± 0.012	2.50e-09
rs61876148_T_C	10	93455145	C	0.146420	0.051 ± 0.008	2.36e-09
rs11186719_A_C	10	93631956	C	0.478450	0.048 ± 0.006	6.38e-16
rs5786975_A_T	10	93642880	T	0.705150	-0.046 ± 0.007	1.85e-12
rs11186933_G_C	10	94108693	C	0.035830	0.099 ± 0.016	6.75e-10
rs4944312_G_A	11	81316330	A	0.662530	-0.037 ± 0.006	7.06e-09
rs67981690_A_G	12	21343886	G	0.129530	0.053 ± 0.009	3.62e-09
rs265679_A_C	13	83689360	C	0.144080	-0.052 ± 0.009	1.01e-09
rs28929474_C_T	14	94844947	T	0.020930	0.132 ± 0.021	2.79e-10
rs148470008_A_G	15	42664157	G	0.019420	0.165 ± 0.022	2.43e-14
rs143875230_G_A	15	43278726	A	0.025980	0.181 ± 0.019	8.10e-22
rs45617141_T_G	15	43623397	G	0.016610	0.205 ± 0.023	2.20e-18
rs150844304_A_C	15	43726625	C	0.026060	0.203 ± 0.019	3.27e-27
rs139974673_T_C	15	44027885	C	0.026490	0.206 ± 0.019	1.96e-28
rs117519814_G_A	15	44052248	A	0.066410	0.076 ± 0.012	2.52e-10
rs11853395_A_C	15	44234776	C	0.043330	0.112 ± 0.015	2.75e-14
rs148596463_G_C	15	44305780	C	0.026050	0.175 ± 0.019	1.21e-20
rs2615299_G_A	15	44492937	A	0.059420	0.078 ± 0.013	8.96e-10
rs144972973_A_G	15	44564692	G	0.026370	0.147 ± 0.019	3.57e-15
rs151291132_A_G	15	44842210	G	0.026280	0.142 ± 0.019	3.53e-14
rs56094641_A_G	16	53806453	G	0.400580	0.041 ± 0.006	2.38e-11
rs58542926_C_T	19	19379549	T	0.075370	0.09 ± 0.011	1.54e-15
rs12608729_C_T	19	19700552	T	0.085140	0.079 ± 0.011	1.73e-13
rs429358_T_C	19	45411941	C	0.148710	-0.055 ± 0.008	8.83e-11

Table 5.6: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **visceral fat volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) ± the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs848522_T_G	2	36750822	G	0.433130	-0.035 ± 0.006	1.20e-09
rs11715304_G_A	3	49861282	A	0.318770	0.036 ± 0.006	5.46e-09
rs17451107_T_C	3	156797609	C	0.391720	0.037 ± 0.006	4.29e-10
rs256904_A_T	5	55810305	T	0.747980	-0.047 ± 0.007	9.29e-13
rs577721086_T_C	6	127440047	C	0.050410	0.086 ± 0.013	5.85e-11
rs11985475_C_T	8	25464656	T	0.414810	0.036 ± 0.006	5.21e-10
rs73221948_G_T	8	25464670	T	0.294400	-0.051 ± 0.006	4.05e-16
rs11992444_G_T	8	25464690	T	0.504450	0.045 ± 0.006	8.97e-15
rs10756360_T_G	9	12288829	G	0.475300	-0.036 ± 0.006	5.24e-10
rs17669477_C_A	11	37954195	A	0.195510	-0.044 ± 0.007	1.70e-09
rs9537128_C_T	13	56059237	T	0.052110	0.086 ± 0.013	2.53e-11
rs9537128_C_A	13	56059237	A	0.279480	0.037 ± 0.006	1.17e-09
rs9537128_C_A	13	56059237	A	0.279480	-0.039 ± 0.006	1.17e-09
rs4808766_G_C	19	18335715	C	0.264740	0.039 ± 0.007	1.53e-09
rs1469024_C_T	19	18384544	T	0.635680	-0.037 ± 0.006	7.15e-10
rs8101436_G_A	19	33787041	A	0.136720	-0.052 ± 0.008	6.72e-10
rs10403561_A_G	19	33790502	G	0.427380	-0.046 ± 0.006	1.97e-15
rs73026242_A_G	19	33834096	G	0.073070	-0.086 ± 0.011	8.77e-15
rs1061338_A_G	19	33878198	G	0.262380	0.04 ± 0.007	7.32e-10
rs62102718_A_T	19	33891013	T	0.282640	-0.037 ± 0.006	6.25e-09
rs3786901_A_C	19	33897478	C	0.426340	0.045 ± 0.006	1.46e-14
rs4239576_A_G	19	33915360	G	0.312740	0.038 ± 0.006	7.23e-10
rs153701_T_C	19	34003255	C	0.540780	-0.043 ± 0.006	1.63e-13
rs41119_G_T	19	34015904	T	0.348060	-0.037 ± 0.006	1.21e-09
rs429358_T_C	19	45411941	C	0.148710	-0.049 ± 0.008	1.69e-09

Table 5.7: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **lung volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) $\pm$ the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs6691985_C_T	1	17289911	T	0.294980	-0.039 ± 0.006	1.60e-09
rs9435731_C_A	1	17306029	A	0.520080	0.046 ± 0.006	2.58e-15
rs35108553_T_C	1	17404802	C	0.573200	0.037 ± 0.006	3.24e-10
rs6604085_C_T	1	92696189	T	0.805880	0.043 ± 0.007	5.11e-09
rs4970698_T_C	1	92972933	C	0.798390	0.044 ± 0.007	1.66e-09
rs34962978_T_C	1	93290643	C	0.098790	-0.057 ± 0.01	6.15e-09
rs1040457_C_T	1	176802119	T	0.503470	0.034 ± 0.006	6.09e-09
rs11125803_C_T	2	25052177	T	0.739610	0.042 ± 0.007	2.96e-10
rs1470039_G_C	2	25193469	C	0.509450	0.037 ± 0.006	1.49e-10
rs3791679_A_G	2	56096892	G	0.225950	-0.045 ± 0.007	1.14e-10
rs2041740_A_G	2	102989734	G	0.774670	0.041 ± 0.007	5.71e-09
rs13077048_A_T	3	141106954	T	0.433970	0.038 ± 0.006	1.52e-10
rs2194411_G_A	3	185548663	A	0.127710	0.061 ± 0.009	2.20e-12
rs2320299_G_A	4	17972372	A	0.733320	-0.047 ± 0.007	1.83e-12
rs9389985_A_G	6	142653898	G	0.286820	-0.062 ± 0.006	1.14e-21
rs171891_G_A	6	142850612	A	0.175050	-0.056 ± 0.008	4.72e-13
rs166399_G_A	6	142924286	A	0.348340	-0.037 ± 0.006	2.42e-09
rs28679544_G_A	7	92215430	A	0.167250	0.051 ± 0.008	7.01e-11
rs28679544_G_T	7	92215430	T	0.019740	0.123 ± 0.021	4.26e-09
rs10756360_T_G	9	12288829	G	0.475300	0.036 ± 0.006	4.60e-10
rs10756360_T_C	9	12288829	C	0.023600	0.115 ± 0.019	2.42e-09
rs28427480_A_C	9	98264942	C	0.092730	0.076 ± 0.01	5.61e-14
rs564819152_A_G	10	21820650	G	0.316430	-0.043 ± 0.006	1.06e-11
rs664317_A_C	10	89812230	C	0.838060	-0.048 ± 0.008	1.04e-12
rs664317_A_C	10	89812230	C	0.838060	-0.057 ± 0.008	1.04e-12
rs7075615_T_C	10	125002717	C	0.827560	0.046 ± 0.008	2.25e-09
rs76895963_T_G	12	4384844	G	0.020940	0.139 ± 0.02	8.93e-12
rs11058282_A_G	12	122951100	G	0.748770	0.044 ± 0.007	4.38e-11
rs12820906_A_G	12	123493123	G	0.248900	0.043 ± 0.007	2.50e-10
rs12824685_G_T	12	123817569	T	0.204490	0.043 ± 0.007	2.00e-09
rs3118904_G_A	13	51100097	A	0.349880	-0.036 ± 0.006	6.50e-09
rs3116616_C_G	13	51162913	G	0.213350	-0.046 ± 0.007	7.90e-11
rs147110934_G_T	19	55993436	T	0.023530	-0.118 ± 0.019	8.24e-10

Table 5.8: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **left kidney volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) ± the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	β ± SE	P value
rs807624_G_T	2	15782471	T	0.357970	0.044 ± 0.007	2.94e-10
rs7578604_G_T	2	121308660	T	0.257000	0.049 ± 0.008	1.29e-10
rs4684847_C_T	3	12386337	T	0.118790	-0.061 ± 0.01	3.62e-09
rs2870237_G_T	4	77367580	T	0.441600	-0.04 ± 0.007	2.27e-09
rs881858_G_A	6	43806609	A	0.694670	-0.043 ± 0.007	3.97e-09
rs12202204_A_G	6	126686407	G	0.316290	-0.044 ± 0.007	1.18e-09
rs1490384_C_T	6	126851160	T	0.496960	0.045 ± 0.007	1.37e-11
rs13220518_T_A	6	127048609	A	0.244680	-0.05 ± 0.008	1.68e-10
rs2077111_A_G	6	127135075	G	0.462370	0.047 ± 0.007	2.65e-12
rs7813900_A_G	8	23724762	G	0.581380	-0.04 ± 0.007	5.60e-09
rs56005989_T_C	9	134601458	C	0.261260	0.047 ± 0.008	8.66e-10
rs7956634_T_C	12	15321194	C	0.203410	0.05 ± 0.008	2.51e-09
rs76900422_G_A	12	66166141	A	0.051790	-0.088 ± 0.015	4.70e-09
rs1042725_C_T	12	66358347	T	0.489000	-0.045 ± 0.007	1.30e-11
rs80269832_A_T	16	20354473	T	0.072660	0.082 ± 0.013	2.36e-10
rs7203451_G_A	16	20358376	A	0.295860	0.046 ± 0.007	2.68e-10
rs13329952_T_C	16	20366507	C	0.194900	0.068 ± 0.008	1.11e-15
rs77924615_G_A	16	20392332	A	0.198080	0.078 ± 0.008	1.68e-20
rs12446492_T_A	16	20408377	A	0.424940	0.051 ± 0.007	5.90e-14
rs9929296_A_C	16	20411380	C	0.352270	0.057 ± 0.007	6.43e-16
rs9895661_C_T	17	59456589	T	0.830790	0.052 ± 0.009	4.65e-09

Table 5.9: Lead SNPs corresponding to the study-wide significant ($P < 7 \times 10^{-9}$) loci identified by POP-GWAS for **right kidney volume**. Included within the table are the Effect Allele (EA) Frequency (EAFreq), Effect Size (β) $\pm$ the Standard Error (SE), and P value.

SNP	CHR	BP	EA	EAFreq	$\beta \pm$ SE	P value
rs807624_G_T	2	15782471	T	0.357970	0.05 ± 0.007	7.71e-13
rs10035253_G_T	5	42215812	T	0.681500	0.042 ± 0.007	6.80e-09
rs744103_T_A	6	43805362	A	0.686430	-0.05 ± 0.007	5.39e-12
rs2077111_A_G	6	127135075	G	0.462370	0.045 ± 0.007	2.32e-11
rs2077111_A_C	6	127135075	C	0.059150	-0.09 ± 0.014	1.93e-10
rs9650405_A_G	8	23772530	G	0.591350	-0.041 ± 0.007	2.48e-09
rs17786744_A_G	8	23777006	G	0.412670	-0.052 ± 0.007	2.57e-14
rs12917707_G_T	16	20367690	T	0.184600	0.066 ± 0.009	2.29e-14
rs77924615_G_A	16	20392332	A	0.198080	0.062 ± 0.008	1.21e-13
rs4968532_C_A	17	59243878	A	0.813510	-0.056 ± 0.009	4.90e-11
rs9895661_C_T	17	59456589	T	0.830790	0.062 ± 0.009	5.12e-12
rs201268638_A_G	17	59486438	G	0.038690	-0.105 ± 0.017	1.47e-09

Bibliography

[1] Abdel Abdellaoui, Loic Yengo, Karin JH Verweij, and Peter M Visscher. 15 years of gwas discovery: realizing the promise. *The American Journal of Human Genetics*, 110(2):179–194, 2023.

[2] Jay Alammar. The illustrated transformer, april 2024.

[3] Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, (3):299–307, 1967.

[4] Eugenia Anello. Convolutional autoencoder in pytorch on mnist dataset, april. [Online; posted Jun 28, 2021].

[5] Aqeel Anwar. Difference between autoencoder (ae) and variational autoencoder (vae), april 2024. [Online; posted Nov 3, 2021].

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[7] Evan A Boyle, Yang I Li, and Jonathan K Pritchard. An expanded view of complex traits: from polygenic to omnigenic. *Cell*, 169(7):1177–1186, 2017.

[8] Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, et al. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, 2018.

[9] Pablo Caceres. Introduction to neural network models of cognition, april 2024.

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2021.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision trans-

formers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[12] Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience*, 4(1):s13742–015, 2015.

[13] Xiaona Chang, Jianchao Wang, Guanjun Zhang, Ming Yang, Yanfeng Xi, Chenghang Xi, Gang Chen, Xiu Nie, Bin Meng, and Xueping Quan. Predicting colorectal cancer microsatellite instability with a self-attention-enabled convolutional neural network. *Cell Reports Medicine*, 4(2), 2023.

[14] Soumick Chatterjee. *Reducing Artefacts in MRI Using Deep Learning: Enhancing Automatic Image Processing Pipelines*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, 2022.

[15] Chengkuan Chen, Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Andrew J Schaumberg, and Faisal Mahmood. Fast and scalable search of whole-slide images via self-supervised deep learning. *Nature biomedical engineering*, 6(12):1420–1434, 2022.

[16] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, pages 1–13, 2024.

[17] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[18] Francesco Cisternino, Sara Ometto, Soumick Chatterjee, Edoardo Giacopuzzi, Adam P Levine, and Craig A Glastonbury. Self-supervised learning for characterising histomorphological diversity and spatial rna expression prediction across 23 human tissue types. *bioRxiv*, pages 2023–08, 2023.

[19] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023.

[20] W T Dixon. Simple proton spectroscopic imaging. *Radiology*, 153(1):189–194, 1984. PMID: 6089263.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold,

Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[22] Lloyd T Elliott, Kevin Sharp, Fidel Alfaro-Almagro, Sinan Shi, Karla L Miller, Gwenaëlle Douaud, Jonathan Marchini, and Stephen M Smith. Genome-wide association studies of brain imaging phenotypes in uk biobank. *Nature*, 562(7726):210–216, 2018.

[23] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Shun-ichi Amari and Michael A. Arbib, editors, *Competition and Cooperation in Neural Nets*, pages 267–285, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. pages 15979–15988, 2022.

[25] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. volume 3, 2017.

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[28] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[29] Fabian Hörst, Moritz Rempe, Lukas Heine, Constantin Seibold, Julius Keyl, Giulia Baldini, Selma Ugurel, Jens Siveke, Barbara Grünwald, Jan Egger, et al. Cellvit: Vision transformers for precise cell segmentation and classification. *Medical Image Analysis*, page 103143, 2024.

[30] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[31] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pages 2649–2658. PMLR, 2018.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[33] Andrew S Lan, Andrew E Waters, Christoph Studer, and Richard G Baraniuk. Sparse

factor analysis for learning and content analytics. *The Journal of Machine Learning Research*, 15(1):1959–2008, 2014.

[34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 12 1989.

[35] Yann LeCun. Self-supervised learning: The dark matter of intelligence, april. [Online; posted March 4, 2021].

[36] Thomas J. Littlejohns, Jo Holliday, Lorna M. Gibson, Steve Garratt, Niels Oesingmann, Fidel Alfaro-Almagro, Jimmy D. Bell, Chris Boultwood, Rory Collins, Megan C. Conroy, Nicola Crabtree, Nicola Doherty, Alejandro F. Frangi, Nicholas C. Harvey, Paul Leeson, Karla L. Miller, Stefan Neubauer, Steffen E. Petersen, Jonathan Sellors, Simon Sheard, Stephen M. Smith, Cathie L. M. Sudlow, Paul M. Matthews, and Naomi E. Allen. The uk biobank imaging enhancement of 100,000 participants: rationale, data collection, management and future directions. *Nature Communications*, 11(1):2624, May 2020.

[37] Yi Liu, Nicolas Basty, Brandon Whitcher, Jimmy D Bell, Elena P Sorokin, Nick van Bruggen, E Louise Thomas, and Madeleine Cule. Genetic architecture of 11 organ traits derived from abdominal mri using deep learning. *Elife*, 10:e65554, 2021.

[38] Yi Liu, Nicolas Basty, Brandon Whitcher, Jimmy D Bell, Elena P Sorokin, Nick van Bruggen, E Louise Thomas, and Madeleine Cule. Genetic architecture of 11 organ traits derived from abdominal mri using deep learning. *eLife*, 10:e65554, jun 2021.

[39] Fergal J Martin, M Ridwan Amode, Alisha Aneja, Olanrewaju Austine-Orimoloye, Andrey G Azov, If Barnes, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, et al. Ensembl 2023. *Nucleic acids research*, 51(D1):D933–D941, 2023.

[40] Joelle Mbatchou, Leland Barnard, Joshua Backman, Anthony Marcketta, Jack A Kosmicki, Andrey Ziyatdinov, Christian Benner, Colm O'Dushlaine, Mathew Barber, Boris Boutkov, et al. Computationally efficient whole-genome regression for quantitative and binary traits. *Nature genetics*, 53(7):1097–1103, 2021.

[41] Mark I McCarthy, Gonçalo R Abecasis, Lon R Cardon, David B Goldstein, Julian Little, John PA Ioannidis, and Joel N Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature reviews genetics*, 9(5):356–369, 2008.

[42] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in

nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, December 1943.

[43] Jiacheng Miao, Yixuan Wu, Zhongxuan Sun, Xinran Miao, Tianyuan Lu, Jiwei Zhao, and Qiongshi Lu. Valid inference for machine learning-assisted gwas. *medRxiv*, 2024.

[44] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

[45] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022.

[46] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Cornell Aeronautical Laboratory. Report no. VG-1196-G-8. Spartan Books, 1962.

[47] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[48] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607, 1986.

[49] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[50] Juergen Schmidhuber. Annotated history of modern ai and deep learning, 2022.

[51] scikit scikit. Template matching, april 2024.

[52] Harshit Sharma. Softmax temperature, april 2024. [Online; posted Jul 15, 2022].

[53] Benjamin B Sun, Joshua Chiou, Matthew Traylor, Christian Benner, Yi-Hsiang Hsu, Tom G Richardson, Praveen Surendran, Anubha Mahajan, Chloe Robins, Steven G Vasquez-Grinnell, et al. Genetic regulation of the human plasma proteome in 54,306 uk biobank participants. *BioRxiv*, pages 2022–06, 2022.

[54] Aiham Taleb, Matthias Kirchler, Remo Monti, and Christoph Lippert. Contig: Self-supervised multimodal contrastive learning for medical imaging with genetics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20921, 2022.

[55] Alan Mathison Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

[56] Claudia Vanea, Jelisaveta Džigurski, Valentina Rukins, Omri Dodi, Siim Siigur, Liis Salumäe, Karen Meir, W Tony Parks, Drorith Hochner-Celnikier, Abigail Fraser, et al. Mapping cell-to-tissue graphs across human placenta histology whole slide images using deep learning with happy. *Nature Communications*, 15(1):2710, 2024.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[58] Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, and Jian Yang. 10 years of gwas discovery: biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017.

[59] Jiahao Wang, Xiaodong Zhu, Kai Chen, Lei Hao, and Yuanning Liu. Hahnet: a convolutional neural network for her2 status classification of breast cancer. *BMC bioinformatics*, 24(1):353, 2023.

[60] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020.

[61] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention, 2021.

[62] Hang Yuan, Tatiana Plekhanova, Rosemary Walmsley, Amy C Reynolds, Kathleen J Maddison, Maja Bucan, Philip Gehrman, Alex Rowlands, David W Ray, Derrick Bennett, et al. Self-supervised learning of accelerometer data provides new insights for sleep and its association with mortality. *medRxiv*, 2023.

[63] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences, 2020.

[64] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022.

[65] Erdi Çallı, Ecem Sogancioglu, Bram van Ginneken, Kicky G. van Leeuwen, and Keelin Murphy. Deep learning for chest x-ray analysis: A survey. *Medical Image Analysis*, 72:102125, 2021.