Utrecht University
Department of Information and Computing Science

Artificial Intelligence Master Thesis Proposal

Pay Attention to your Neighbours: Leveraging Attention for Dimensionality Reduction

First examiner:
Dr. A. Telea

Candidate:
C.W. Smet, BSc

Second examiner:
A. Machado dos Reis, MSc

In cooperation with:
Visualisation and Graphics Group

January 7, 2026

Abstract

Dimensionality reduction techniques are essential for visualising and interpreting high-dimensional data. Traditional methods such as t-SNE are widely used due to the quality of their projections, but they suffer from limitations such as slow computation, lack of out-of-sample support, and the need for extensive parameter tuning. Some of these issues were addressed through the introduction of Neural Network Projection (NNP), which uses a supervised fully connected feedforward neural network to learn projections. However, NNP has lower projection quality compared to its ground-truth projection method. We hypothesize that this is because NNP processes samples individually, without being able to model inter-sample relationships, which form the basis of dimensionality reduction techniques such as t-SNE.

This thesis proposes an attention-based approach to projection, leveraging the multi-head attention mechanism to model interactions between multiple samples. By explicitly capturing inter-sample relationships, this method aims to improve projection quality while maintaining the scalability and generalisability of Neural Network Projection (NNP). This attention mechanism is implemented through a modified Transformer Encoder. We refer to this class of DR methods as Projection with ATtention (PAT). In addition, we provide a set of well-performing default hyperparameters that work well across a range of datasets, reducing the need for extensive parameter tuning.

We also propose an interpretability method that generalises to all attention-based projection methods. This method allows the visualisation of attention patterns at both global and local levels. It provides insight into how attention is distributed throughout the projection process globally, and which samples are most influential during the projection of an individual sample.

We propose five architecture variants, the parameters of which are based on a random, non-exhaustive search of a large parameter space. These five reasonable architectures were then tested on six datasets representing different datatypes, number of dimensions, and number of samples. In five out of six datasets, at least one modified Transformer Encoders beats the baseline model (NNP) on most metrics, often with fewer training samples. On the sixth dataset, the modified Transformer encoders beat NNP only on the True Neighbors metric, but despite this, create visually preferable projections.

Notably, across the evaluated datasets, our methods typically outperform t-SNE on the Distance Consistency (M_{DC}) metric, among others. NNP generally exhibits lower M_{DC} than t-SNE.

This increased projection quality does come at the cost of scalability, as all the PAT models are slower and scale worse than NNP.

Abstract (Nederlands)

Dimensionaliteitsreductietechnieken zijn essentieel voor het visualiseren en interpreteren van hoog-dimensionale data. Traditionele methoden zoals t-SNE leveren projecties van hoge kwaliteit, maar kennen beperkingen zoals trage berekening, het ontbreken van out-of-sample ondersteuning en de noodzaak van uitgebreide afstelling. Enkele van deze beperkingen zijn aangepakt met de introductie van Neural Network Projection (NNP), dat gebruikmaakt van een gesuperviseerd, feedforward neuraal netwerk om projecties te leren. Desondanks is de projectiekwaliteit van NNP lager dan die van de onderliggende grondwaarheidsprojectiemethode. Wij veronderstellen dat dit komt doordat NNP samples afzonderlijk verwerkt en daardoor geen inter-sample relaties kan modelleren, die juist centraal staan in technieken zoals t-SNE.

Dit proefschrift introduceert een projectiemethode die het multi-head attention-mechanisme gebruikt om interacties tussen meerdere samples te modelleren. Door inter-sample relaties expliciet vast te leggen, verbetert deze methode de projectiekwaliteit, terwijl de schaalbaarheid en generaliseerbaarheid van neuraal gebaseerde methoden grotendeels behouden blijven. Het attention-mechanisme is geïmplementeerd via een aangepaste Transformer Encoder. Deze klasse van DR-methoden wordt aangeduid als Projection with ATtention (PAT). Daarnaast presenteren wij goed presterende standaardhyperparameters die toepasbaar zijn op een breed scala aan datasets, waardoor uitgebreide afstelling minder noodzakelijk is.

Wij introduceren tevens een interpreteerbaarheidsmethode die toepasbaar is op alle attention-gebaseerde projectiemethoden. Deze methode maakt het mogelijk om attentionpatronen op globaal en lokaal niveau te visualiseren en biedt inzicht in welke samples het projectieproces het sterkst beïnvloeden.

Vijf architectuurvarianten zijn onderzocht, met parameters verkregen via een niet-uitputtende zoektocht in een grote parameterruimte. Deze architecturen zijn geëvalueerd op zes datasets met uiteenlopende datatypen, dimensies en aantallen samples. In vijf van de zes datasets presteert ten minste één aangepaste Transformer Encoder beter dan het basismodel (NNP) op de meeste metriekwaarden. Op de zesde dataset wordt NNP alleen overtroffen op de True Neighbors-metriek, terwijl de projecties visueel overtuigender zijn.

Over de geëvalueerde datasets heen presteren de voorgestelde methoden doorgaans beter dan t-SNE op de Distance Consistency-metriek. NNP vertoont daarbij meestal een lagere afstandsconsistentie dan t-SNE.

De verbeterde projectiekwaliteit gaat ten koste van schaalbaarheid, aangezien alle PAT-modellen trager zijn en slechter schalen dan NNP.

Abstract (한글)

차원 축소(Dimensionality Reduction, DR) 기법은 고차원 데이터를 시각화하고 해석하는 데 필수적이다. t-SNE와 같은 전통적인 기법은 투영 결과의 우수성으로 널리 사용되어 왔으나, 계산 속도가 느리고, 새로운 샘플을 처리하는 능력이 부족하며, 대규모 파라미터 튜닝이 필요하다는 한계를 지닌다. 이러한 문제 중 일부는 지도 학습 기반의 완전연결 순방향 신경망을 이용해 투영을 학습하는 Neural Network Projection (NNP)의 도입을 통해 개선되었다. 그러나 NNP는 ground-truth 투영 기법에 비해 투영 품질이 낮다는 한계가 있다. 본 연구에서는 이러한 문제가 NNP가 샘플을 개별적으로 처리하여, t-SNE와 같은 차원 축소 기법의 핵심을 이루는 샘플 간 관계를 모델링하지 못하기 때문이라는 가설을 제시한다.

본 논문은 다중 샘플 간 상호작용을 모델링하기 위해 멀티헤드 어텐션 메커니즘을 활용한 어텐션 기반 투영 접근법을 제안한다. 샘플 간 관계를 명시적으로 포착함으로써, 이 방법은 신경망 기반 접근법의 확장성과 일반화 성능을 유지하면서도 투영 품질을 향상시키는 것을 목표로 한다. 해당 어텐션 메커니즘은 변형된 트랜스포머 인코더를 통해 구현되며, 본 논문에서는 이러한 차원 축소 기법을 Projection with ATtention (PAT) 이라 명명한다. 또한 다양한 데이터셋 전반에서 안정적으로 동작하는 기본 하이퍼파라미터 세트를 제시함으로써, 과도한 파라미터 튜닝의 필요성을 줄이고자 한다.

아울러 본 논문은 모든 어텐션 기반 투영 기법에 적용 가능한 해석(interpretability) 기법을 제안한다. 이 방법은 전역적(global) 및 국소적(local) 수준에서의 어텐션 패턴을 시각화할 수 있게 해주며, 전역적 수준에서 투영 과정에 걸쳐 어텐션이 어떻게 분포되는지와, 개별 샘플의 투영 시 어떤 샘플들이 가장 큰 영향을 미치는지를 이해할 수 있도록 한다.

본 연구에서는 대규모 파라미터 공간에 대한 무작위적이며 비완전한 탐색을 기반으로 다섯 가지 아키텍처 변형을 제안하였다. 선별된 아키텍처들은 데이터 유형, 차원 수, 샘플 수가 서로 다른 여섯 개의 데이터셋을 대상으로 테스트되었다. 여섯 개의 데이터셋 중 다섯 개에서, 변형된 트랜스포머 인코더들 중 최소 하나는 대부분의 평가 지표에서 기준 모델인 NNP를 능가하였으며, 경우에 따라 더 적은 학습 샘플만으로도 더 나은 성능을 보였다. 남은 한 개 데이터셋에서는 True Neighbors (M_{TN}) 지표에서만 NNP를 상회하였음에도 불구하고, 시각적으로 더 선호되는 투영 결과를 생성하였다.

특히, 평가된 데이터셋에서 PAT 기법은 Distance Consistency (M_{DC}) 지표를 포함한 여러 지표에서 t-SNE보다 우수한 성능을 보였다. 반면 NNP는 대체로 t-SNE보다 낮은 M_{DC} 값을 나타냈다.

한편 모든 PAT 모델이 NNP보다 느리며, 데이터 규모가 커질수록 확장성 또한 상대적으로 낮은 것으로 나타났다. 이는 투영 품질의 향상이 확장성이라는 대가를 수반함을 의미한다.

Foreword

I want to thank my first supervisor, Dr. Alex Telea, for this opportunity and the patience and support he afforded me during my thesis in a difficult time. I also want to thank my second supervisor, Alister Machado dos Reis, for their infectious enthusiasm and the stimulating discussions we shared. Their "LLM roast" during one of the VIG group meetings was especially memorable.

I would also like to thank the other members of the VIG group for accepting me as an actively participating member during this time. Additionally I would like to thank my friend and Korean tutor, 정 유진.

Finally, I want to offer a reflection. Henri Cartier-Bresson, a photography idol of mine, famously said the following:

> *"Your first 10,000 photographs are your worst."*

This quote has taken on a personal meaning beyond photography as I bring my master's thesis, and with it, this chapter of my academic journey, to a close. Just as Cartier-Bresson acknowledged that mastery comes only after countless attempts, these past years have been filled with learning, experimentation, and gradual growth. Each draft, each experiment, and each discussion has been a step toward this culmination. This thesis was my proverbial ten-thousandth photograph, the culmination of all the work before it, and the proof that I am now a Master (of Science). While this thesis may feel like the "last photograph" of my student years, it also marks the beginning of a new phase, where the lessons I've learned will inform the work I do next.

I am grateful to everyone who has supported me along this path, and I look forward to building on this foundation in the years to come.

Contents

List of Figures

1. Introduction

High-dimensional data is everywhere, from hundreds of patient measurements in healthcare to large vector representations of words in language models. Humans can only perceive a few dimensions at a time, so visualising this data is essential to understand patterns, clusters, and relationships.

Dimensionality Reduction (DR) methods reduce high-dimensional data to lower dimensions while preserving its structure. These lower-dimensional representations, i.e. *projections*, can then be visualised with, e.g., scatterplots. Classic dimensionality reduction methods, like t-SNE [1], produce high-quality projections but are slow and typically cannot handle new data points.

A Neural network-based DR method, namely NNP [2], was introduced to address these issues by enabling fast projections and out-of-sample support, though their projections are of lower quality and can appear more diffuse than those generated with, e.g. t-SNE.

This thesis explores attention-based neural architectures to improve projection quality. By considering relationships between multiple samples simultaneously, Multi-Head Attention can capture both local and global structures, producing sharper, more informative projections while remaining scalable and interpretable.

In section 1.1, we present a more detailed context and rationale for this study. We discuss the challenges posed by high-dimensional data in various fields, such as healthcare and natural language processing, and the critical role of visualisation in making such data interpretable.

Next, in section 1.2, we identify the limitations of current dimensionality reduction techniques, like t-SNE and UMAP [1], [3], in terms of computational efficiency, out-of-sample support, and inverse projection capabilities. This discussion sets the stage for the necessity of a new approach.

Following this, section 1.3 details the specific performance characteristics (e.g., projection quality, scalability) that the proposed method is intended to achieve. We also outline the two distinct attention-based architectures to be explored, each with its own trade-offs.

In section 1.4 we introduce a set of research questions formulated to help us achieve the aims of this thesis. This is followed up by how we plan to evaluate in subsection 1.4.1. We present a short disclaimer about privacy and data ethics in subsection 1.4.2. Finally, in subsection 1.4.3 we introduce a readers guide.

1.1 Context

A common sentiment in *data visualisation* is that humans struggle with interpreting more than three or four dimensions, typically limited to space and colour. Visualisation is key to making data and the models interacting with it understandable. However, data often has far greater dimensionality than just three or four.

Consider, for example, the hundreds of potential health indicators one can collect. Examining each indicator individually is not just overwhelming, but it also obscures the interactions among them.

Alternatively, consider word embeddings from large language models. Each (sub-) word in a text is represented by a vector. These vectors, known as word embeddings, typically have a *dimensionality* of 512 [4] to 1048 [5]. Additionally, unlike e.g., health indicators, the individual variables are not inherently interpretable.

An answer to this problem is *Dimensionality Reduction (DR)*, also known as projection. DR is a category of techniques that find lower-dimensional representations, i.e., *projections*, of high-dimensional data.

For example, projections are useful to see how difficult a classification task is by projecting labelled data. The classification task is easier when the classes are clearly separated in the projection [6]. Projection is also useful to visually identify outliers in the data, as they should also be outliers in the projection.

Furthermore, projection can be used to aid in data labelling. Clusters in the projection should indicate similarities, from which follows class membership. Semi-automated labelling tools have been implemented to this effect, allowing the user to select and confirm labels for clusters of points in projections, significantly speeding up annotation time [7], [8].

Dimensionality Reduction's usefulness is not limited to explainability. It is also the backbone of many deep learning models. For example, *autoencoder* type models [9], [10] first project to a small number of dimensions using its neural encoder, to then inverse their projection back to the original data with its decoder. The encoder can be used to create powerful compressed representations of the input data, while the decoder can be used for generative AI.

1.2 Problem Definition

There are six DR characteristics that are commonly used to compare DR methods[2].

- **Projection Quality (C1)** describes both how well a projection captures the structure of the high-dimensional data, but also how intuitive it is to understand.

- **Scalability (C2)** describes the speed of both fitting and inference as a function of the size of the projected dataset.
- **Ease of use (C3)** describes how much effort is required it is to use the DR method, e.g. in terms of how long it takes to fine-tune a method's hyperparameters.
- **Genericity (C4)** describes what variety of data the DR method can handle, while this is typically limited to any kind of high-dimensional data that can be represented as real valued vectors. A more generic DR should handle datasets of various levels of, e.g., sparsity, dimensionality, and complexity.
- **Stability and out-of-sample support (C5)** describes whether the model can project previously unseen data, and what effects projecting previously unseen data have on projection quality. (Stability).
- **Inverse Mapping (C6)** describes if a model is capable of inverse projection, and the quality thereof.

The projections generated with the popular Dimensionality Reduction algorithms *t-Distributed Stochastic Neighbour Embedding (t-SNE)* [1] and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [3] are considered the best at preserving aspects of the high-dimensional data's structure as evaluated on a variety of applications (C4) with several quality metrics (high C1)[11]. However, these methods are relatively slow (Low C2). Additionally, t-SNE does not support out-of-sample data (Low C5), and neither supports inverse projection (Low C6). Furthermore, both are sensitive to data perturbations (low C5), and require parameter tuning to consistently create high quality projections (Low C3)

Recently, potential solutions were introduced for these issues. The solutions use fully-connected regression neural networks. In some examples, these solutions learn arbitrary DR methods [2], including t-SNE and UMAP, on a small subset of the data. These neural networks are then used to project the unseen data, thus mimicking the behaviour of the DR method. This method, known as *Neural Network Projection (NNP)*, complies with all the requirements listed above except one.

NNP inherently supports out-of-sample data (C5). It is also far faster compared to other DR methods like t-SNE (C2), requires no parameter tuning (C3), works with all real-valued data (C4), and can also be implemented for inverse projection (C6) [12]. However, its projection quality is lower than that of methods like t-SNE and UMAP (low C1).

We believe this is due to the principal difference in how neural networks and more classical DR methods like t-SNE process data. Neural networks process samples individually and completely rely on the network weights to encode their distribution. Meanwhile, methods like t-SNE process the entire dataset simultaneously, considering not only individual samples but also their neighbours and the global distribution of data [1].

1.3 Aim

The initial aims of this thesis are illustrated through the five of the six previously introduced DR characteristics[2].

- **Quality (C1):** Better cluster separation than state-of-the-art deep-learning-based methods for projections like NNP [13] while preserving the global structure of the underlying DR method;

- **Scalability (C2):** The proposed method must be faster than the underlying DR method, but may be slower than NNP;

- **Ease of use (C3):** The proposed method may require more parameter tuning than NNP, but must remain minimal. The method should be delivered with sensible parameters from which to start tuning;

- **Genericity (C4):** The proposed method must handle any kind of high-dimensional data that can be represented as real valued vectors;

- **Stability and out-of-sample support (C5):** The proposed method must have out-of-sample support. When adding previously unseen data-However, as re-projecting the entire dataset might increase overall projection quality when adding out-of-sample data. Therefore, stability is not a necessity if the method is fast enough;

A secondary aim of this thesis surrounds the topic of interpretability. Attention has inherently interpretable aspects, as each sample is given a certain, traceable amount of attention at each layer. As current DR methods are not inherently interpretable, this gives us the unique chance to investigate interpretability in projection methods. This method should be general to all attention-based projection methods.

1.4 Research Questions

In this thesis we explore an alternative neural architecture for DR. Specifically, we propose using Multi-Head Attention [4] in a supervised manner to bridge the functional gap described in section 1.2. Unlike traditional feedforward layers that process each sample in isolation, Multi-Head Attention computes pairwise interactions between all samples in a dataset. Each attention head learns to focus on different aspects of these relationships, allowing the model to dynamically assess how strongly each sample relates to every other sample.

This mechanism could capture both local similarities, ensuring that closely related samples are grouped together, and global structures, which preserve the overall distribution of the data. By aggregating the output from multiple

heads, the model forms a thorough understanding of the inter-sample relationships, thereby potentially more closely mimicking the behaviour of methods like t-SNE that consider local and global structures simultaneously.

Using this architecture, we aim to answer the following research questions:

> Can we design an attention-based neural architecture for projection that matches the projection quality (C1) of traditional projection methods better than state-of-the-art deep-learning-based methods for projections without sacrificing their scalability (C2) and ease of use (C3)?

This primary research question is refined into the following sub-questions:
SQ1: How does projecting via W^O within the multi-head attention mechanism compare to using a modified transformer encoder in terms of projection quality (C1)?
SQ2: How does the number of layers in the Attention-Only or Modified-Encoder projection model impact the overall performance (i.e., projection quality (C1), out-of-sample support (C5), and scalability (C2))?
SQ3: What is the effect of varying the number of attention heads on the projection quality (C1) and out-of-sample support (C5) of the model?
SQ4: How do the attention-related hyperparameters d_k, d_v, and the feed-forward dimension d_{ff} affect the projection quality (C1) of attention-based projection methods?
SQ5: What is the impact of dataset size and dimensionality on the scalability (C2) of Attention-based models?

And a secondary research question for this thesis is:

> Can we design a method for visualising attention patterns in projection usecases for analysis?

1.4.1 Evaluation

To validate the proposed methods, experiments will be carried out on standard benchmark datasets such as MNIST [14] and Fashion-MNIST (fMNIST) [15], with potential extensions to synthetic and non-image datasets to further demonstrate genericity (C4). The performance will be compared against existing Dimensionality Reduction techniques using projection quality metrics (C1) and additional relevant performance measures(C2). This will confirm whether the proposed method meets the defined characteristics.

The code for the experiments, including architecture, evaluation, and development, is shared publicly for reproducibility purposes [16].

1.4.2 Ethics and Privacy Statement

The Ethics and Privacy Quick Scan of the Utrecht University Research Institute of Information and Computing Sciences was conducted (see Appendix A).

It classified this research as low-risk with no further ethics review or privacy assessment required.

1.4.3 Readers Guide

Chapter 2 provides an in-depth background and technical review of related work. These sections delve into the theory of dimensionality reduction, comparisons of current methods for projection and inverse projection methods, how these methods can be evaluated, and the mechanics of attention and transformer architectures.

In chapter 3 thereafter, we explore the methodology behind this thesis. These sections describe the considerations behind architectural choices, how the proposed models will be evaluated, and how the datasets used to evaluate them on were chosen. Additionally, this chapter includes a section proposing a method for visualising attention patterns for projection usecases.

We expand on these choices in chapter 4: Experimental Setup. These sections go further into the technical details of the proposed models, data preprocessing and evaluation.

Chapter 5 shows the results of the evaluation steps, including a quantitative and qualitative analysis of projection quality over the chosen datasets, a more detailed convergence analysis over MNIST, and a scalability analysis on synthetic data. It also includes an analysis of the projection patterns of one of the models on MNIST using our proposed attention visualisation method.

Finally, chapter 6 summarises the proposed methods and their evaluations to answer the research questions posed in section 1.4. It also comments on some methodological limitations, and includes a section proposing future research.

2. Background and Related Work

Understanding high-dimensional data is a fundamental challenge in many fields, from machine learning and bioinformatics to natural language processing and data visualisation. Since humans struggle to interpret data beyond three dimensions, Dimensionality Reduction (DR) techniques are used to create meaningful low-dimensional representations of high-dimensional datasets. These techniques aim to preserve essential structures within the data while reducing complexity, making it easier to visualise and analyse patterns.

Section 2.1 provides an overview of key dimensionality reduction methods, with a focus on both traditional non-machine-learning-based approaches and more recent neural-based techniques. Subsection 2.1.1 explains how Dimensionality Reduction and Machine Learning relate to each other using their respective notations and definitions. Subsection 2.1.2 discusses classic DR methods such as Principal Component Analysis (PCA), t-SNE, and UMAP, highlighting their strengths and limitations. Subsection 2.1.3 explores machine-learning-based DR techniques, including autoencoders and Neural Network Projection (NNP), which offer improved scalability and out-of-sample support but often compromise on projection quality.

Next, section 2.2 describes the projection quality metrics which can be used to evaluate DR techniques. Continuing from this, section 2.3 discusses the datasets upon which our methods are evaluated.

Finally, section 2.4 introduces the Transformer and its multi-head attention mechanism, explaining how attention mechanisms can address challenges in DR by explicitly modelling inter-sample relationships.

By establishing this foundation, the chapter sets the stage for the proposed attention-based projection method, situating it within the broader landscape of dimensionality reduction research.

2.1 Dimensionality Reduction

Dimensionality Reduction (DR), also known as projection, is mapping high-dimensional data to a lower-dimensional representation. DR allows for visualising high-dimensional data, which is an important tool for explainability and can aid in machine learning research [17].

2.1.1 Notations and Definitions

Let $D = \{x_i\}$ be a dataset of n-dimensional samples or points x_i. A point is defined as $x_i = (x_i^1, \ldots, x_i^n)$, where $1 \leq i \leq n$. That is, a point x_i is defined

in n dimensions or $x_i \in \mathbb{R}^n$, where each dimension represents a feature of the point. Likewise, the set $X^j = (x_1^j, \ldots, x_n^j)$, $1 \leq i \leq n$ are referred to as the dimensions or features of the dataset D. With few exceptions, Dimensionality Reduction (DR) and Machine Learning (ML) algorithms only support real-valued features[1], i.e., $x_i \in \mathbb{R}^n$. Ergo, for the purposes of this thesis, we assume this to always be true.

A DR technique takes a dataset D and finds a representation of it. That is, the technique, or projection P is a function that maps D to $P(D) = \{y_i\}$, where $y_i \in \mathbb{R}^q$ is the projection of x_i, making $P : \mathcal{P}(\mathbb{R}^n) \to \mathcal{P}(\mathbb{R}^q)$. Typically, the representation is of a far lower dimensionality than the original, i.e., $q \ll n$. For data visualisation, 2- or 3D projections are the norm. This thesis aims to mimic existing DR methods for creating scatterplots. Ergo, for the purposes of this thesis, 2D projections ($q = 2$) is all that is considered.

For the purposes of this thesis, we split DR methods into two groups: Machine Learning (ML)- and non-ML methods. How ML relates to projection requires some further definitions. Defining Machine Learning (ML) first requires defining an extension of dataset D as defined in subsection 2.1.1. Namely, that of the annotated dataset. An annotated dataset D_a extends the dataset D by associating each sample $x_i \in D$ with an annotation $y_i \in A$.

In the context of ML, the goal is to construct models $f : \mathbb{R}^n \to A$, which, when applied to a test set $D_T \subset D_a$, produce predictions $f(x_i)$ that are as close as possible to the true annotations y_i for each $x_i \in D_T$. This closeness is quantified by a error function $d : A \times A \to \mathbb{R}^+$, such that ideally, $d\big(f(x_i), y_i\big) \approx 0$ for all $x_i \in D_T$. The model f is trained on a separate training set $D_t \subset D_a$ (with $D_t \cap D_T = \varnothing$) in order to adjust its parameters to minimize this prediction error.

For the Deep Learning (DL) subset of ML models, the error function is typically replaced with a *loss* function, which functions similarly to the distance function in that the model f is trained to reduce it to zero. However, it must be differentiable.

Machine learning models can be broadly classified into two types: classifiers and regressors. Classifiers predict categorical labels. Regressors, on the other hand, predict real and continuous, values. The latter model best aligns with DR, as projection to 2D is equivalent to $f : \mathbb{R}^n \to A$ where $A \in \mathbb{R}^2$.

2.1.2 Non-ML-based Projection Methods

There are numerous DR methods. One relatively recent survey paper lists over eighty known techniques, of which it evaluates 44[11]. It would be unreasonable to describe them all. Instead, we choose to highlight three important non-ML-based DR methods: PCA, t-SNE, and UMAP. The latter two perform the

[1]When features are not real-valued, e.g., because they are categorical, they are typically pre-processed with a method like one-hot encoding to make them real-valued.

best in preserving important aspects of the high-dimensional data in various applications according to a wide variety of quality metrics as tested in the aforementioned survey[11]. One of these methods, namely t-SNE, is also the method that is mimicked using our proposed methods.

In this thesis, we focus on mimicking dimensionality reduction techniques that belong to the class of neighbourhood preservation methods. These approaches are designed to maintain the local structure of high-dimensional data in a low-dimensional representation. That is, they ensure that points which are close together in the original space remain close in the projection, and similarly, points that are far apart continue to be separated. The precise relationships between neighbourhoods, i.e., global structure, is less important for visualisation tasks.

A classic example of a DR method, that should be mentioned for historical reasons, is **Principal Component Analysis (PCA)** [18], which identifies the directions (principal components) that capture the largest variance in the data. By projecting the data onto these components, PCA provides a linear projection of the high-dimensional data to a lower-dimensional representation. Due to its deterministic and simplistic use, it scores high on scalability (C2) and ease of use (C3). However, due to its linearity and poor ability to capture local structure, its projections have poor quality (C1).

Another widely used technique is **t-Distributed Stochastic Neighbour Embedding (t-SNE)** [1]. t-SNE transforms high-dimensional distances between all points into probabilities and then seeks a low-dimensional probability distribution that minimizes the Kullback-Leibler divergence between these probability distributions. This approach is particularly effective at preserving the local structure, making clusters of similar points clearly discernible (high C1). However, it requires parameter tuning to make these high-quality visualisations (low C3), while also being stochastic and lacking out-of-sample support (C5). Finally, it is also quite slow (low C2).

More recently, **Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)** [3] has emerged as an alternative to t-SNE, offering improved speed and scalability while preserving high projection quality. Like t-SNE, UMAP seeks to preserve local neighbourhoods. However, it assumes the data is distributed on a Riemannian manifold, and explicitly tries to find a lower-dimensional projection of that data with the closest topological structure.

Compared to t-SNE, UMAP generally produces similar projection quality (C1) but is considerably faster (higher C2) due to its more efficient optimization process. Additionally, unlike t-SNE, it supports out-of-sample data, allowing new data points to be projected without recomputing the entire embedding (C5). However, UMAP remains sensitive to small changes in input data (lowering C5) and, like t-SNE, requires careful parameter tuning to achieve optimal results (lower C3).

2.1.3 ML-based Projection Methods

The non-ML dimensionality reduction methods discussed in subsection 2.1.2 exhibit several limitations. Typically, they require extensive parameter tuning (C3), can be relatively slow even when optimized (C2), or may yield projections with suboptimal quality (C1). Furthermore, these classic techniques often lack support for out-of-sample data and tend to suffer from stability issues (C5), while they do not inherently provide an inverse projection (C6). As a promising alternative or complement, Machine Learning-based implementations have been introduced.

The first prominent approach employed an **autoencoder**[9], [10], an unsupervised model that learns a mapping from high-dimensional space to a low-dimensional representation via its encoder, while its decoder learns to reconstruct (i.e., inverse project) the original input. Autoencoders are advantageous in that they are fast (C2), require no labels (C3), naturally support out-of-sample data (C5), and provide a built-in mechanism for inverse projection (C6). However, the low-dimensional representations produced by autoencoders often lack the projection quality (C1) observed in classic DR methods.

More recently, a supervised approach known as *Neural Network Projection (NNP)* has been introduced [2], [13]. In NNP, a fully-connected regression neural network is trained to learn an arbitrary DR method. A small subset of the data is first projected using a standard method (e.g., t-SNE or UMAP), and this projection $P(D)$ serves as the ground truth A for training the network. Once trained, the network can be applied to unseen data, thus offering a fast, generic (C4) solution that supports out-of-sample projections (C5). Although NNP is simpler to use (C3) and faster (C2) than many classic methods, its projections typically exhibit lower quality (C1) compared to the original DR techniques it was trained off of.

In the following years, attempts were made to improve on NNP. First was *K-Nearest Neighbours Neural Network Projection (kNNP)* [19], which added limited multi-sample support by considering not just the sample to be projected but also its k nearest neighbours. This increased overall projection quality compared to NNP, but was still poorer than ground truth (C1).

Eventually, the research focus shifted from mimicking traditional DR methods to self-supervised approaches designed specifically for data visualisation [20], [21]. These methods can be seen as a modification of the autoencoder [9], [10].

Self-Supervised Network Projection (SSNP) optimises for cluster separation by adding a clustering learning objective, leading to well-defined groupings (C1). This additional learning objective also makes it considerably faster to converge than an equivalent autoencoder (C2). However, because SSNP's training objectives prioritize cluster discovery, and completely disregard traditional DR outputs, the resulting projections lack some of the organic, intuitive layouts found through methods like t-SNE (lower C1).

The most recent DR technique in this category of self-supervised models is *Shape-Regularized Multidimensional Projections (SHaRP)*[22]. SHaRP is an extension of SSNP, which adds user-controlled shape regularisation to curb SSNP's unnatural looking cluster shapes. While this did result in more pleasant visualisations, it does come at the cost of accurately representing the local structure of the high-dimensional data. In turn, results in a mixed projection quality (C1). Overall, it is also slightly slower than SSNP, while still being faster than autoencoders, t-SNE and UMAP (C2).

Self-supervised projection methods excel at forming well-separated clusters, yet their outputs often lack the organic, intuitive structure found in classic DR techniques like t-SNE. On the other hand, supervised approaches such as NNP generate visually coherent projections that resemble traditional DR methods but struggle to achieve the same level of cluster separation. This contrast highlights a fundamental limitation in current neural projection methods: they either prioritize local structure or global structure. This trade-off reflects a deeper issue in how neural projection methods process data, raising the question of whether a different approach could better capture both local and global structures.

2.2 Projection Quality Metrics

Projections should aim to preserve the structure of the high-dimensional data. Whether a projection properly preserves this structure can be measured using *Projection Quality Metrics*. These metrics are a function $M\big(D, P(D)\big) \to \mathbb{R}^+$. There are at least 17 known projection quality metrics [23], [24]. However, many of these have been shown to have high correlation or can easily be fooled without generating high-quality projections [24]. Hence, we chose to use the eight metrics that Machado et al. [24] identify as being more difficult to fool and having low inter-metric correlation.

Trustworthiness (M_t): This metric, with values in $[0, 1]$ (1 is best), measures the proportion of points in D that are also close in $P(D)$. This indicates how many local patterns in the high-dimensional data are preserved in the projection [25]. Formally, let $U_i(K)$ be the set of points that are among the K nearest neighbours of i in the projected space but not in the original space, and let $r(i, j)$ be the rank of point j among the nearest neighbours of i in the projection. We set $K = 7$ following [11]. M_t is defined as:

$$M_t = 1 - \frac{2}{n\,K\,(2n - 3K - 1)} \sum_{i=1}^{n} \sum_{j \in U_i(K)} \big(r(i, j) - K\big). \tag{2.1}$$

Continuity (M_c): Also in $[0, 1]$ (1 is best), M_c checks how many points that are close in the original space remain close in the projection [25]. Let

$V_i(K)$ be the set of points that are among the K nearest neighbours of i in the original space but not in the projection, and let $\hat{r}(i,j)$ be the rank of j among the nearest neighbours of i in the original space. As with M_t, $K = 7$. M_c is defined as:

$$M_c = 1 - \frac{2}{n\,K\,(2n - 3K - 1)} \sum_{i=1}^{n} \sum_{j \in V_i(K)} \big(\hat{r}(i,j) - K\big). \tag{2.2}$$

Scale Normalised Stress (M_σ): Scale Normalized Stress $[0, +\infty]$ (0 is better) measures how well a projection preserves all pairwise distances with the use of a global scaling factor. It evaluates how closely the projected distances $\Delta^t(P(x_i), P(x_j))$ match the original distances $\Delta^n(x_i, x_j)$ after choosing the best possible scale $\alpha > 0$ [26]. The value of α is selected to minimize the squared error between the two distance matrices:

$$M_\sigma = \min_{\alpha>0} \frac{\sum_{i,j} \big[\Delta^n(x_i, x_j) - \alpha \Delta^t(P(x_i), P(x_j))\big]^2}{\sum_{i,j} \Delta^n(x_i, x_j)^2}. \tag{2.3}$$

The optimal scaling factor, as used by Machado[27], has the closed form

$$\alpha = \frac{\sum_{i,j} \Delta^n(x_i, x_j)\, \Delta^t(P(x_i), P(x_j))}{\sum_{i,j} \left[\Delta^t(P(x_i), P(x_j))\right]^2}, \tag{2.4}$$

Neighbourhood Hit (M_{NH}): This metric, in $[0, 1]$ (1 is best), is defined for labelled data [28]. Let $N_i(K)$ be the set of K neighbours of i in the projection, and let l_i be the label of point i. M_{NH} is the fraction of neighbours that share the same label as i:

$$M_{NH} = \frac{1}{n\,K} \sum_{i=1}^{n} \Big| \{\, j \in N_i(K) : l_j = l_i \,\} \Big|. \tag{2.5}$$

True Neighbors (M_{TN}): This metric, in $[0, 1]$ (1 is best), measures neighbourhood preservation between the high-dimensional space and the low-dimensional projection [29]. Let $N_i^H(K)$ be the set of the K nearest neighbours of point i in the high-dimensional space, and $N_i^L(K)$ the corresponding set in the projection. The True Neighbors rate is the fraction of neighbours in the projection that are also neighbours in the original space:

$$M_{\mathrm{TN}} = \frac{1}{n\,K} \sum_{i=1}^{n} \Big| \{\, j \in N_i^L(K) : j \in N_i^H(K) \,\} \Big|. \tag{2.6}$$

Distance Consistency (M_{DC}): Distance Consistency, in $[0, 1]$ (1 is best), estimates visual separation in a projection. It quantifies the fraction of projected points that remain closer to the centroid of their own class than to any other class centroid [30].

Let $\mathrm{centr}'(c)$ denote the centroid of class c in the projection and $\mathrm{clabel}(i)$ the class label of point i. The metric is:

$$M_{DC} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\Big[\, \mathrm{clabel}(i) = \arg\min_{c} \big\| p_i - \mathrm{centr}'(c) \big\| \Big], \tag{2.7}$$

where $\mathbf{1}[\cdot]$ is the indicator function, which returns 1 when the point is closest to the centroid of its own class and 0 otherwise.

Procrustes Statistic (M_P): The Procrustes statistic, in $[0, \infty)$ (0 is best) measures how well the projection represents the original data after a number of transformations [31]. For each point, a neighbourhood in the projection is rescaled, rotated, and translated so that it best matches the corresponding neighbourhood in the original space, and the residual reconstruction error is computed. The final score is the average of these local errors. Let $x_i \in \mathbb{R}^D$ denote the i-th data point in the original space, and $y_i \in \mathbb{R}^d$ its corresponding point in the projection. Let $S_k^D(x_i)$ be the matrix whose rows are the k-nearest neighbours of x_i in the original space, and let $S_k^2(y_i)$ be the analogous matrix of the k-nearest neighbours of y_i in the projection. Let $H = I_k - \frac{1}{k}\mathbf{1}\mathbf{1}^\top$ denote the $k \times k$ centering matrix. For two neighbourhood matrices $X, Y \in \mathbb{R}^{k \times d}$, define

$$G(X, Y) = \| HX - HYA^\top \|_F^2, \tag{2.8}$$

where $A = UV^\top$ and $U\Sigma V^\top$ is the singular value decomposition of $(HX)^\top (HY)$.

The local Procrustes error around point i is

$$G\big(S_k^D(x_i),\, S_k^2(y_i)\big). \tag{2.9}$$

The global Procrustes statistic is then:

$$M_{\mathrm{P}} = \frac{1}{n} \sum_{i=1}^{n} \frac{G\big(S_k^D(x_i),\, S_k^2(y_i)\big)}{\| H\, S_k^D(x_i)\|_F^2}. \tag{2.10}$$

Pearson Correlation of Distances(M_r):

The Pearson Correlation of Distances $[-1, 1]$ (± 1 best, 0 worst) measures how well a projection preserves the internal structure of the original data. It does this by calculating the linear correlation between the intersample distances in D and $P(D)$ [32]. If the intrinsic structure of the data is unchanged by scaling, then the correlation between the distance vectors of the original space and the projection provides an effective global quality measure.

Let DV be the vector of all pairwise distances in the original space and let DV$'$ be the corresponding vector in the projection. The correlation coefficient is

$$M_{pc} = \rho(\mathrm{DV}, \mathrm{DV}') = \frac{\langle \mathrm{DV}\, \mathrm{DV}' \rangle - \langle \mathrm{DV} \rangle \langle \mathrm{DV}' \rangle}{\sigma(\mathrm{DV})\, \sigma(\mathrm{DV}')}. \tag{2.11}$$

2.3 Datasets

In this section, we give a technical overview of a set of datasets commonly used in dimensionality reduction research. Each dataset is or was a commonly used as benchmark in their respective field. Five of the six datasets listed in this section, namely fMNIST, CIFAR-10, HAR, Spambase, and CNAE-9, are a subset of the more complete list of 18 dimensionality reduction benchmark datasets described in Espadato et al.'s survey of dimensionality reduction techniques [11]. The sixth dataset, MNIST, is included for historical reasons. For more on why these datasets were chosen, see section 3.4.

2.3.1 MNIST

The modified National Institute of Standards and Technology Database (MNIST) dataset [14] is a benchmark for handwritten digit recognition composed of 70,000 grayscale 28×28 images. It has been widely used to evaluate and compare classification models and learning algorithms, and it played a central role in early developments of convolutional networks and gradient-based optimization for image tasks. Results on MNIST are often reported to demonstrate a model's capacity to learn low-level visual features and to provide a sanity check before moving to more challenging datasets. An example digit from each class in the dataset is shown in Figure 2.1.

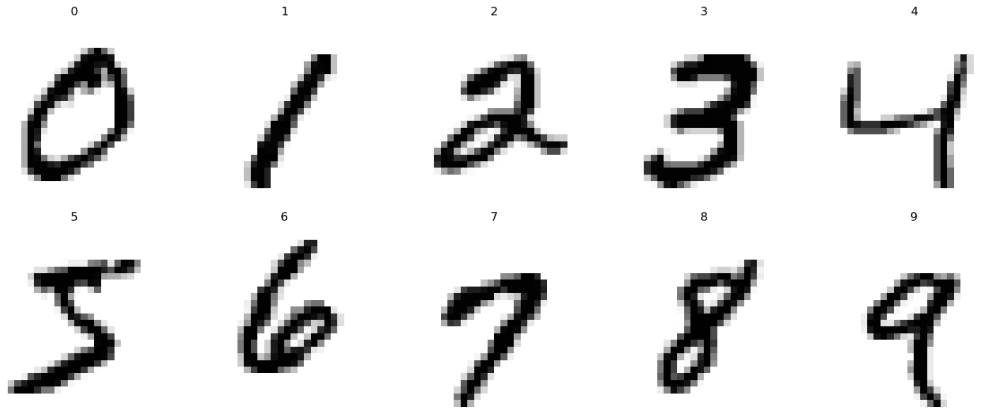


Figure 2.1: Examples of the handwritten digits found in MNIST. One for each class.

2.3.2 fMNIST

The Fashion-MNIST (fMNIST) dataset [15] was introduced as a drop-in replacement for MNIST with the same image size and predetermined train-test split but containing ten classes of fashion items (e.g., shirts, shoes, bags). fMNIST preserves MNIST's simplicity while offering greater visual and semantic complexity, making it useful for evaluating robustness and generalization of image classifiers where MNIST may be too easy. An example clothing item from each class in the dataset is shown in Figure 2.2.

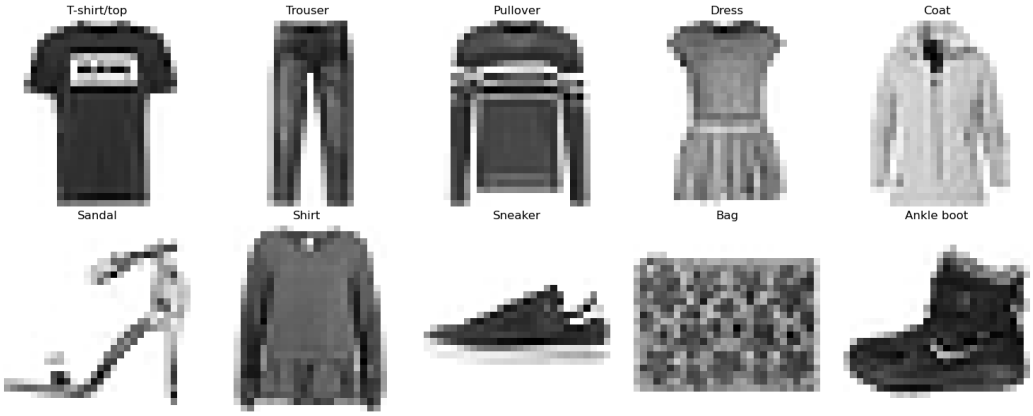


Figure 2.2: Examples of the grayscale photographs of fashion items found in fMNIST. One for each class.

2.3.3 CIFAR-10

The Canadian Institute For Advanced Research datasets (CIFAR-10 and CIFAR-100) [33] are small natural-image benchmarks composed of 32×32 colour images across multiple object categories. CIFAR has served as a standard testbed for convolutional and residual architectures, data augmentation strategies, and regularization techniques in mid-scale image recognition. Its greater intra-class

variability and colour information make it substantially more challenging than MNIST-like datasets. An example image for each class is shown in Figure 2.3.

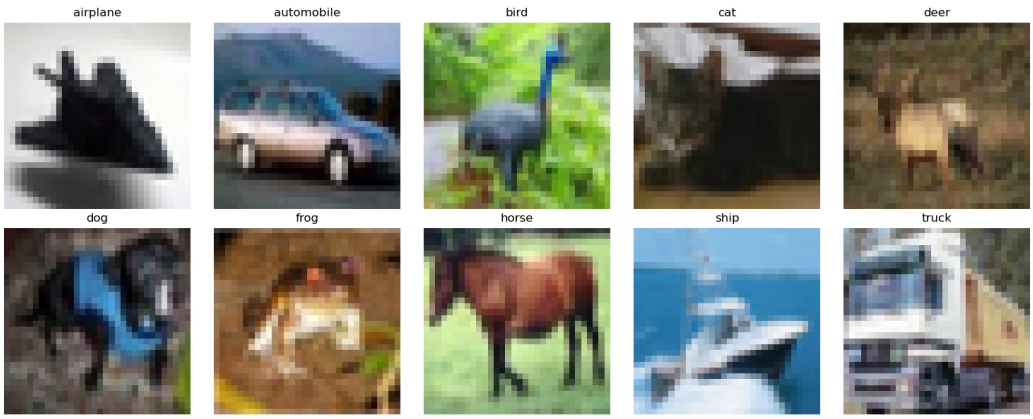


Figure 2.3: Examples of the colour photographs found in CIFAR-10. One for each class.

2.3.4 HAR

The Human Action Recognition (HAR) dataset [34] contains wearable-sensor recordings collected from 21 participants performing daily activities. Each recording consists of synchronized 3-axis accelerometer and gyroscope signals sampled at 50Hz collected via a smartphone strapped to the subjects' waist (see Figure 2.4). The raw signals were lightly filtered to reduce noise, and a set of descriptive features, which capture basic movement patterns such as average motion, variability, simple frequency information, and signal magnitude, was computed from short sliding windows.

Each activity segment is represented as a 561-dimensional feature vector, with individual sequences containing 281-409 time steps. This benchmark can be used for evaluating time-series classification models and temporal feature extraction.

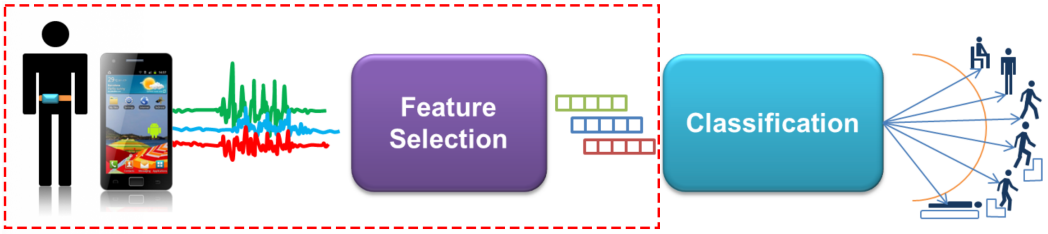


Figure 2.4: Activity Recognition process pipeline taken from Hutchison et al. [34], modified with a dotted rectangle to emphasize data extraction pipeline.

2.3.5 Spambase

The Spambase dataset [35] consists of email samples encoded as numerical feature vectors that capture word-frequency statistics, specific character fre-

quencies, and additional heuristic features. Each sample is represented as a 57-dimensional real-valued vector, making the dataset a suitable benchmark for analyzing binary (text) classification, feature selection, and robustness to noisy features.

2.3.6 CNAE-9

The National Classification of Economic Activities (Classificação Nacional de Atividades Econômicas) (CNAE-9) dataset [36] comprises 1080 text documents categorized into nine administrative classes. Each document is represented as an 856-length sparse vector of word frequencies. CNAE-9 is used to study multiclass text classification, feature extraction, and the behaviour of predictive models on sparse input representations.

2.4 Transformers and Attention

The principal technology behind the DR method proposed in this thesis is the *Attention* mechanism. This mechanism finds its origins in the field of *Natural Language Processing (NLP)*.

For some time, variable-length sequence processing models relied on Recurrent Neural Networks (RNN) for tasks such as text processing [37]. However, these models struggled with long-range dependencies. Information from earlier timesteps tended to vanish as later timesteps were processed. Although later RNN-variants alleviated this issue to some extent, they did not completely resolve it.

In the field of Natural Language Processing (NLP), the encoder-decoder architecture emerged as an alternative to pure RNNs [38], [39]. In these models, an encoder compresses all the information from the input sequence into a single fixed-length context vector, which the decoder then maps back to a variable-length output sequence. Although this approach improved upon pure RNNs, it still relied on recurrence to process variable-length sequences and thus inherited the problem of long-range dependency loss. To address this limitation, the *Attention* mechanism was introduced [40].

The *Attention* mechanism enables the decoder to access information from all encoder hidden states, not limiting it to just the final state, i.e., the single fixed-length context vector. By assigning a learned score to each hidden state and computing a weighted sum of these states at each decoding step, the model can effectively "pay attention" to the most relevant parts of the source sequence. This approach not only improved performance, especially on longer sentences, but also provided some level of interpretability regarding which parts of the input were most influential for each output.

Despite these improvements, encoder-decoder models with attention still relied on recurrence, which limited parallelization, which in turn severely hampered scalability. This limitation motivated Vaswani et al. to propose the

Transformer architecture [4]. In the Transformer, the overall encoder-decoder framework is retained, but the recurrence is replaced entirely by attention. Importantly, the attention mechanism is fully parallelizable, which significantly reduces training time and has contributed to the Transformer's state-of-the-art performance. Today, Transformer-based models form the basis not only of most state-of-the-art NLP systems (including large language models) but also of advances in computer vision (e.g., Vision Transformers) and audio processing.

While recursion forces an order onto a set of samples, attention allows neural networks to process a variable-length *unordered* set while still being able to model inter-sample relationships. This unique quality may allow a neural DR method to better mimick classic DR techniques such as t-SNE.

2.4.1 Multi-Head Attention

Vaswani et al. introduced a variant of dot-product (multiplicative) attention, namely **Scaled** Dot-Product Attention, see Equation 2.12. The only change from dot-product attention is that it scales the dot products by $\frac{1}{\sqrt{d_k}}$ in order to avoid disappearing gradients[4]. There are three vectors to account for: Query: Q, Keys: K, Values: V. The queries Q and keys K must be of the same dimension d_k, whereas the values Vcan be differently sized, i.e., d_v.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.12}$$

Moreover, the Transformer introduces *Multi-Head Attention*, where several attention layers (heads) operate in parallel (Figure 2.5). Each head learns to capture different inter-token relationships simultaneously, allowing the model to simultaneously attend to multiple kinds of contextual information. Building on scaled dot-product attention, the Transformer introduces multi-head attention, which applies this mechanism in parallel; formally, it is defined in Equation 2.13.

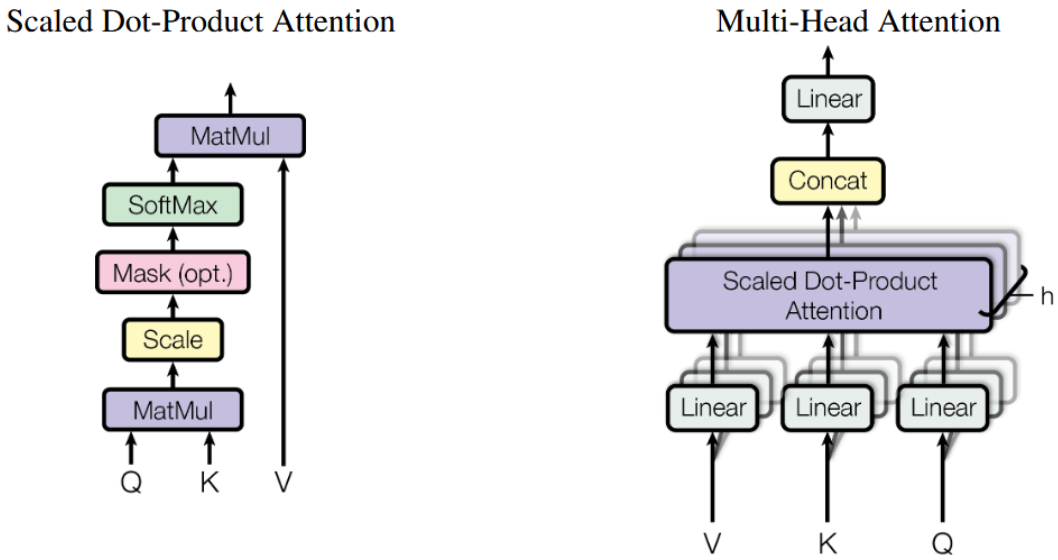


Figure 2.5: Scaled Dot-Product Attention compared to Multi-Head Attention, taken from [4].

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{where head} &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.13}$$

The (Linear) projections are matrices of learnable parameters. Here d_{model} is the dimensionality of the input sequence, $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $W^O \in \mathbb{R}^{d_v \times d_{\text{model}}}$.

The authors suggest that for each of the h heads, $d_k = d_v = d_{\text{model}}/h$, as the reduced dimension of each head ensures that the total computational cost of multi-head attention is similar to single-head attention[2] with full dimensionality.

The Transformer employs multi-head attention in three distinct ways:

1. **Encoder Self-Attention:** Within the encoder, self-attention layers are employed, where the queries (Q), keys (K), and values (V) are all generated from the output of the preceding encoder layer. This allows each position in the encoder to attend to all positions in that layer. For the **first** self-attention layer, the input is the input sequence.
2. **Decoder Self-Attention:** Similarly, the decoder utilizes self-attention layers. However, in this case, each position in the decoder is permitted to attend only to positions up to and including itself, in order to preserve the auto-regressive property required for generation. This restriction is

[2] I.e, 'regular' Attention

implemented in the scaled dot-product attention by masking out (i.e., setting to $-\infty$) any elements corresponding to illegal connections.

3. **Encoder-Decoder Attention:** In some layers in the decoder, the queries Q are derived from the previous decoder layer, while the keys K and values V are obtained from the encoder's output. This design enables every position in the decoder to attend to all positions in the input sequence, thereby mimicking the typical encoder-decoder attention mechanisms found in traditional sequence-to-sequence models [39].

2.4.2 Complete Transformer Architecture

The Transformer is composed of more than just multi-head attention. As discussed previously, the Transformer is an encoder-decoder-class architecture. The encoder maps a variable-length input sequence into a sequence of context vectors of equal length, while the decoder generates an output sequence autoregressively[4]. This design is effective for language. However, since we must assume our data, in the context of DR, is an unordered set, and thus does not exhibit an inherent sequential order, and thus does not benefit from autoregressive generation, we focus exclusively on the encoder and disregard the decoder.

RNNs inherently support the sequential nature of language by processing the sequence sequentially. Transformers do not. Ergo, a positional encoding is added to the input sequence before it is fed to the encoder. Since our data is unordered, we should omit the positional encoding.

The encoder consists of a stack of identical layers ('encoder layers'). Each encoder layer contains two sub-layers:

1. A multi-head attention layer.
2. A position-wise fully connected feed-forward network, which is applied individually and identically for each position in the input sequence.

The feed-forward network is defined as

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.14}$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ projects from the model dimension d_{model} to an intermediate dimension d_{ff}, and $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ projects back to d_{model}. Note that the intermediate dimension d_{ff} may vary across layers.

Each of the two sub-layers in an encoder layer is wrapped in a residual connection[41], followed by layer normalization[42]. That is, the output of each sub-layer is computed as

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x)) \tag{2.15}$$

where Sublayer(x) denotes the function implemented by the sub-layer itself. Succinctly, an encoder layer can be expressed as

$$\begin{aligned}\text{Layer}(x) = \text{LayerNorm}\Big(&\text{FFN}\big(\text{LayerNorm}(\text{MultiHead}(x) + x)\big) \\ &+ \text{LayerNorm}(\text{MultiHead}(x) + x)\Big)\end{aligned} \tag{2.16}$$

This use of residual connections requires that all sub-layers produce outputs with the same dimension, namely d_{model}.

2.4.3 Interpretability and Attention Visualisation

Since the introduction of the Transformer architecture, attention mechanisms have often been presented as an interpretable component of neural sequence models [43]. Early work following the introduction of the Transformer [4] highlighted the potential of attention weights to reveal which input tokens a model focuses on when forming predictions. The Tensor2Tensor library [44] subsequently provided initial functionality for viewing attention patterns in sequence-to-sequence tasks.

One of the most widely used developments in this space is BertViz, introduced by Vig [45] and extended in later releases [46]. BertViz, cited in over 900 publications, offers an interactive suite of attention visualisations tailored to transformer models. It expands upon prior research by supporting encoder- and decoder-only models, such as BERT [5] and GPT-2 [47] respectively. Additionally, unlike earlier static plots, it enables multi-level interactive inspection of attention behaviour, supporting both global analysis across layers and heads and fine-grained examination of individual computations.

BertViz provides three complementary visualisation modes: the head view, the model view, and the neuron view.

Head View: The head view displays the attention patterns of one or more heads within a selected layer. As illustrated in Figure 2.6, it shows how each head distributes attention from a chosen token to the rest of the sequence, with colour coding used to differentiate heads. This representation is based on that found in Tensor2Tensor, and remains a common method for examining phenomena such as syntactic dependencies or token interactions.

Model View: The model view provides a global overview of attention behaviour across the entire architecture. In this layout, layers form rows and attention heads form columns, enabling inspection of attention distributions at multiple depths (Figure 2.7). By selecting an input token, researchers can trace how attention patterns evolve across layers, making this view useful for identifying global structural tendencies within pretrained models.

Neuron View: The neuron view offers the most detailed perspective by visualising the individual components of the query and key vectors that con-

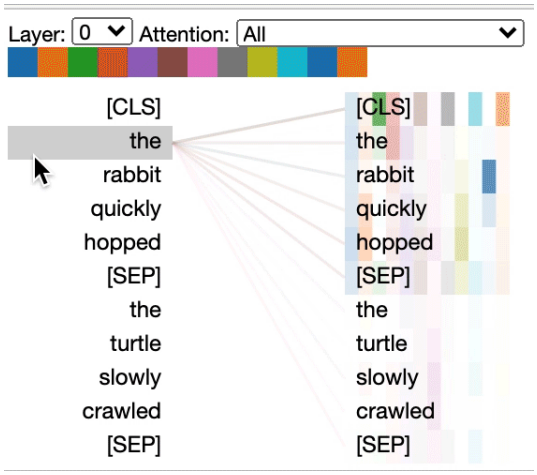


Figure 2.6: 'Head view' taken from BertViz[45], [46], showing what individual heads, as indicated by colour, in layer 0 attends to for the word 'the'.

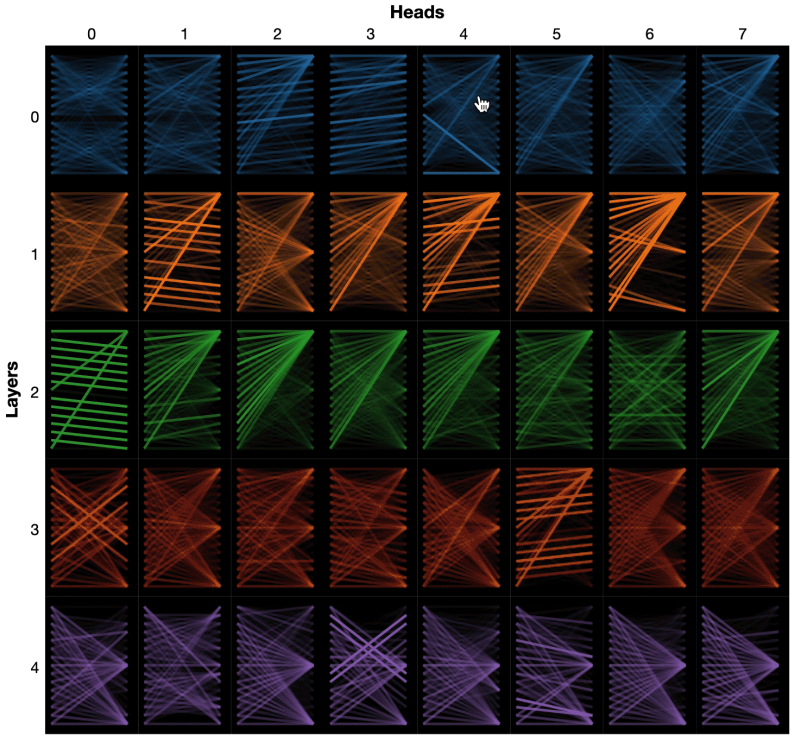


Figure 2.7: 'Model view' taken from BertViz[45], [46], showing a global view of attention weights per individual head and layer, as indicated by colour.

tribute to attention scores. As shown in Figure 2.8, it highlights neuron-level activations within a specific head and layer, thereby illustrating how particular attention weights arise. This mode supports more local mechanistic analyses of transformer behaviour and complements the higher-level insights provided by the head and model views.

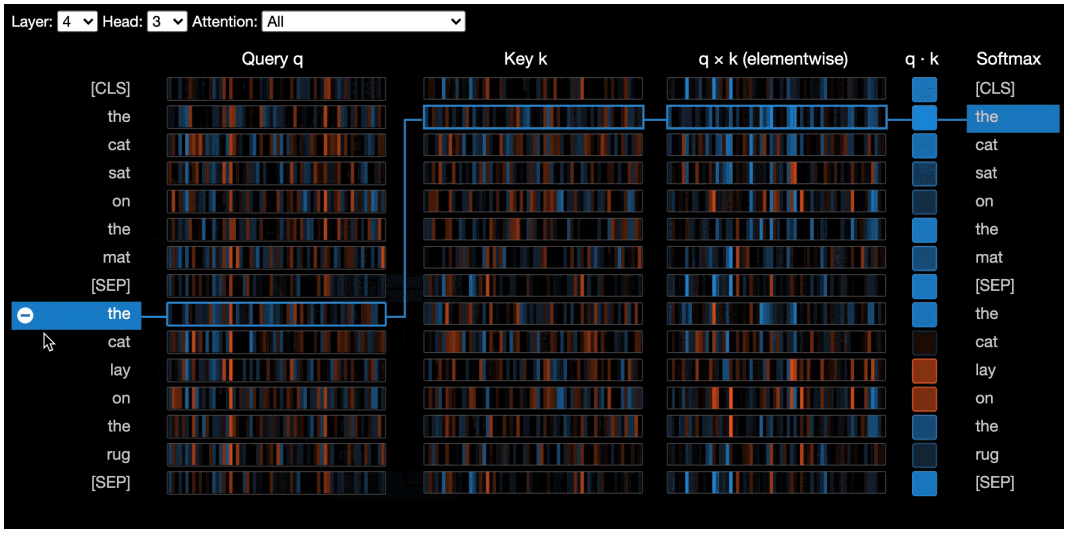


Figure 2.8: 'Neuron view' taken from BertViz[45], [46], showing the activations of neurons in the query and key of head three of layer four for the word 'the'.

2.4.4 Transformer Dimensionality Reduction

Transformers have previously been applied to DR tasks in computer vision, leveraging the spatial relationships among adjacent pixels [48]. Ran et al. employ a standard Vision Transformer (ViT) architecture [49], in which images are partitioned into patches.

The model they propose, Transformer-DR, is configured as an autoencoder. The encoder module performs Dimensionality Reduction, and the decoder does inverse projection. Although this approach produces high-fidelity inverse projections, the intermediary low-dimensional projections tend to be diffuse and visually uninformative, a behaviour characteristic of unsupervised autoencoder-based techniques [9]. Consequently, the embeddings exhibit limited interpretability and are easily outperformed by methods such as t-SNE and NNP.

As mentioned in subsection 2.1.3, adding a self-supervised objective to a non-attention-based autoencoder has positively contributed to projection quality in the past (SSNP [20]), but does not reach the projection quality of t-SNE and NNP. This method could be adapted to Transformer-DR, but has yet to be.

Furthermore, Transformer-DR is not easily generalisable beyond image data. In contrast, our approach must be designed to extend to a broader

class of high-dimensional, real-valued datasets.

3. Methodology

This chapter describes the design choices and evaluation methodology for the PAT model for supervised dimensionality reduction. It begins by presenting the conceptual reasoning behind replacing traditional feed-forward neural networks with a transformer-based architecture in section 3.1, followed by a detailed explanation of the architectural design in section 3.2. The chapter then addresses interpretability through attention visualisation in section 3.3, and concludes with an overview of the datasets used in this study in section 3.4.

The PAT model replaces feed-forward neural networks with a transformer-based design, allowing each sample to attend to others and capture both global and local structures in the projected space. Attention mechanisms learn multiple types of inter-sample relationships without predefined neighbourhood parameters, as discussed in section 3.1. Preliminary experiments on MNIST guided the choice of the three-layer Modified-Encoder configuration as the most effective design, which is further explained in section 3.2.

A key focus of the methodology is interpretability. Attention weights are used to understand which samples influence each other, both globally and locally. Global patterns are analysed using histograms of attention scores across layers and heads, while local patterns are visualised for selected samples by scaling and greying points according to attention weight, as described in section 3.3. This two-step approach provides insight into how the model distributes attention across the dataset.

Finally, the chapter introduces the six datasets selected to cover a range of data types, dimensionalities, sparsity levels, and manifold complexities, providing a representative benchmark for evaluating the model's performance. Details on these datasets and their preprocessing are provided in section 3.4.

3.1 Concept

Attention-based methods have the potential to fundamentally improve projection techniques by explicitly modelling interactions between samples. Traditional neighbourhood-preserving Dimensionality Reduction methods such as t-SNE and UMAP capture inter-sample relationships in different ways. t-SNE converts pairwise distances into probabilities, while UMAP constructs a graph to preserve local topological structure. Notably, neither method processes samples in isolation. Building on this concept, methods like kNNP have demonstrated that considering multiple samples simultaneously can enhance projection quality. However, kNNP is limited to a single type of relationship (i.e., closeness) and depends on a hyperparameter to define the number of

neighbours. In contrast, attention mechanisms are capable of learning multiple types of relationships concurrently without requiring an explicit k-nearest neighbours parameter, thereby offering a more flexible and comprehensive approach to capturing inter-sample interactions. Furthermore, this concept potentially extends to inverse projection.

Therefore, we propose fully replacing the feed-forward neural networks from NNP with an attention-based architecture. NNP processes each sample independently, it does not explicitly model interactions between points. In contrast, our attention-based approach maintains permutation invariance while allowing each sample to attend to all other samples, ideally focusing more on its nearest neighbours and capturing inter-sample relationships. This allows the model to project the dataset collectively, modelling inter-sample relationships, rather than projecting samples independently like NNP.

3.2 Architectural Design

Similar to NNP, PAT is structured as a single task inference model. Its singular objective is to learn $P : \mathbb{R}^n \to \mathbb{R}^q$, where $\mathbb{R}^n$ is the original high-dimensional data, and $\mathbb{R}^q$ is the lower-dimensional representation (i.e. projection) from a traditional DR method (e.g. t-SNE).

Two architecture variants come to mind when using attention for supervised Dimensionality Reduction. We will refer to these as Attention-Only and Modified-Encoder. The former of these would be the computationally lightest option, where only Multi-Head attention is used for projection. Here, the W^O matrix of learnable parameters would be responsible for projection. Multiple Multi-Head attention layers may be stacked, and these layers may be followed by layer normalisation [42].

The Modified-Encoder refers to a modified version of the Transformer Encoder. Here, the position-wise fully connected feed-forward network (FFN) will be responsible for projection. I.e., $\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$, where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$, and $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_q}$ This modification would require the removal of the last residual connection in the last layer, as the dimensions of the data exiting the FFN would mismatch with d_{model}. This will be more computationally expensive, but allows for the benefits of the FFN and (one of) the residual connections per encoder layer. Additionally, the second Layer Normalisation layer must be removed, as it disables the model's ability to learn projections.

As generalised Projection with ATtention (PAT), i.e. not specialised on computer vision, is a novel concept, we have little idea which configuration of parameters is reasonable for this application. We address this by conducting preliminary experiments on MNIST. Herein, we non-exhaustively explored a very wide search space for the two configurations and their (hyper-) parameters.

The Modified-Encoder group of models consistently outperformed the Attention-

Only models on all metrics, particularly at a depth of three Modified-Encoder layers. Therefore, we have chosen to continue exploring the three-layer Modified-Encoder configuration. See subsection 4.2.1 and subsection 4.2.2 for more details on Architectural Design.

3.3 Interpretability through Attention Visualisation

A secondary research question concerns model interpretability and explainability. Attention itself has been used as a form of explainability in NLP since the conception of Transformers [4], [44], [45]. Extending these techniques to projection with attention could provide insight into how the PAT models function, and how to improve them.

Attention weights, as can be seen in Equation 3.1, model how attention is distributed.

$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{Attention Weight}(Q, K)V \\ \text{where Attention Weight}(Q, K) &= \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \end{aligned} \tag{3.1}$$

Existing visualisation tools, such as BertViz [45], [46], show which tokens each attention head focusses on in NLP tasks. They do this by drawing lines from the source sample to all the samples in the sequence. The opacity and thickness of the line are determined by the attention weight. This can be seen in Figure 3.1.

While these tools provide insight for sequences of manageable length, directly applying them to projections is not practical due to the large number of samples and the way attention is typically distributed. Lines connecting all source and target samples quickly become unintelligible.

For this reason, we focus on a two-step visualisation and analysis of attention patterns. We start by examining global attention patterns. For every layer and every head, the attention weights are calculated for all samples in the test set. Their distributions are visualised using log-scaled histograms. This gives insight into how attention is allocated across the dataset, and how the attention patterns evolve from layer to layer.

The second analysis examines local attention patterns, i.e. for individual samples. All samples in the projected space are visualised. The proposed visualisation technique is inspired by BertViz's use of varying opacity and thickness of the lines by attention weight. Instead of varying the opacity and thickness of the lines, we vary the thickness and saturation of the points themselves.

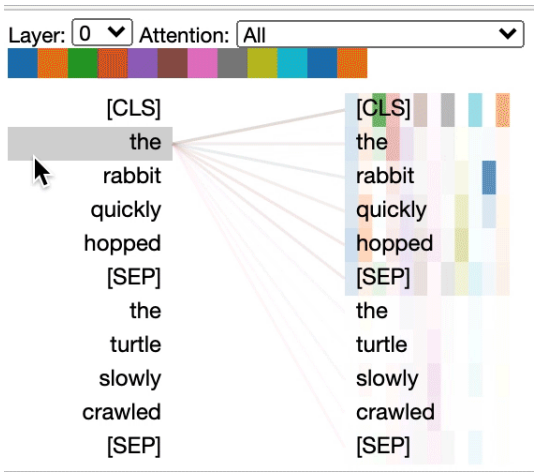


Figure 3.1: 'Head view' taken from BertViz[45], [46], showing what individual heads, as indicated by colour, in layer 0 attends to for the word 'the'. This isn't directly portable to projections, as the number of samples is far too big.

Samples with negligible attention are greyed out. The remaining points are scaled according to their attention weight. This highlights which neighbourhoods or clusters each head considers important for that particular sample and allows comparison of attention behaviour between heads and layers.

3.4 Datasets

Espadato et al proposed a set of 18 datasets for benchmarking dimensionality reduction algorithms [11]. They based their selection to be representative on five criteria: Type, Size, Dimensionality, Intrinsic Dimensionality, and sparsity. Due to computational limitations, we must limit this to fewer datasets.

Specifically, we have chosen to restrict it to the following datasets:

1. modified National Institute of Standards and Technology Database (MNIST) (subsection 2.3.1) [14]
2. Fashion-MNIST (fMNIST) (subsection 2.3.2) [15]
3. Canadian Institute For Advanced Research (CIFAR-10) (subsection 2.3.3) [33]
4. Human Action Recognition (HAR) (subsection 2.3.4) [34]
5. Spambase (subsection 2.3.5) [35]
6. National Classification of Economic Activities (Classificação Nacional de Atividades Econômicas) (CNAE-9) (subsection 2.3.6) [36]

This list of six datasets should reasonably cover the minimal expected variation of datasets one would encounter. The first dataset on this list, MNIST, was not on the original list of 18 [11]. However, we have chosen to include it for historical comparisons, as it features heavily in projection research (e.g.

[13], [19]).

Beyond historical relevance, these six datasets collectively span a broad spectrum of data characteristics while remaining computationally feasible. In terms of image data, they provide a controlled progression of visual complexity. MNIST, although relatively simple, serves as a widely recognised baseline, and its inclusion enables sanity checking against the extensive body of existing dimensionality reduction literature.

fMNIST preserves the same input dimensionality as MNIST, but introduces more challenging class boundaries and richer textures. This ensures that differences in performance arise from increased data complexity rather than changes in input shape. CIFAR-10 further increases difficulty by incorporating colour channels, higher intrinsic variability, and substantial visual noise and texture. This makes it an effective stress test for the ability of methods to preserve global structure. For instance, t-SNE is known to struggle on CIFAR-10 and does not typically produce clearly separable clusters.

The tabular text-representation datasets, CNAE-9 and Spambase, fill in both sparsity extremes. CNAE-9 is extremely sparse and therefore challenges algorithms that are sensitive to high-dimensional flat directions or sparsity patterns. Spambase, in contrast, is dense and more representative of classical tabular feature sets. Together, these datasets span the range of sparsity conditions commonly encountered in non-image data. They also differ in intrinsic dimensionality. CNAE-9 contains many near-irrelevant or low-weight dimensions, whereas Spambase exhibits a more uniformly distributed signal across features.

Finally, HAR introduces a distinct data modality: multivariate time-series measurements obtained from wearable sensors. Unlike images or text-derived representations, HAR data encode temporal correlations that reflect human movement patterns. Such sequences often reside on curved or cyclic manifolds, which makes HAR a valuable benchmark for assessing the robustness of dimensionality reduction methods when the underlying geometry differs substantially from spatial or textual structure.

Across these six datasets, we therefore cover a representative range of *types* (image, tabular, and sequential data), sparsity levels, dimensionalities, intrinsic dimensionalities, and problem difficulties. This includes a progression from simple to challenging image datasets, sparse and dense forms of tabular data, and a non-image, non-text sequential dataset. These datasets also span a range of sample sizes and feature dimensionalities. While not as exhaustive as the full set of eighteen proposed by Espadoto et al., this selection covers the essential variation required for our study. A more detailed discussion of dataset sizes, the last criterion, is provided in the section 4.1 of the Experimental Setup chapter.

4. Experimental Setup

This chapter presents the experimental setup used to evaluate the proposed transformer-based dimensionality reduction approach, PAT, across multiple datasets. It first describes the preprocessing pipeline in section 4.1, then details how reasonable model parameters were identified and refined in section 4.2, followed by the setup for comparative evaluation in section 4.3 and the analysis of scalability in section 4.4. Finally, the chapter outlines how attention visualisation is used to investigate interpretability in section 4.5 and implementation considerations in section 4.6.

Datasets are preprocessed uniformly as described in section 4.1, where samples are flattened to remove spatial or temporal structure, min-max scaled, and projected with t-SNE as a reference. Training and test sets are stratified to preserve label distributions, and training sets are generally smaller than test sets to emulate the NNP-style scenario in which models encounter unseen data.

Reasonable model parameters were determined through a wide preliminary search on MNIST presented in subsection 4.2.1, which explored architectural and training parameters including layer depth, attention heads, and learning rate schedules. This search demonstrated that three-layer Modified-Encoder (MEnc) models consistently outperform Attention-Only variants. The results of this exploration informed the refinement of four final MEnc variants and one additional preprocessing modification, as described in subsection 4.2.2, while optimisation settings and hyperparameters were fixed to reduce evaluation complexity.

Comparative evaluation is conducted on all six datasets across multiple training set sizes, as outlined in section 4.3, allowing analysis of projection quality, generalisability, and sample efficiency. Convergence is examined in detail on MNIST, as shown in subsection 4.3.2, to determine stability and appropriate epoch counts for training.

Scalability experiments, detailed in section 4.4, test the effect of varying sample sizes and input dimensions under both batched and unbatched training, providing insight into the practical limitations of the models. The chapter also investigates interpretability through attention visualisation in section 4.5, showing how PAT distributes focus across data points.

Finally, the chapter documents implementation considerations in section 4.6, including software choices and memory optimisations, to ensure reproducibility and practical feasibility. Together, these sections establish a structured framework for designing, evaluating, and understanding PAT, supporting the analyses and conclusions presented in later chapters.

4.1 Datasets

Each dataset goes through the same data processing pipeline. The only parameters for which are the parameters used for t-SNE, and the number of train- and test samples.

For each dataset:

1. Each sample is flattened row-major order style [50]. This destroys any inherent spatial or temporal sequences. For example, an image of size $w \times h$ is reshaped into a vector of length $w \cdot h$. This is intentional, as one of the requirements of NNP class methods is that they are generally applicable. This is potentially not the best way to flatten the HAR dataset, as this dataset is made from multiple subjects and tasks, causing projections to have clusters with subclusters. This is intentional, as it results in a good test for the recreation fidelity of local structure.
2. The data is min-max scaled, so it falls in the range [0..1]. This is practically a requirement for training neural networks, but seemingly has no impact on t-SNE. Using min-max scaling without standardisation does leave this pipeline vulnerable to outliers, although this was not an issue with these datasets.
3. The data is projected using t-SNE[51], with the parameters as used by [22], [52].
4. The projection is also min-max scaled. This is practically a requirement for training neural networks.
5. The dataset is train-test split, stratified on the labels, ensuring the train and test sets have similar class distributions. See Table 4.1 for how the datasets are split.

Dataset	# Test Samples	# Train samples	Size Class
MNIST	7000	2000, 3000, 4000, 5000, **6000**	Medium-Large
fMNIST	6000	2000, 3000, **4000**	Medium
CIFAR-10	4000	500, 1000, 2000, **3000**	Small-Medium
HAR	4000	500, 1000, 2000, **3300**	Small-Large
CNAE-9	540	126, 252, **540**	Small
Spambase	2301	350, 700, 1300, **2300**	Small-Medium

Table 4.1: Number of samples in test and train sets per dataset. Default number of training samples in bold.

In NNP-class use cases, the number of samples unseen by the ground truth model typically exceeds the number of samples it has been trained on. To emulate this scenario, we therefore ensure that the training set is always smaller than the test set, except in cases where this is not feasible due to a very limited dataset size, such as with CNAE-9. Additionally, for some datasets, the maximum number of usable samples is constrained by hardware limitations

arising from the cost of computing the evaluation metrics. This restriction applies in particular to MNIST, fMNIST, and CIFAR-10.

Despite the limitations in terms of maximum number of usable samples, these datasets portray the full range of dataset sizes as described by Espadoto et al. [11].

4.2 Model Architecture and Variants

This section outlines how the final set of PAT model architectures was established. Instead of attempting an exhaustive hyperparameter search, which is not feasible given the size of the design space, we performed a broad exploratory study on MNIST and used the results to identify a small number of promising architectural directions. Across roughly 900 experimental runs, this process progressively reduced the search space to a set of configurations that were consistently stable and performant.

Subsection 4.2.1 describes the preliminary search, which identified the *Modified-Encoder (MEnc)* family as the most reliable starting point and established the general optimisation settings used for all further experiments.

Subsection 4.2.2 then details the refinement of architectural parameters within the three-layer MEnc design, which led to four final variants and one additional modification selected for full evaluation across all datasets. These models represent the most practical balance between performance, stability, and computational cost that could be achieved within the constraints of this project.

4.2.1 Preliminary Model Search

The initial search for a reasonable configuration and (hyper-)parameter set, performed on MNIST, was very wide. It explored the following (hyper-)parameters:

- Number of layers: (1-4),
- Number of attention heads per layer: (1-7),
- d_v, d_k per layer: (32-512),
- d_{ff} per layer: (512-2048),
- Loss function: Mean Squared Error (MSE) and ℓ_1,
- Learning rate: (0.00001, 0.01),
- Learning rate scheduler: Cosine Annealing Learning Rate Scheduler [53] or None,
- T_{max} for Cosine Annealing Learning Rate Scheduler: (3, 100)

It is not feasible to search this space comprehensively as it is far too large. Instead, we chose to search this space using a random grid search. This method roughly samples the search space, helping to identify configurations that con-

sistently produce high-quality projections.

Results from this exploratory phase show that Modified-Encoder models consistently outperform Attention-Only models on all metrics, leading to the exclusion of Attention-Only models from further research. Furthermore, three-layer Modified-Encoder models were found to perform better and more consistently compared to smaller and larger models, making it the focus of further parameter search.

The preliminary parameter search also revealed that the MEnc-style models, and by extension the broader family of PAT variants, benefit consistently from using a cosine-annealed learning rate schedule [53]. Based on these observations, the optimisation hyperparameters were finalised prior to the main evaluation phase.

The selected settings were, on average, the most globally stable and performant across datasets, and fixing them allowed subsequent experiments to focus exclusively on architectural differences (e.g. attention dimensions, number of heads). All PAT models are therefore trained using an ℓ_1 loss function, the Adam optimiser [54] with a learning rate of 2.2×10^{-4}, and the CosineAnnealingLR scheduler [53], [55] with $T_{\max} = 20$, following the PyTorch implementation. Each model is trained for 100 epochs.

Memory constraints of the available hardware made non-batched training impractical. Although flash attention prevented out-of-memory errors [56], the resulting training speed was so slow that it failed to meet the scalability requirements of this thesis. Moreover, transformer architectures are known to benefit substantially from batched training[4]. For these reasons, we adopted a batch size of 56, selected through limited preliminary testing. This value is unlikely to be optimal, and future work should include a more systematic exploration of batch-size effects. Unless specified otherwise, it is also important to note that experiments were conducted on aging 2017-era hardware, which ultimately failed after the quantitative evaluation phase, further constraining the scope of experimentation.

Additionally, limited experimentation showed that using batches for inference lowered the projection quality, albeit to a very small degree. Based on this, we chose to run inference and evaluate on the entire test, rather than in smaller batches.

4.2.2 Finalising Model Parameters

The final exploratory phase focused on identifying effective architectural settings for the three-layer Modified-Encoder (MEnc). In particular, the search varied the key and value dimensions (d_k and d_v), the number of attention heads, and the feedforward dimension d_{ff} within each Transformer block.

To analyse the influence of individual and combined parameters, short decision tree regressors with a depth of two to three were trained on the collected experimental results. These trees provided interpretable relationships between

Table 4.2: Final three-layer Modified-Encoder (MEnc) variants and the additional PreprocessedEHEncoder (Pre-EH) modification. Each model consists of three `ModifiedEncoderLayer` blocks; only the key architectural differences are shown.

Variant	Layer	d_k	d_v	d_{ff}	Heads (H)
EarlyHeavyEncoder (EH)	1	256	512	2048	6
	2	256	384	2048	4
	3	256	256	2048	3
PreprocessedEHEncoder (Pre-EH)	Pre-pass Enc.	–	–	– As in EH	
ModeBalancedEncoder (MB)	1	192	256	1024	5
	2	256	256	2048	4
	3	256	256	1024	3
LateBoostEncoder (LB)	1	256	384	1024	4
	2	256	384	2048	4
	3	256	512	2048	4
HighCapEverywhereEncoder (HCE)	1	512	512	2048	6
	2	512	512	2048	6
	3	512	512	2048	6

parameters and model performance, making it possible to identify which configurations were consistently beneficial. Insights from this analysis informed the selection and definition of four final three-layer MEnc variants, which were chosen for comprehensive evaluation across all datasets. These variants are summarised in Table 4.2.

Finally, for datasets on which these variants failed to achieve satisfactory performance, a slight modification was introduced to one of the selected models. This modification adds a single linear layer before the attention layer, responsible for downsampling the high-dimensional input to a fixed representation size of $d_{model} = 784$. This value corresponds to the dimensionality of MNIST, for which earlier experiments had demonstrated that PAT models perform reliably. The modified variant is also included in Table 4.2.

4.3 Comparative Evaluation of PAT Variants

The five MEnc variants introduced in Table 4.2, together with Neural Network Projection (NNP) as a baseline and t-SNE as a ground truth, are evaluated on the six datasets described in section 2.3. Evaluating all models across multiple datasets and across a range of training set sizes provides a comprehensive

assessment of the projection quality and generalisability of the PAT approach.

Each model is trained twice on every dataset and for each of the training-set sizes listed in Table 4.1. This setup enables an analysis of sample efficiency, that is, the number of training samples required for models to achieve adequate projection quality.

Training progress is monitored continuously, and all models are evaluated on the test set at every second epoch to capture their convergence behaviour and to observe when performance stabilises. Although this temporal analysis is recorded for every dataset, a detailed convergence study is conducted only for MNIST, see subsection 4.3.2.

4.3.1 NNP Parameters

NNP serves as the baseline model, as it is the earliest and most widely used supervised neural projection method. Following the (hyper-)parameters recommended by Modrakowski et al. [19], the network consists of three fully connected layers of sizes 600, 240, and 600 with ReLU activations, followed by a sigmoid layer and a final linear projection layer mapping to the target dimensionality. All layers use a small bias value of 0.0001. The model is trained with the Adam optimiser using a learning rate of 0.001 and mean squared error (MSE) loss, as suggested by Modrakowski[19].

We do deviate from one parameter, namely the number of epochs NNP is trained. For the per-dataset analysis, we use 100 epochs rather than following Modrakowski et al.'s Early Stopping rule, to ensure we get a complete overview of NNP's capabilities.

4.3.2 Convergence Analysis on MNIST

In this experiment, we conduct both a quantitative and a qualitative analysis of model convergence on the MNIST dataset. All models are trained on the full MNIST training set (6000 samples), while all projection quality metrics and 2D projections are computed on the test set (7000 samples). All reported values represent the mean over 10 independent runs.

For NNP, we analyse convergence using three fixed training regimes:

1. Early stopping following the rule proposed by Modrakowski et al., typically around epoch 46.
2. Fixed-length training for 100 epochs.
3. Extended training for approximately 200 epochs, representing full convergence.

These regimes are shown in the convergence plots as dotted horizontal lines, each corresponding to the mean performance over 10 runs.

For EH, projection quality metrics are recorded every second epoch throughout training. These projection quality metrics are then plotted individually,

the resulting curves showing the mean value at each recorded time step, together with min-max error bars. These continuous metric traces allow direct comparison to the three NNP regimes in terms of projection quality metrics.

To complement the quantitative results, we also compare the visual structure of the generated projections. For NNP, we again use the three training regimes (50, 100, and 200 epochs). For EarlyHeavyEncoder (EH), as defined in Table 4.2, we select two representative checkpoints, namely 50 and 100 epochs, which align with the comparison points used in the quantitative analysis.

This allows us to identify when, if at all, EH is capable of outperforming the baseline NNP. Which in turn allows us to recommend default hyperparameters for using PAT in the field.

4.3.3 Projection Quality Metrics

All models are evaluated using the eight projection-quality metrics proposed by Machado et al. [24], which were selected for their robustness and low inter-metric correlation. These metrics, described in detail in section 2.2, are:

1. Procrustes Error (M_P)
2. Trustworthiness (M_t)
3. Continuity (M_c)
4. Scale-Normalized Stress (M_σ)
5. Neighbourhood Hit (M_{NH})
6. True Neighbors (M_{TN})
7. Distance Consistency (M_{DC})
8. Pearson Correlation (M_r)

4.3.4 Visual Comparison of Projections

In addition to numerical evaluation, each dataset is also examined visually. The following projections are compared:

1. NNP trained for 100 epochs on the largest available training set,
2. the best and most consistent PAT model (trained for 100 epochs on the largest training set), and
3. the t-SNE ground truth fitted on the full dataset, with only the test split shown.

The best and most consistent PAT model for each dataset is selected primarily on the basis of quantitative metrics, with additional consideration for model reliability. Models that exhibit failures on a dataset outside the controlled data-collection runs of this experiment are excluded from selection, even if their metric performance is competitive. This ensures that the chosen models are both strong performers and stable in practice.

4.4 Evaluating Scalability

Scalability is a main motivation for NNP-class projection algorithms, so we conducted a scalability evaluation for the four final MEnc variants (including the preprocessing variant) and compared them to NNP. We tested two factors: the number of samples and the input dimensionality.

We performed two separate experiments to assess scalability along two axes. In the first experiment, we varied the number of samples while keeping the input dimensionality fixed at 784, mimicking MNIST dimensionality with randomly generated data. In the second experiment, we varied the input dimensionality while keeping the number of samples fixed at 3000. All models were evaluated in inference as well as in both batched and unbatched training, with a batch size of 56 matching that used in the projection-quality evaluation. Due to hardware limitations on the 2017 Windows laptop used for the projection-quality experiments, these scalability tests were run on a different machine with an M5 chip and 24 GB of memory.

The tested sample sizes were 500, 2000, 4000, and 6000, representing small, medium, and large datasets. The tested input dimensionalities were 8, 299, 590, 882, 1173, 1465, 1756, and 2048. These values stay below the dimensionality of CIFAR-10, where the models fail, and are more than twice as large as the dataset with the next largest dimensionality, CNAE-9. We explored a wider range of input dimensions because running these tests is computationally cheaper.

Each configuration was run six times in a row using the same model instance and randomly generated data as input. The first epoch of each run was discarded because all models show startup delays caused by JIT compilation or similar effects. The reported results are the mean of the remaining five epochs. The use of random synthetic data should not affect the performance measurements.

4.5 Interpretability through Attention Visualisation

To investigate how the PAT model distributes attention in practice, the analysis focuses primarily on the MNIST dataset. This choice is motivated by practical time constraints and by the fact that MNIST samples are straightforward to interpret visually. Equivalent visualisations were generated for all datasets and are included in Appendix B, although they are considered only briefly in the main text.

The model which calculates the attention weights is trained on the same hyperparemeters as described in section 4.3. The model used for all datasets is EH, as the most consistently well-performing model. The exception being CNAE-9 and CIFAR-10, which use their best performing PAT model: Pre-EH.

For each dataset, attention weights were computed for every head in all layers. To examine the global behaviour of the model, the distributions of these scores were analysed using log-scaled histograms for each head. This provides an overview of how attention is allocated across the dataset and how this allocation changes between layers. To study the local attention patterns, a set of representative samples was selected from different regions of the projection space. These include samples located within clusters, samples situated on identifiable local structures, and samples appearing in areas of uncertainty or confusion. Only two of these samples are included in the main text, since presenting the full set would require an excessive amount of space.

For the local visualisations, a threshold was applied to identify samples that receive negligible attention. The threshold is defined as the reciprocal of the number of samples, for example 1/7000 for MNIST. The threshold is intended to separate meaningful attention from negligible contributions. Since attention weights sum to 1 across all samples, the reciprocal of the number of samples represents the average attention a sample would receive under uniform allocation. Samples receiving less than this reciprocal are effectively ignored by the model, while those above it are receiving disproportionately high attention. Using this threshold highlights the points that the model focuses on, filtering out background noise and making attention patterns easier to interpret. Samples with attention below this value were greyed out in the figures.

4.6 Implementation Notes

The neural networks were implemented in PyTorch [55] because it is familiar to the author and provides access to memory-efficient implementations of Attention [56] while remaining highly customisable. The previous implementation of projection quality metrics by Machado [27] was developed in TensorFlow and encountered memory limitations on older GPU hardware.

To address this, a memory-optimised variant [57] of a subset of eight metrics was implemented in PyTorch. This approach minimises repeated calculations and manually clears variables when they are no longer needed, allowing experiments on up to 6000 samples on MNIST rather than the previous limit of 3000. Furthermore, it accomplishes this with an 8x speed-up[1] compared to Machado's library for the same metrics.

There are two limitations for this implementation. The first limitation of this approach is that all metrics must use the same value of K for nearest neighbors. Although choosing different K values for each metric may be more optimal, K was set to seven based on recommendations for Continuity and Trustworthiness [11].

The second limitation is that it is not as flexible as Machado's library[27].

[1]Tested on a 2017 Windows Laptop, our implementation on CPU, Machado's on GPU.

Our implementation is specialised on those eight metrics, and would require modification to allow for other metrics. Machado's library supports 17 metrics and any subset thereof.

5. Results

This chapter presents the experimental evaluation of the proposed PAT models across multiple datasets, comparing their performance to the baseline NNP and ground truth t-SNE projections. The results are organized to highlight projection quality, convergence behavior, scalability, and model interpretability through attention patterns.

We begin by assessing projection quality, as described in section 5.1, across six datasets: MNIST, fMNIST, Spambase, HAR, CIFAR-10, and CNAE-9. Performance is evaluated quantitatively using metrics that capture both global and local embedding properties, and qualitatively through visualisations of the learned projections. This approach highlights differences in cluster structure, diffusion, and fidelity to the ground truth. Across most datasets, PAT models outperform NNP once sufficient training samples are available, although some datasets reveal limitations in stability or local neighborhood preservation.

Next, section 5.2 analyzes convergence behavior on MNIST, tracking how projection quality evolves over training epochs for NNP and EH. This comparison demonstrates trade-offs between global structure and neighborhood preservation, showing that PAT models can achieve competitive or superior performance with fewer epochs.

Section 5.3 examines scalability, evaluating runtime with respect to input dimensionality and dataset size. These experiments provide practical insight into training and inference efficiency. While NNP remains substantially faster, several PAT variants achieve competitive projection quality within reasonable computational limits.

Finally, section 5.4 explores model interpretability through attention visualisations, illustrating how PAT models distribute focus across data points. Early layers capture global structure, intermediate layers refine local details, and the final layer exhibits highly selective attention patterns. Although these results do not yield fully interpretable rules, they provide insight into how the models balance global and local information when generating embeddings.

Together, these analyses establish the capabilities, limitations, and operational characteristics of the PAT models, providing a foundation for the discussion in the following chapter.

5.1 Projection Quality

This section presents the projection quality results for all PAT models across the six evaluation datasets, compared against the NNP baseline and the ground truth t-SNE embeddings. Overall, the PAT models show consistent but mod-

erate improvements: on five of the six datasets, they outperform NNP on most quantitative metrics, and the visual projections suggest slightly clearer structure than NNP across all datasets. However, the gains are not dramatic, and in several cases, the differences are subtle or dependent on the chosen metric or training size.

5.1.1 MNIST Dataset

Table 5.1 shows that, from the minimum number of samples tested (2000) and onwards, a PAT model outperforms the baseline NNP on all metrics except for Procrustes Error (M_P), in which NNP is always marginally better.

Among the PAT models, HCE is the best performer at the minimal number of samples, but is then supplanted by EH on all metrics except for Pearson Correlation (M_r), where LB and MB marginally outperform them.

HCE shows stability issues at the maximum number of samples (6000), where its performance in all metrics plummets.

The PAT models perform similarly compared to their ground truth t-SNE on most metrics. They marginally outperform it on M_c, M_{DC} and M_r, and marginally underperform it on M_σ. Compared to t-SNE, PAT significantly underperforms on M_t, M_{NH}, and most importantly, M_{TN}. It does, however, show a small but significant increase in these metrics compared to NNP.

EH needs only 4000 training samples to gain better or equivalent projection quality compared to NNP on 6000 samples.

In Figure 5.1, we chose to visualise the overall best and most consistent PAT model for MNIST: EarlyHeavyEncoder (EH) trained on 6000 training samples executed on 7000 test samples. We compare it to NNP trained and executed on the same samples, and its ground truth t-SNE fitted on the combination of the two, though only the test set is shown.

Visually, the clusters are globally somewhat less diffuse compared to NNP. The shapes, or outlines, of the clusters are also better retained from t-SNE. For example, look toward cluster 1 (orange). EH has a much cleaner 'banana' shape, more closely matching t-SNE. Cluster 0 (dark blue) is far less diffuse compared to NNP, with far fewer outliers above it and to its left. The difficult case of class 4 (purple), which overlaps with class 9 (cyan), causing class 4 to be separated into two subclusters, is also handled marginally better by EH. As the edges of both the larger subcluster at the bottom and the smaller subcluster at the top are slightly better preserved.

Overall, there is still a presence of inter-cluster diffusion, but the global outline, i.e. where clusters do not touch in the ground truth, has little to no diffusion.

Table 5.1: Metrics on MNIST at the smallest, median, and largest training-set sizes. t-SNE is trained on all samples (train + test), metrics are calculated exclusively on the test set. Values in **bold** indicate the best score among all evaluated models, while **underscored and bold** values indicate that the ground truth (t-SNE) achieves a better score than every tested model.

Metric	t-SNE	NNP	EH	HCE	LB	MB
M_t@2000		0.8518	0.8601	**0.8694**	0.8503	0.8558
M_t@4000		0.8865	**0.8980**	0.8957	0.8858	0.8877
M_t@6000		0.9008	**0.9121**	0.7385	0.9057	0.9071
M_t	**0.9822**					
M_c@2000		0.9564	0.9644	**0.9662**	0.9622	0.9634
M_c@4000		0.9658	**0.9711**	0.9708	0.9695	0.9694
M_c@6000		0.9691	**0.9728**	0.7503	0.9720	0.9725
M_c	0.9715					
M_σ@2000		0.1564	0.1562	**0.1557**	0.1566	0.1559
M_σ@4000		0.1550	0.1548	**0.1546**	0.1553	0.1551
M_σ@6000		0.1554	0.1546	0.3142	**0.1543**	0.1544
M_σ	**0.1540**					
M_{NH}@2000		0.7214	0.7558	**0.7632**	0.7403	0.7437
M_{NH}@4000		0.7884	**0.8171**	0.8083	0.7979	0.7983
M_{NH}@6000		0.8121	**0.8371**	0.4998	0.8255	0.8292
M_{NH}	**0.9132**					
M_{TN}@2000		0.0643	0.0796	**0.0855**	0.0719	0.0746
M_{TN}@4000		0.0953	**0.1168**	0.1141	0.1058	0.1089
M_{TN}@6000		0.1173	**0.1351**	0.0686	0.1286	0.1287
M_{TN}	**0.3784**					
M_{DC}@2000		0.7708	**0.7966**	0.7948	0.7836	0.7815
M_{DC}@4000		0.8187	**0.8332**	0.8285	0.8208	0.8221
M_{DC}@6000		0.8357	**0.8470**	0.4789	0.8419	0.8409
M_{DC}	0.8422					
M_P@2000		**0.9919**	0.9945	0.9944	0.9943	0.9945
M_P@4000		**0.9945**	0.9949	0.9948	0.9949	0.9948
M_P@6000		**0.9947**	0.9950	0.9975	0.9949	0.9950
M_P	0.9941					
M_r@2000		0.4220	0.4398	0.4399	**0.4443**	0.4401
M_r@4000		0.4383	0.4407	0.4421	0.4425	**0.4434**
M_r@6000		0.4354	0.4392	0.2181	**0.4422**	0.4402
M_r	0.4424					

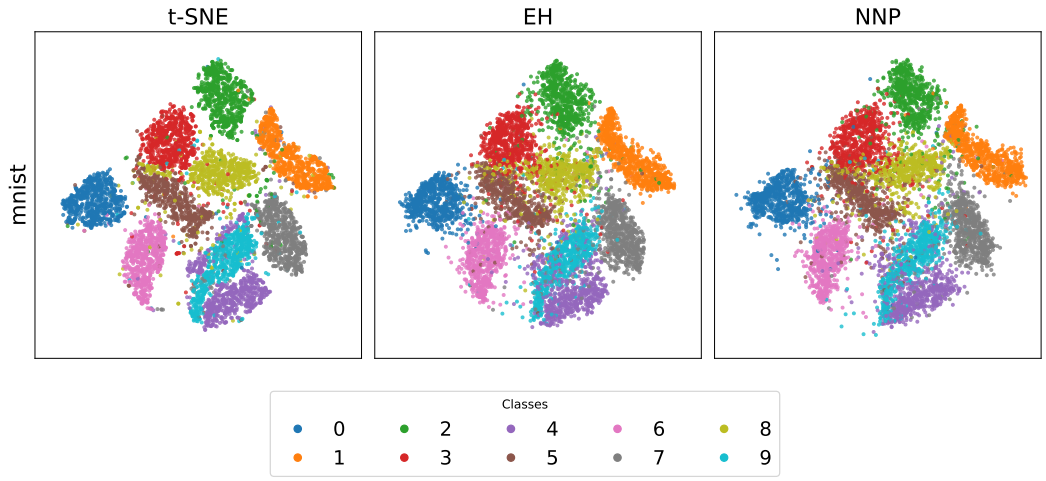


Figure 5.1: EH, NNP, and t-SNE (Ground Truth) on MNIST.

5.1.2 fMNIST Dataset

Table 5.2 shows that PAT and NNP are effectively equivalent for the majority of the metrics on fMNIST. Across training sizes, the NNP baseline performs best at the smallest sample size (2000), consistently achieving the strongest scores in M_σ, M_P, and M_r. However, as soon as more training data becomes available (from 3000 samples onward), the PAT models begin to outperform NNP on metrics like M_t, M_c, M_{NH}, and M_{TN}. Although NNP remains marginally superior in M_σ, M_P, and M_r, these differences are very small and practically negligible.

Compared to the ground-truth t-SNE, all learned models achieve similar performance on M_σ, M_{DC}, M_r, and M_P, while underperming substantially on M_{NH} and M_{TN}, where t-SNE performs markedly better. Overall, the PAT models become competitive or superior to NNP once sufficient training samples (3000 or more) are available, while still falling short of t-SNE on neighborhood-based metrics.

At 3000 samples, the PAT models outperform NNP on the maximum number of training samples (4000) on all but three metrics. Namely M_σ and M_r, in which NNP is marginally better, and M_P in which the models are equivalent.

Table 5.2: Metrics on fMNIST at the smallest, median, and largest training-set sizes. t-SNE is trained on all samples (train + test), metrics are calculated exclusively on the test set. Values in **bold** indicate the best score among all evaluated models, while **underscored and bold** values indicate that the ground truth (t-SNE) achieves a better score than every tested model.

Metric	t-SNE	NNP	EH	HCE	LB	MB
M_t@2000		**0.9537**	0.9510	0.9527	0.9487	0.9486
M_t@3000		0.9563	0.9576	**0.9608**	0.9551	0.9567
M_t@4000		0.9584	0.9621	**0.9627**	0.9594	0.9599
M_t	**0.9890**					
M_c@2000		**0.9838**	0.9833	0.9836	0.9824	0.9827
M_c@3000		0.9834	**0.9852**	0.9851	0.9842	0.9844
M_c@4000		0.9831	**0.9863**	0.9862	0.9855	0.9856
M_c	**0.9871**					
M_σ@2000		**0.1093**	0.1157	0.1156	0.1158	0.1155
M_σ@3000		**0.1100**	0.1145	0.1145	0.1149	0.1146
M_σ@4000		**0.1128**	0.1141	0.1138	0.1142	0.1143
M_σ	0.1126					
M_{NH}@2000		0.6286	0.6477	**0.6488**	0.6458	0.6447
M_{NH}@3000		0.6416	0.6500	**0.6561**	0.6513	0.6543
M_{NH}@4000		0.6451	0.6570	0.6565	**0.6590**	0.6580
M_{NH}	**0.7401**					
M_{TN}@2000		0.1073	0.1189	**0.1251**	0.1153	0.1177
M_{TN}@3000		0.1192	0.1363	**0.1404**	0.1301	0.1335
M_{TN}@4000		0.1266	0.1487	**0.1499**	0.1418	0.1443
M_{TN}	**0.3730**					
M_{DC}@2000		0.6522	0.6657	0.6639	**0.6663**	0.6637
M_{DC}@3000		0.6564	0.6645	0.6665	**0.6672**	0.6634
M_{DC}@4000		0.6619	**0.6657**	0.6647	0.6644	0.6654
M_{DC}	0.6647					
M_P@2000		**0.9973**	0.9976	0.9976	0.9975	0.9975
M_P@3000		**0.9975**	0.9977	0.9977	0.9977	0.9977
M_P@4000		**0.9977**	0.9978	0.9978	0.9977	0.9978
M_P	0.9979					
M_r@2000		**0.6841**	0.6495	0.6490	0.6490	0.6495
M_r@3000		**0.6766**	0.6557	0.6544	0.6539	0.6540
M_r@4000		**0.6647**	0.6575	0.6590	0.6569	0.6566
M_r	0.6649					

One aspect that was not captured in the sampling used to construct Table 5.2 is HighCapEverywhereEncoder (HCE)'s propensity to stability issues on fMNIST. It was rerun twice for visualisation purposes, and both runs failed. We did not test this often enough to give a proper indication for a failure rate; however we choose to err on the side of caution and use EarlyHeavyEncoder (EH) for Figure 5.2 instead, which has not shown this same propensity while performing very similarly to HCE's best case.

In Figure 5.2, both EH and NNP are trained on the same 4000 training samples, and executed on 6000 test samples. The ground truth t-SNE is fitted on the combination of the two, though only the test set is shown.

Visually, PAT much closer emulates the shapes of the clusters in the ground truth. It is noticeably less diffuse compared to NNP, but still more diffuse than t-SNE. Cluster 1 (orange) shows particularly marked improvement over NNP. The outline of the cluster in EH's projection is more sharply defined and captures the hole-y structure of the cluster better. Cluster 2 (lime green) is less diffuse, and its outline more closely follows the ground truth.

In the ground truth, there is a large amount of confusion surrounding classes 0 (dark blue), 2 (green), 3 (red), 4 (purple), and 6 (pink). Both EH and NNP clearly struggle with this. However, EH is marginally better at separating these classes, and more accurately recreates the (sub-) clusters.

Overall, EH has reduced inter-cluster diffusion, better recreation of local and global cluster features, and little to no diffusion of the global outline, i.e., where clusters do not touch in the ground truth.

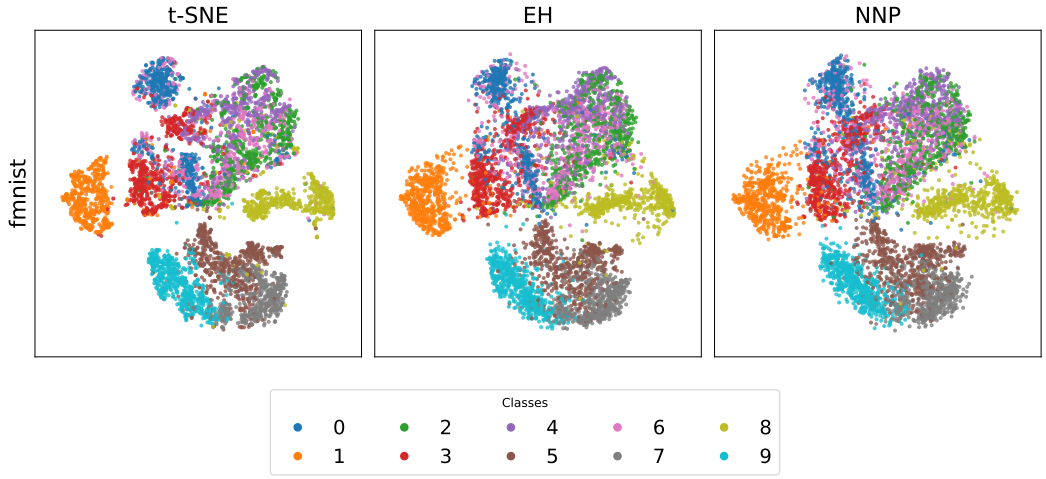


Figure 5.2: EH, NNP, and t-SNE (Ground Truth) on fMNIST.

5.1.3 Spambase Dataset

Table 5.3 shows that there is no clear best model at the minimum number of training samples tested (350). NNP outperforms the PAT models in M_σ, M_r, and importantly M_{TN}, while most PAT models outperform NNP in M_t, M_c, M_{NH}, M_{DC}, and M_P. In general, the PAT models match or outperform NNP from 1300 samples and onwards. The most notable increase is on M_{TN}, while losing its advantage on M_P (Table 5.3).

Notably, some metrics show non-monotonic behaviour with additional training data. Several models exhibit clear declines as sample size increases. For instance, NNP shows steadily worsening M_σ and fluctuating M_P, and it also

displays non-monotonic behaviour in M_{TN}, where performance improves at 1300 samples but then decreases slightly at 2300 samples. Other metrics similarly degrade with more samples.

NNP, EH, and LB all show non-monotonic trends in M_{NH}, M_{DC}, or M_r, with M_r in particular worsening for almost all models. The strongest instability occurs for Pre-EH, which experiences a substantial drop at 2300 samples across several metrics, including M_t, M_c, M_{NH}, M_{TN}, and M_{DC}. These patterns indicate that increasing the amount of training data does not necessarily yield better or more stable embedding quality.

The PAT models perform similarly compared to their ground truth t-SNE on several metrics. They outperform NNP on most metrics from 1300 training samples onward and already surpass it on M_t, M_c, M_{NH}, and M_{DC} at smaller sample sizes. Compared to t-SNE, the PAT models still significantly underperform on M_t, M_{NH}, and most notably M_{TN}. Overall, the PAT models provide a small improvement over NNP, though underperform compared to t-SNE on neighbourhood-based metrics.

In Figure 5.3, we chose to visualise EH, as it performs better on M_{NH} and M_{DC}, is only slightly worse on M_{TN}, and is similar on other metrics compared to HCE.

The ground truth t-SNE projection for Spambase is diffuse in nature, with the classes spam and not spam overlapping significantly, making this a particularly challenging dataset to project. We compare EH trained on 2300 samples and executed on 2301 samples to NNP trained on the same data, with the ground truth t-SNE fitted on all samples, though only the test set is shown.

Visually, EH produces a projection that is closer to the ground truth than NNP, placing fewer points outside the main body of the projection and appearing more centered. It does a far better job at retaining the global outline of the projection, whereas NNP appears more diffuse overall. Both models struggle with the 'antenna' shape at the top of the projection; while neither reproduces the circular dot-like feature at its tip, EH is less diffuse, whereas NNP correctly positions the rightmost dot while EH places it slightly lower.

Overall, the global distribution of points is better retained by EH, with less diffusion outside the main clusters, even though the overlap between classes remains challenging.

Table 5.3: Metrics on Spambase at the smallest, median, and largest training-set sizes. t-SNE is trained on all samples (train + test), metrics are calculated exclusively on the test set. Values in **bold** indicate the best score among all evaluated models, while **underscored and bold** values indicate that the ground truth (t-SNE) achieves a better score than every tested model.

Metric	t-SNE	NNP	EH	HCE	LB	MB	Pre-EH
M_t@350		0.7587	0.7546	**0.7636**	0.7510	0.7515	0.7548
M_t@1300		0.7657	0.7844	0.7881	0.7857	0.7827	**0.8049**
M_t@2300		0.7920	0.8047	0.8027	**0.8048**	0.8031	0.5783
M_t	**0.9380**						
M_c@350		0.9341	0.9347	0.9353	0.9311	0.9316	**0.9367**
M_c@1300		0.9341	0.9409	0.9429	0.9394	0.9408	**0.9454**
M_c@2300		0.9399	0.9448	**0.9455**	0.9441	0.9434	0.6733
M_c	**0.9467**						
M_σ@350		**0.2356**	0.2631	0.2595	0.2699	0.2688	0.2412
M_σ@1300		**0.2386**	0.2641	0.2643	0.2655	0.2650	0.2535
M_σ@2300		**0.2484**	0.2620	0.2653	0.2617	0.2626	0.4380
M_σ	0.2543						
M_{NH}@350		0.7658	0.7816	0.7713	0.7780	0.7726	**0.7896**
M_{NH}@1300		**0.7844**	0.7805	0.7771	0.7757	0.7775	0.7834
M_{NH}@2300		0.7717	**0.7853**	0.7816	0.7852	0.7816	0.6142
M_{NH}	**0.8380**						
M_{TN}@350		**0.1689**	0.1508	0.1564	0.1542	0.1573	0.1245
M_{TN}@1300		0.1683	0.1836	0.1837	0.1862	0.1809	**0.1863**
M_{TN}@2300		0.1858	0.2024	**0.2062**	0.2056	0.2012	0.0087
M_{TN}	**0.4694**						
M_{DC}@350		0.7990	0.8209	0.8129	0.8038	0.8025	**0.8375**
M_{DC}@1300		**0.8125**	0.8038	0.8003	0.8038	0.8066	0.8094
M_{DC}@2300		0.7975	**0.8073**	0.8023	0.8014	0.8025	0.7025
M_{DC}	0.7944						
M_P@350		0.9487	0.9483	0.9486	**0.9471**	0.9472	0.9500
M_P@1300		**0.9391**	0.9487	0.9492	0.9489	0.9488	0.9503
M_P@2300		**0.9471**	0.9495	0.9496	0.9493	0.9494	0.9636
M_P	0.9496						
M_r@350		**0.4460**	0.3659	0.3707	0.3453	0.3498	0.4292
M_r@1300		**0.4301**	0.3540	0.3545	0.3516	0.3522	0.3863
M_r@2300		**0.3981**	0.3593	0.3520	0.3603	0.3587	0.3972
M_r	0.3751						

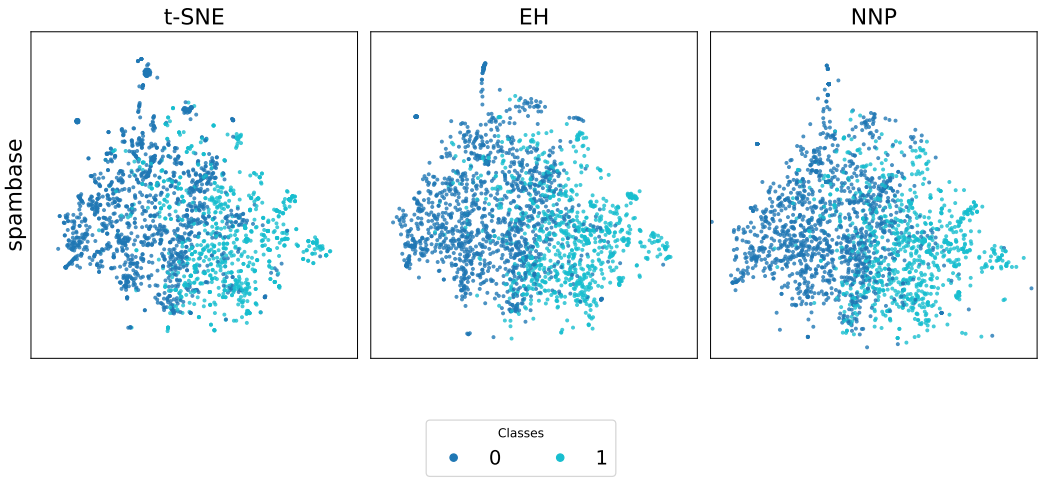


Figure 5.3: EH, NNP, and t-SNE (Ground Truth) on Spambase.

5.1.4 HAR

Table 5.4 shows that the PAT models are equivalent or better than NNP. This pattern holds for all numbers of training samples used. Between the PAT models, HCE marginally outperforms EH, both of which typically outperform the others.

At 2000 training samples, Pre-EH outperforms the others on the metrics M_t, M_{NH}, M_{TN}. However, it does show stability issues at the maximum number of samples (3300). Here, the metric scores plummet.

Compared to the ground truth t-SNE, both NNP and PAT grossly underperform on M_{TN} and M_{NH}, while performing similarly in all other metrics.

Of the PAT models HCE is the overall best performer. Additionally, it does not show the stability issues for the HAR dataset as it has for others. Therefore, we have chosen to visualise this model's projections in Figure 5.4.

In Figure 5.4, both HCE and NNP are trained on the same 3300 training samples, and executed on 4000 test samples. The ground truth t-SNE is fitted on the combination of the two, though only the test set is shown.

The HAR projection offers very fine local features. It offers clear intercluster separation (classes 0, 1, 2, 5) and very clear but small intracluster separation (subclusters). It simultaneously offers a very diffuse sector, with the clusters for classes 3 and 4 being interwoven.

Visually, both models struggle and are clearly more diffuse than t-SNE. However, HCE is visibly less diffuse in class 0 (dark blue). It gets closer to approximating the subclusters than NNP, but still clearly falls short compared to t-SNE. HCE better recreates the two class 0 subclusters above class 1 (green). It also better retains the shape of class 5 (light blue), far better recreates the outlines of cluster 2 (purple), and 3 and 4 (lime and pink).

Table 5.4: Metrics on HAR at the smallest, median, and largest training-set sizes. t-SNE is trained on all samples (train + test), metrics are calculated exclusively on the test set. Values in **bold** indicate the best score among all evaluated models, while **underscored and bold** values indicate that the ground truth (t-SNE) achieves a better score than every tested model.

Metric	t-SNE	NNP	EH	HCE	LB	MB	Pre-EH
M_t@500		0.9279	0.9315	**0.9324**	0.9292	0.9288	0.9288
M_t@2000		0.9468	0.9527	0.9517	0.9478	0.9490	**0.9540**
M_t@3300		0.9530	0.9593	**0.9614**	0.9562	0.9555	0.7335
M_t	**0.9896**						
M_c@500		0.9717	0.9750	**0.9757**	0.9737	0.9739	0.9741
M_c@2000		0.9782	0.9809	**0.9811**	0.9797	0.9799	0.9807
M_c@3300		0.9797	0.9824	**0.9829**	0.9819	0.9818	0.7399
M_c	**0.9833**						
M_σ@500		0.1161	0.1121	0.1121	**0.1106**	0.1132	0.1169
M_σ@2000		0.1226	0.1196	**0.1185**	0.1188	0.1192	0.1216
M_σ@3300		**0.1201**	0.1212	0.1206	0.1204	0.1201	0.4525
M_σ	0.1241						
M_{NH}@500		0.7799	**0.7992**	0.7876	0.7903	0.7856	0.7943
M_{NH}@2000		0.8258	0.8370	0.8414	0.8265	0.8327	**0.8458**
M_{NH}@3300		0.8442	0.8568	**0.8649**	0.8512	0.8480	0.5153
M_{NH}	**0.9211**						
M_{TN}@500		0.0842	0.0928	**0.0969**	0.0881	0.0872	0.0878
M_{TN}@2000		0.1273	0.1531	0.1542	0.1378	0.1380	**0.1591**
M_{TN}@3300		0.1497	0.1854	**0.1935**	0.1681	0.1688	0.0968
M_{TN}	**0.4669**						
M_{DC}@500		0.7831	0.7943	**0.7963**	0.7950	0.7911	0.7931
M_{DC}@2000		0.8043	0.8002	0.8044	0.7994	0.8014	**0.8075**
M_{DC}@3300		0.8056	0.8078	**0.8115**	0.8073	0.8056	0.4831
M_{DC}	**0.8125**						
M_P@500		**0.9947**	0.9959	0.9961	0.9956	0.9957	0.9957
M_P@2000		**0.9962**	0.9969	0.9969	0.9968	0.9968	0.9968
M_P@3300		**0.9967**	0.9973	0.9973	0.9972	0.9972	0.9986
M_P	0.9976						
M_r@500		0.6469	0.6590	0.6582	**0.6621**	0.6550	0.6434
M_r@2000		0.6222	0.6315	**0.6352**	0.6346	0.6333	0.6248
M_r@3300		**0.6319**	0.6259	0.6286	0.6291	0.6299	0.3130
M_r	0.6151						

Overall, HCE does a better job at retaining the global characteristics of the projection, and while improving on local features, it still lacks the granularity of t-SNE.

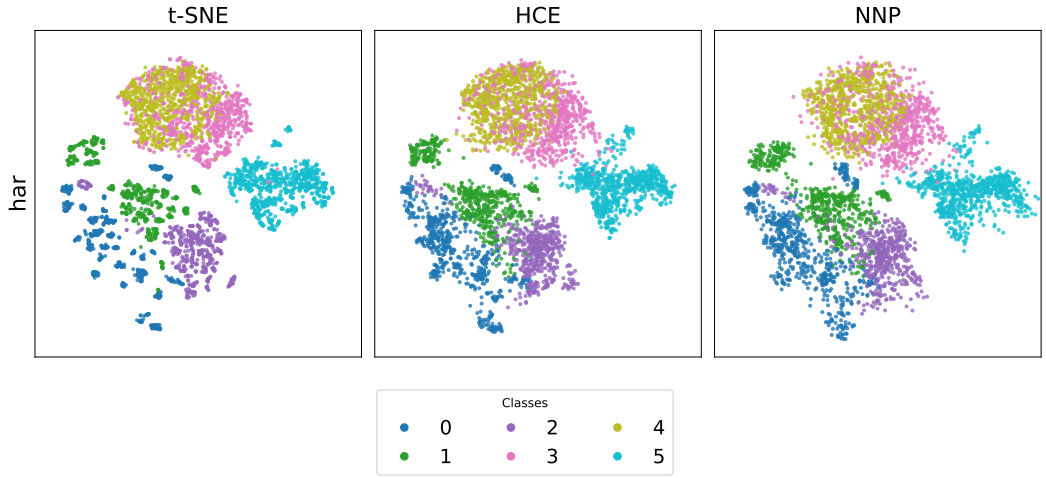


Figure 5.4: HCE, NNP, and t-SNE (Ground Truth) on HAR.

5.1.5 CIFAR-10

All four initially proposed PAT models completely fail in projecting CIFAR-10. Their projections are effectively random, and are thus left out of Table 5.5. The Pre-EH variant, where the EH is modified by prepending a downsampling layer, is able to project CIFAR-10.

CIFAR-10 has very high input dimensionality ($32 \times 32 \times 3 = 3072$), which likely contributes to the failure of the initial PAT models. In the MEnc architecture, residual connections require repeated transformations of the input back to the full model dimension ($d_{\text{model}} = 3072$) through the weight matrices W^O, W_1, and W_2. This results in extremely large weight matrices, making it difficult for the network to learn effective projections. The success of the Pre-EH variant, which reduces the model dimension from 3072 to a more manageable 784 via a linear layer, supports this explanation.

Table 5.5 shows that Pre-EH is equivalent or better in all metrics except for M_r starting at the minimal number of training samples (500). The performance gap between Pre-EH and NNP grows thereafter, showing its performance is superior on CIFAR-10 in almost every way.

Compared to t-SNE, both models perform similarly. Pre-EH shows a small improvement in M_{DC}, indicating better visual separation of classes. There are negligible differences in other metrics, but it does underperform in M_{NH} and M_{TN}.

It is worth noting that Pre-EH does occasionally fail at a higher number of training samples on CIFAR-10. This is not reflected in Table 5.5, as it did not occur at this point in the data gathering process. Additionally, this is not

a big issue, as Pre-EH with 1000 training samples already outperforms NNP at the maximum number of training samples (3000).

Table 5.5: Metrics on CIFAR-10 at the smallest, median, and largest training-set sizes. t-SNE is trained on all samples (train + test), metrics are calculated exclusively on the test set. Values in **bold** indicate the best score among all evaluated models, while **underscored and bold** values indicate that the ground truth (t-SNE) achieves a better score than every tested model.

Metric	t-SNE	NNP	Pre-EH
M_t@500		0.8016	**0.8031**
M_t@2000		0.8049	**0.8126**
M_t@3000		0.8067	**0.8185**
M_t	**0.8905**		
M_c@500		**0.9256**	0.9255
M_c@2000		0.9270	**0.9293**
M_c@3000		0.9285	**0.9294**
M_c	0.9254		
M_σ@500		0.1155	**0.1131**
M_σ@2000		0.1184	**0.1115**
M_σ@3000		0.1134	**0.1113**
M_σ	0.1116		
M_{NH}@500		**0.1441**	0.1437
M_{NH}@2000		0.1496	**0.1554**
M_{NH}@3000		0.1486	**0.1561**
M_{NH}	**0.1999**		
M_{TN}@500		0.0285	**0.0297**
M_{TN}@2000		0.0339	**0.0355**
M_{TN}@3000		0.0336	**0.0417**
M_{TN}	**0.1407**		
M_{DC}@500		**0.2123**	0.2073
M_{DC}@2000		0.2112	**0.2207**
M_{DC}@3000		0.2159	**0.2227**
M_{DC}	0.2142		
M_P@500		**0.9856**	0.9862
M_P@2000		**0.9867**	0.9867
M_P@3000		0.9888	**0.9858**
M_P	**0.9837**		
M_r@500		**0.7426**	0.7173
M_r@2000		**0.7200**	0.7140
M_r@3000		**0.7337**	0.7008
M_r	0.6884		

The projections visualised in Figure 5.5 are from Pre-EH and NNP trained and tested on the same 3000 and 4000 samples, respectively. t-SNE was fitted on the sum of the two, although only the test set is visualised.

When projected with t-SNE, as can be seen in Figure 5.5, CIFAR-10, has

little to no cluster separation. The visual quality of the learned projections (Pre-EH and NNP) can therefore only be judged on the outline of the projection and the level of diffusion surrounding it.

NNP's projection notably does not follow the same shape as those generated with t-SNE. It appears stretched vertically, to the point that points fall outside of the [0, 1] y-axis range. It is also visibly far more diffuse.

Pre-EH recreates the original shape far better. Interestingly, its edges are also far less diffuse compared to t-SNE.

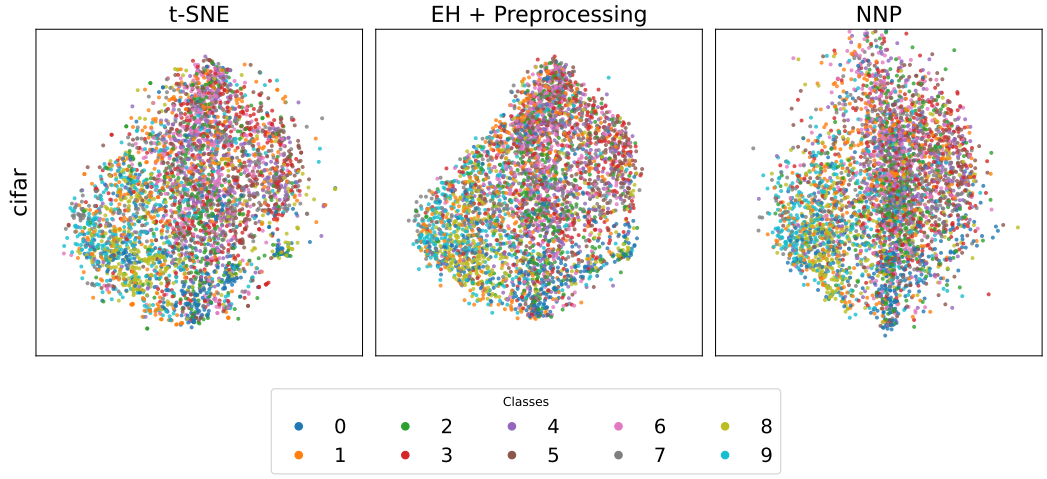


Figure 5.5: Pre-EH, NNP, and t-SNE (Ground Truth) on CIFAR-10.

5.1.6 CNAE-9

All four PAT models are consistently worse than NNP on all metrics except M_P. Using the Pre-EH variant at the maximum number of training samples (540) allows PAT to marginally outperform NNP on M_{TN}, and marginally underperform on other metrics.

Compared to the ground truth t-SNE, both models underperform similarly. Only considerably outperforming t-SNE in M_{DC}.

Table 5.6: Metrics on CNAE-9 at the smallest, median, and largest training-set sizes. t-SNE is trained on all samples (train + test), metrics are calculated exclusively on the test set. Values in **bold** indicate the best score among all evaluated models, while **underscored and bold** values indicate that the ground truth (t-SNE) achieves a better score than every tested model.

Metric	t-SNE	NNP	EH	HCE	LB	MB	Pre-EH
M_t@126		**0.6913**	0.6126	0.6234	0.6198	0.6271	0.6656
M_t@252		**0.6987**	0.6402	0.6348	0.6378	0.6422	0.6722
M_t@540		**0.7076**	0.6739	0.6657	0.6658	0.6660	0.7068
M_t	**0.7928**						
M_c@126		**0.8776**	0.7534	0.7509	0.7492	0.7821	0.8561
M_c@252		**0.8978**	0.8082	0.7854	0.7952	0.8268	0.8623
M_c@540		**0.9006**	0.8467	0.8349	0.8421	0.8533	0.8926
M_c	**0.9110**						
M_σ@126		**0.2663**	0.2850	0.2887	0.2807	0.2818	0.2936
M_σ@252		**0.2484**	0.2811	0.2867	0.2751	0.2852	0.2832
M_σ@540		**0.2451**	0.2775	0.2802	0.2793	0.2785	0.2590
M_σ	**0.2158**						
M_{NH}@126		**0.5292**	0.3685	0.3896	0.3733	0.4082	0.4892
M_{NH}@252		**0.5573**	0.4440	0.4209	0.4275	0.4590	0.5429
M_{NH}@540		**0.5971**	0.5292	0.5146	0.5013	0.5266	0.5942
M_{NH}	**0.6963**						
M_{TN}@126		**0.1713**	0.1081	0.1095	0.1049	0.1085	0.1433
M_{TN}@252		**0.1902**	0.1319	0.1238	0.1262	0.1291	0.1680
M_{TN}@540		0.2140	0.1636	0.1548	0.1612	0.1630	**0.2189**
M_{TN}	**0.3889**						
M_{DC}@126		**0.5759**	0.4269	0.4759	0.4583	0.4843	0.5444
M_{DC}@252		**0.5852**	0.5157	0.4935	0.5074	0.5278	0.5769
M_{DC}@540		**0.6167**	0.5935	0.5861	0.5611	0.5870	0.6148
M_{DC}	0.5481						
M_P@126		0.9824	0.9721	0.9751	**0.9718**	0.9757	0.9857
M_P@252		0.9812	0.9761	**0.9741**	0.9757	0.9785	0.9837
M_P@540		0.9808	0.9778	0.9775	**0.9775**	0.9786	0.9807
M_P	0.9778						
M_r@126		**0.2412**	0.0194	0.0172	0.0482	0.0459	0.1340
M_r@252		**0.2164**	0.0432	0.0180	0.0396	0.0543	0.0982
M_r@540		**0.1858**	0.0469	0.0326	0.0408	0.0473	0.1147
M_r	0.2066						

The PAT model chosen to visualise in Figure 5.6 is Pre-EH, as it is the only model capable of challenging NNP in terms of metrics. Both models are trained and tested on 540 and 540 samples, respectively. The ground truth t-SNE was fitted on the sum of the two, although only the test set is visualised.

As indicated by the poor metrics, this is a difficult dataset to project. The number of samples is low, and the data is highly sparse. t-SNE's projection shows a large amount of diffusion, and few visibly separated clusters.

NNP is very diffuse, to the point that it places clusters outside of the [0, 1] range dictated for the x-axis. Pre-EH notably seems to better retain the shapes of the cluster. Class 7 (lime)'s cluster is poorly identifiable in NNP, whereas Pre-EH places them more tightly together even than t-SNE. The same can be said of classes 5 (pink) and 1 (orange).

This is somewhat surprising, as NNP's higher M_{DC} score would imply its clusters are better visually separated. It is, however, explained by Pre-EH's higher M_{TN} score, as it seems to be better at placing neighbouring points closer together.

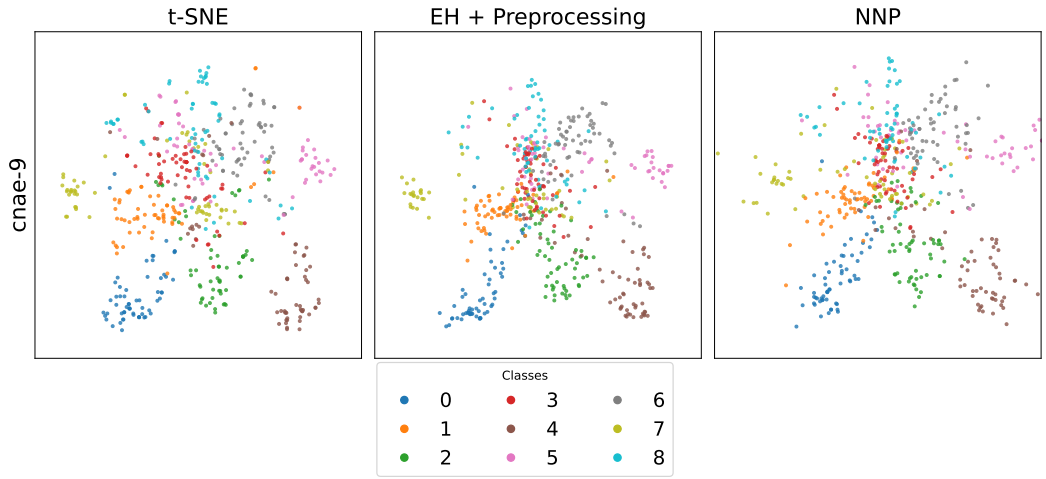


Figure 5.6: Pre-EH, NNP, and t-SNE (Ground Truth) on CNAE-9.

5.2 Convergence Analysis on MNIST

In this section, we compare the convergence behaviour for NNP and EH on the MNIST dataset. We evaluate both models at epochs 50 and 100, and NNP at a further 200 epochs. We compare these models to themselves, each other, and to t-SNE at different time steps. This section contains both a quantitative analysis of the projection quality metrics from section 2.2 and a qualitative analysis of the projections generated during these timesteps.

5.2.1 Projection Quality Metrics Comparison

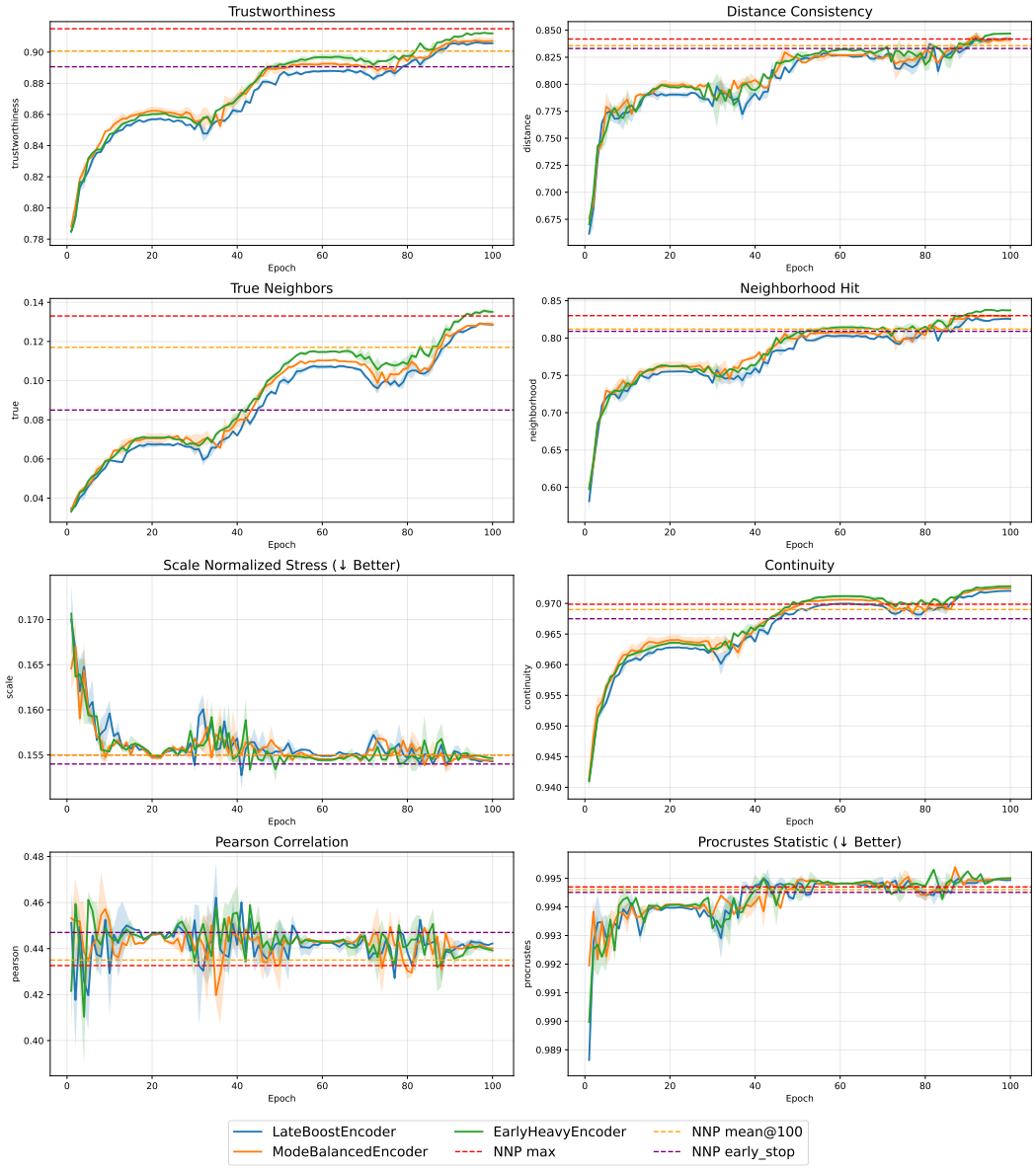


Figure 5.7: The PAT model's performance on eight projection quality metrics on MNIST, compared to NNP at three different time steps. This figure shows that EH still outperforms on most metrics, even when NNP is given more epochs. Some metrics are harmed by training too many epochs.

Figure 5.7 illustrates that Modrakowski et al.'s Early Stopping Criteria (NNP@∼46)[19] yields the best results on global projection quality metrics. Notably, there is a small but significant advantage in M_r, while differences in M_σ and M_P between Early Stopping and further training are negligible. Training further until 100 epochs does provide some improvements in the local neighborhood-preservation metrics, primarily in M_{TN} (+0.032).

Extending training further to ∼200, that is, until no metrics improve any

longer, results in effectively identical performance in global projection quality. However, there are still small improvements in neighborhood preservation metrics. Namely small increases in M_{TN} (+0.016), M_t (+0.0244), and M_{NH} (+0.018), compared to NNP@100.

EH shows the strongest performance among the PAT models and compared to these NNP baselines. NNP@∼46 however still retains a marginal advantage in M_r. In the 40-60 epoch range, EH is already better or comparable to NNP@∼46 in all metrics, and by the 50-60 epoch range it surpasses or matches NNP@100 in all metrics. At 100 epochs, it slightly outperforms NNP@200 on all metrics except for M_t.

When training EH, the performance dips at regular intervals because of the learning rate's cyclical nature due to the CosineAnnealing Learning Rate scheduler[53]. Special attention must be paid to selecting the number of epochs. This behaviour is clearly visible in Figure 5.7.

5.2.2 Visual Comparison

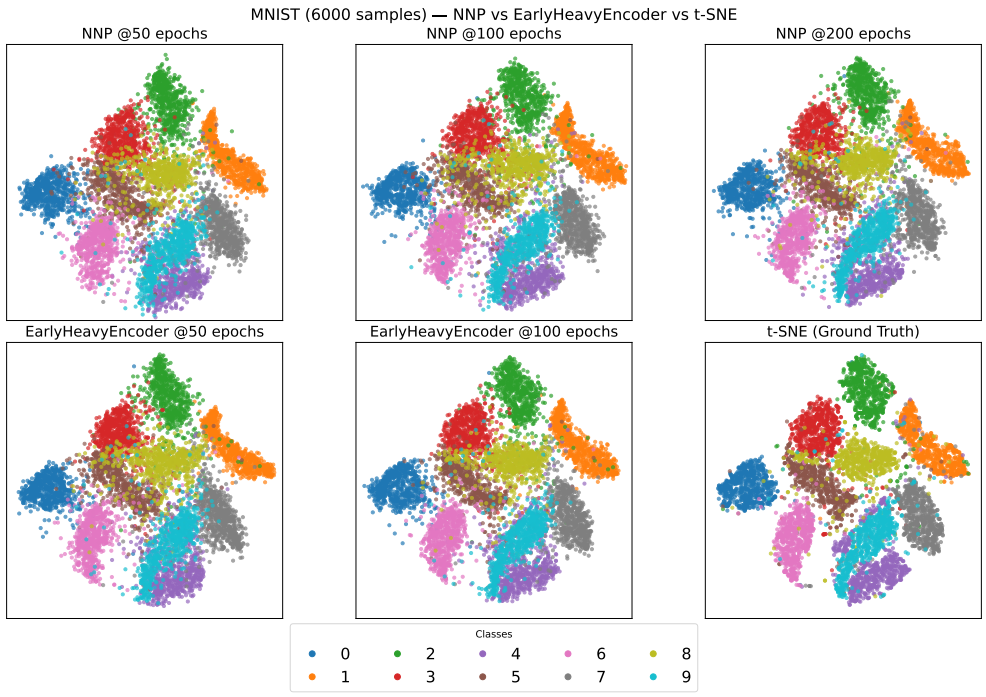


Figure 5.8: Projections from NNP, EH over a range of epochs (50, 100, 200), compared to t-SNE.

The next step is to see how visually different the projections are at these time steps. Figure 5.8 compares NNP at 50, 100, and 200 epochs, to EarlyHeavyEncoder (EH) at 50 and 100 epochs, and the test subset of the ground truth projected by t-SNE.

At 50 epochs, NNP and EH are both clearly more diffuse than the same

models in later timesteps. NNP is particularly diffuse; there are few sharp edges, which is particularly noticeable at the 3-5-8 cluster group (red, brown, lime green respectively) and their neighbouring clusters.

EH@50 is also superior to NNP@50 in shape recreation. Cluster 6 (pink) is a prime example where it far better recreates the shape, including the 'hook' shape at the bottom. It also has far less noise between clusters 6 and 9 (pink and light blue, respectively), where NNP struggles in all time steps.

NNP's projections improve after 100 epochs. Compared to NNP@50, the projection is slightly less diffuse, and its ability to recreate the shapes of the clusters improves. This is particularly evident in cluster 6 (pink), 5 (red), and 7 (grey). However, it still struggles with noise in intercluster spaces.

At 100 epochs EH appears to have higher-quality projections in most ways. Compared to NNP@100 and itself at 50, it's less diffuse, as can be seen in clusters 7, 6, 2, and 5. It is particularly far less diffuse in spaces between clusters, like that between clusters 6 and 9. It also better separates the 3-5-8 cluster group, replicating the shapes of the overlapping clusters more accurately.

Training for a further 100 epochs, i.e., to 200 epochs, yields mixed projection quality results for NNP compared to itself at 100 epochs. It better recreates some local features, like the 'hook' form at the bottom of clusters 6 and 9 (pink and light blue respectively), the latter of which PAT models struggle with. However, it also becomes more diffuse in other areas, like clusters 2 and 7 (green and grey, respectively).

Furthermore, it seems to struggle more in areas with overlapping clusters, like the 3-5-8 cluster group. Despite the improved local structure for clusters 6 and 9, there is still much noise present between them. These are issues EH@100 either does not have, or to a lesser degree.

Comparatively, EH@100 still seems to be the best model. It recreates the outlines of the clusters similarly or better, and is overall less diffuse than NNP at any timestep.

Between EH@50 and NNP@100, EH holds the advantage. Its projections are less diffuse in some areas, and retain local structure better.

5.3 Scalability

The scalability experiments were executed on different hardware than the projection quality evaluations, namely a Macbook Pro M5 with 24 GB of Unified Memory.

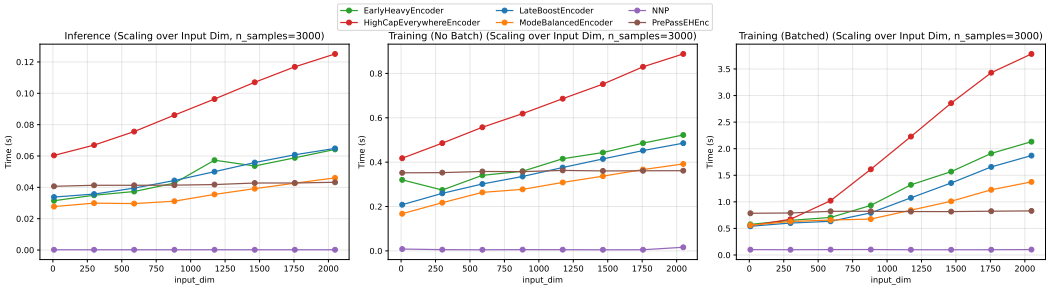


Figure 5.9: Speed scaling over number of input dimensions. Testing all 4+1 MEnc variants and NNP.

Figure 5.9 shows how inference and training time scale with the input dimensionality for all five MEnc variants and the NNP baseline. Across models, there is a clear fixed overhead associated with running a forward or backward pass, after which the growth in runtime is approximately linear in the dimensionality. As expected, HCE incurs the highest cost due to its uniformly high-capacity layers, while MB and Pre-EH are consistently the fastest among the PAT variants. NNP remains substantially faster than all PAT models: for inference and unbatched training, it is typically more than two orders of magnitude faster, although even the slowest PAT encoder still runs well below one second in this regime.

The behaviour under batched training differs markedly. While both NNP and PAT slow down when batching is enabled, their relative slowdowns are not proportional. For moderate dimensionality (≈ 882 input dimensions), EH remains within a factor of ten of NNP, and both models complete a training epoch in under one second. At the highest dimensionality tested, NNP is barely affected, whereas EH becomes nearly twice as slow as at lower dimensionalities. This stands in contrast to earlier experiments performed on a different machine, where batching improved speed by nearly an order of magnitude, highlighting that batching performance is strongly hardware dependent.

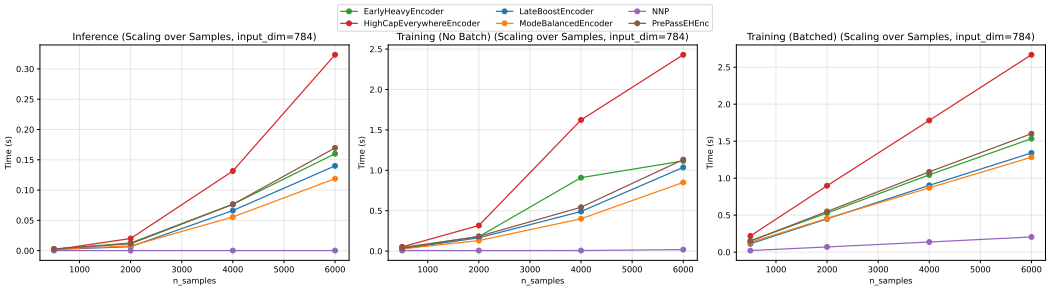


Figure 5.10: Speed scaling over number of samples. Testing all 4+1 MEnc variants and NNP.

Figure 5.10 presents the complementary analysis: runtime as a function of the number of samples. For PAT models, unbatched training time grows polynomially with sample count, whereas batched training increases more smoothly

and approximately linearly. In the tested range, unbatched training is consistently faster than batched training for all variants. Among the PAT models, MB is again the fastest, followed by LB, EH, and Pre-EH, with HCE being the slowest in all scenarios. Inference shows the same ordering. NNP remains faster than every PAT model across all sample sizes; for unbatched training, it is nearly flat with respect to sample count, while PAT runtimes increase steadily.

From a practical standpoint, these results suggest that HCE is not competitive in settings where runtime is a constraint. The remaining four PAT models perform similarly to one another up to roughly 2000 samples, and remain usable until approximately 4000 samples, beyond which NNP becomes increasingly favourable. However, inference time remains negligible for all models. Even at the largest tested sample counts, all projections complete in well under one second.

A direct comparison of representative configurations is provided in Table 5.7. When trained for 50 epochs on 6000 samples, EH reaches competitive projection quality but still requires 76.7 s, compared to 20.5 s for NNP trained for 100 epochs. Extending NNP to 200 epochs still results in a total time of only 41.0 s, which remains substantially faster than any PAT variant. For context, multicore t-SNE [51] on the full MNIST train–test set (13000 samples) completes in approximately 40 s on the same MacBook hardware. This indicates that, in terms of raw runtime, neither NNP nor PAT currently surpasses t-SNE, potentially unless the target dataset scale grows far beyond those evaluated here.

Model	Epochs	Samples	Time (s)
EH	50	6000	76.7
EH	100	4000	104.4
EH	100	6000	153.5
NNP	100	6000	20.5
NNP	200	6000	41.0

Table 5.7: Batched training time comparison for EH and NNP.

5.4 Interpretability through Attention Visualisation

A central question in understanding the behaviour of PAT models is whether individual attention heads specialise in distinguishing between clusters or whether they operate in a more globally distributed manner. Initial expectations were that heads might focus selectively on samples belonging to the same class or cluster as the query point, while other heads point to other clusters. Modelling a one-versus-rest type relation.

The global and local attention patterns show this is not the case. Instead, they imply that attention is globally spread in early layers and becomes increasingly selective only toward the end of the network.

5.4.1 Global Attention Patterns

Figure 5.11 shows the distribution of attention weights for all samples to all samples on all heads across all layers for the MNIST dataset. For all layers, the vast majority of samples receive close to zero attention. In fact, approximately 76%, 75%, and 99.8% of the samples receive less attention than the threshold, i.e., the reciprocal of the number of samples, on layers one, two, and three, respectively.

For the remaining points that do receive attention, the histograms imply that heads of the first two layers tend to distribute their attention broadly. However, in the second layer, the heads do have an order of magnitude higher maximum attention, and therefore become more discriminative. In the third layer, the heads become extremely discriminative, as their maximum attention weight increases to the maximum value for attention weights, i.e., 1.

Furthermore, while the first two layers' global attention weights look similar to a geometric series, the last layer forms an uneven, parabola-like shape with a dominant peak near zero, a secondary, smaller peak near one, and a pronounced low point around 0.5.

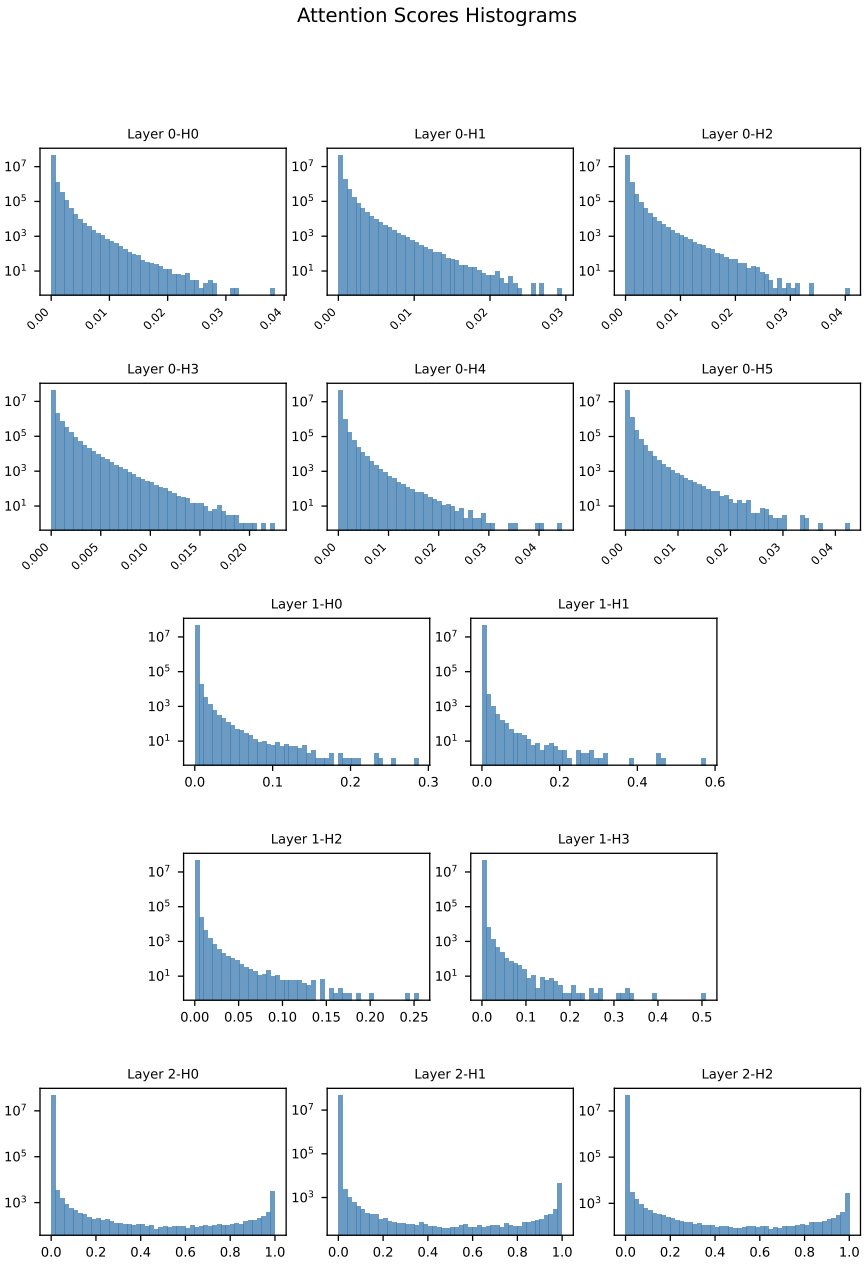


Figure 5.11: Attention weight histograms per head per layer for MNIST, log-scaled. H1 is head 1, H2 is head2, etc.

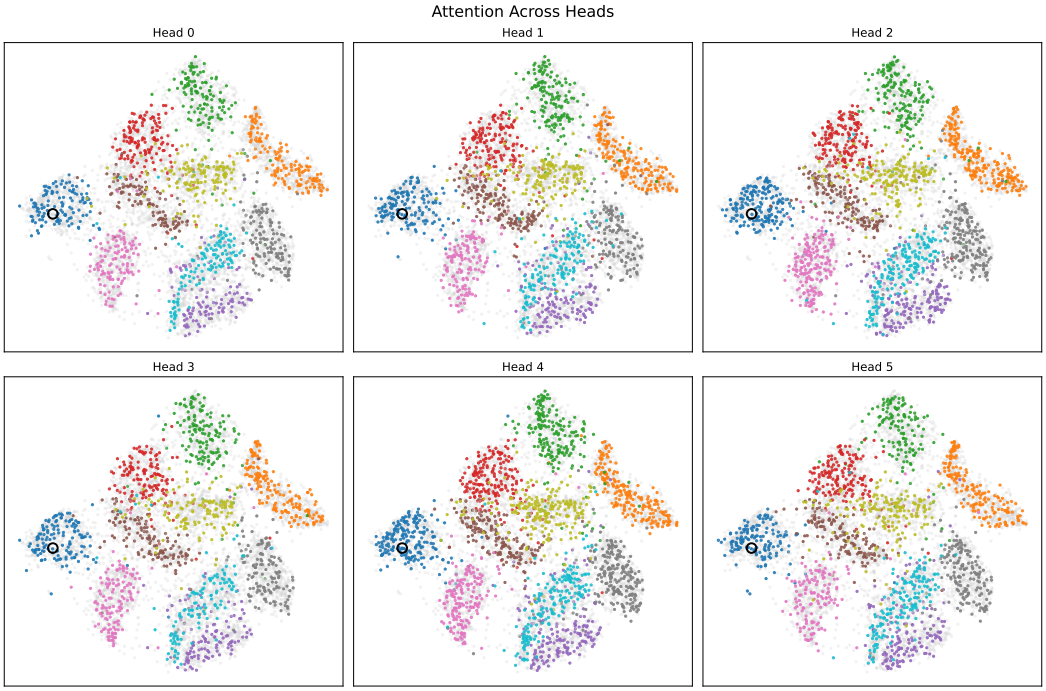
5.4.2 Attention Score Visualisation

Looking at individual samples and which samples they attend to gives insight into how the attention is distributed in practice. Figure 5.12 shows how attention is distributed over the dataset in the first two layers.

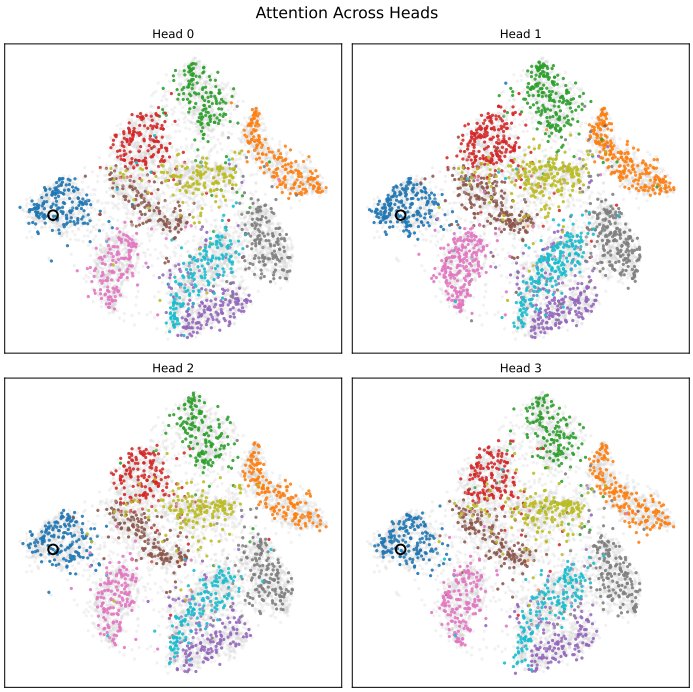
Each head attends to points in all clusters. Visually, however, it seems

that it attends less to points it (potentially erroneously) places in between clusters. As expected, the model becomes more discriminative in the second layer. It attends to fewer points, and the points are overall larger, indicating higher attention weights for those points. It is important to note that it does not exclusively attend less to points in between clusters. Some points that are placed inside the correct cluster are also not attended to.

At first glance, all the heads in the first two layers show similar attention patterns. However, they rarely attend to the exact same points, and the density of attention per cluster also differs. For example, compare the grey clusters in heads three and four in Figure 5.12a. Additionally, only rarely do samples attend to themselves.



(a) Layer one.

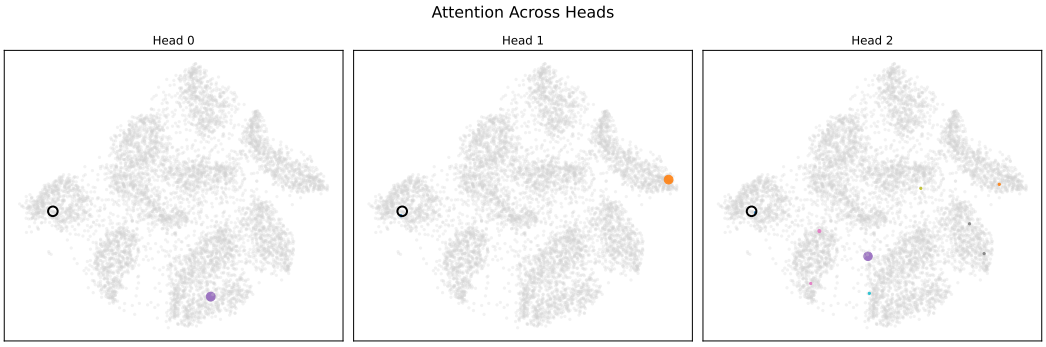


(b) Layer two.

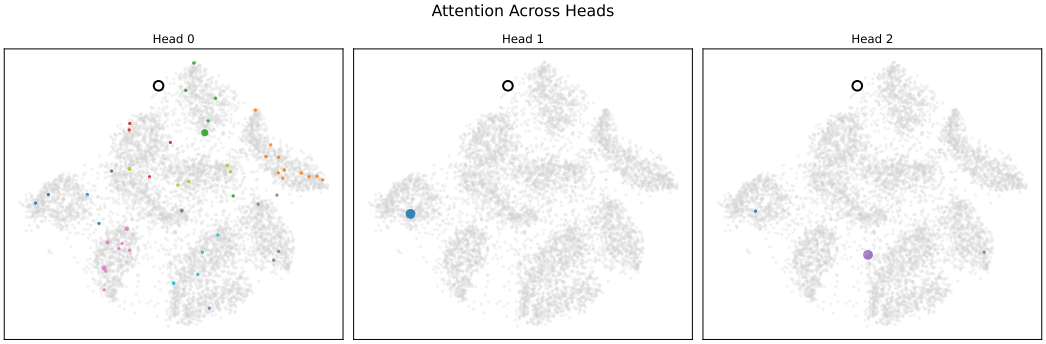
Figure 5.12: Attention weights for the first and second layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out. Sample number 4917 is placed correctly next to other members of its class 0.

The third layer, as predicted in subsection 5.4.1, is extremely discrimina-

tive. As can be seen in Figure 5.13, one or two heads attend to a small number of samples spread throughout the projection, while the remaining heads attend (almost) entirely to a single sample. Which heads attend to the spread of points and which attend to the single points varies according to the source sample.



(a) Head 2 has attention spread over multiple points, whereas 0 and 1 focus on just one. Sample 4917 is placed with digits of its own class (digit 0).



(b) Here head 0 has distributed attention, whereas 1 and 2 focus on individual points. Sample 628 is not placed with other points of its own class (digit 0).

Figure 5.13: Attention weights from the third layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out. These figures show that different heads have different functions depending on the sample.

As MNIST is an image dataset, we can visualise the samples most attended to. See Figure 5.14. There is no clear pattern in how the last layer selects the few samples it attends to.

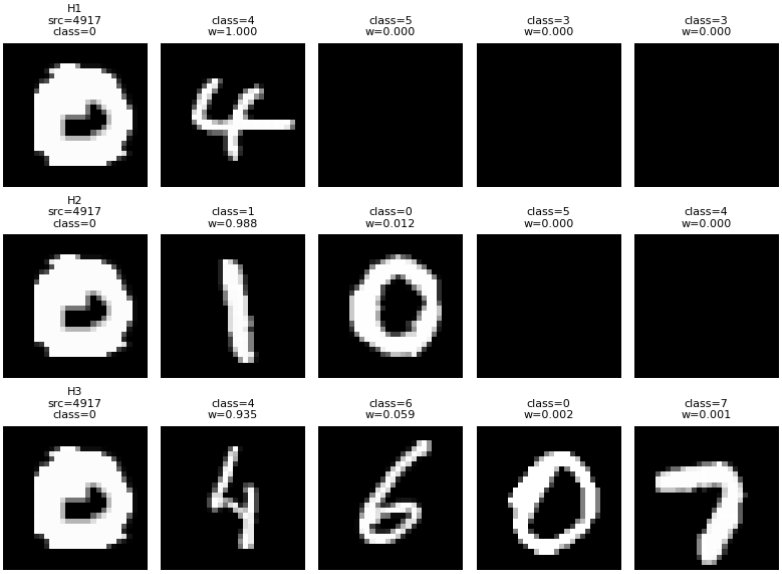


Figure 5.14: Attention weights and correlating images for top four most attended to samples from sample 4917 in the third layer. Sample 4917 is placed with digits of its own class (digit 0), but attends remarkably little to members of its own class.

5.4.3 Other Datasets

The patterns described in the prior subsections also seem present to some extent in the other datasets, particularly for the first two layers. Though to a lesser extent in datasets with ground truth projections without clear cluster separation. Examples of these patterns can be found in Appendix B. Analysing these are left for future research.

5.4.4 Conclusion

The results from visualising the attention patterns, both globally and locally, are inconclusive in terms of interpretability. Acknowledging that, we put forward the following theory on how the PAT models function:

The first layer is used to learn and capture the global structure of the data. It seemingly identifies and separates clusters. Next, it seems to use the second layer to capture finer, local details by attending to small local features and increasing the separation between clusters.

Finally, the last layer is most difficult to interpret. Perhaps it only uses one or two heads to project and ignores the other one. Some members of the VIG group theorised that the function of this layer is to identify the most dissimilar samples, and to use that to identify where it should not place the point.

6. Conclusion and Discussion

This chapter synthesises the findings of this thesis, reflecting on the performance, scalability, and interpretability of PAT models for dimensionality reduction. Drawing on the analysis of the sub-questions presented in section 6.1 and the evaluation of primary research questions in section 6.2, it discusses how the proposed models compare to traditional methods and Neural Network Projection in terms of projection quality, computational efficiency, and practical usability.

The chapter further examines limitations outlined in section 6.3, including hardware constraints, scalability challenges, and dataset-specific considerations, and concludes with directions for further exploration in section 6.4.

The implementation of this research [16] is shared publicly on the GitHub platform.

6.1 Sub-Questions

PQ1 was split into five sub-research questions.

SQ1: The first sub-question was answered by non-exhaustively searching a large parameter space on MNIST, comparing projection via W^O within the multi-head attention mechanism to projection using a modified transformer encoder. The results show that the modified transformer encoder, denoted as MEnc, more consistently produces higher projection quality on average than Attention-Only models.

SQ2: The second sub-question was answered using the same non-exhaustive parameter search, by analysing the effect of model depth, which revealed that three layers form the optimal configuration for MEnc-type models when balancing projection quality, scalability, and out-of-sample support.

Building on these findings, progressively smaller and more focused parameter spaces were explored and analysed to identify parameter combinations that exhibited consistently strong performance. Based on this analysis, four three-layer MEnc variants were constructed. In addition, an alternative architectural variant, Pre-EH, was introduced to address datasets on which the original variants struggled. These models are: EH and its preprocessing-based variant Pre-EH, and HCE, LB, and MB.

SQ3: The five models mentioned above were subsequently evaluated by learning projections that approximate t-SNE embeddings across six datasets. These experiments allow SQ3 to be answered. A higher number of attention heads leads to higher maximum projection quality across all evaluated metrics.

Since all evaluation metrics were computed exclusively on the test set, these conclusions also apply to out-of-sample support (C5). No consistent benefit was observed from using more than six attention heads in any layer. The two highest scoring models overall are HCE and EH, which are also the variants with the highest number of attention heads. However, HCE exhibits stability issues and may fail during training or inference, limiting its practical reliability.

Despite these strengths, EH and the other PAT variants fail outright on CIFAR-10, which is likely attributable to its high input dimensionality, and they underperform on CNAE-9, which is characterised by a high degree of sparsity. In these cases, Pre-EH succeeds where the other variants do not.

On CIFAR-10, Pre-EH outperforms NNP both quantitatively and visually. On CNAE-9, it outperforms NNP on M_{TN} and produces visually less diffuse projections, despite the other metrics indicating the projection is overall poorer. Notably, in these scenarios, Pre-EH is capable of generating projections that appear visually less diffuse than the original t-SNE embeddings.

Overall, PAT models marginally outperform NNP across the evaluated datasets and metrics. Among the PAT variants, EH emerges as the strongest and most consistent performer. This suggests that a funnel-shaped architecture, in which the number of attention heads decrease with depth, is particularly well suited for achieving stable and high-quality projections.

SQ4: Based on the same experiments used to address SQ1, SQ2, and primarily SQ3, we obtained a clear indication of effective configurations for the Transformer hyperparameters d_k, d_v, and the feed-forward dimension d_{ff}. A high value of $d_{ff} = 2048$ for each layer seems to be most effective, as does keeping the d_k consistent, which in the case of EH equals at the middling value of 256. Consistent with the trend observed for the number of attention heads per layer, these models also appear to benefit from a funnel-shaped configuration, in which d_v gradually decreases across successive layers.

SQ5: The last sub-question examined the scalability of PAT models in comparison to NNP. Using randomly generated data, the effects of increasing the number of samples and the number of input dimensions were analysed. The results indicate that training PAT models is 2 orders of magnitude slower than training NNP, to the extent that training can become impractical. In contrast, inference remains well within usable bounds, typically completing within one second.

Among the five evaluated PAT variants, HCE is the slowest, the others are similar to each other. They also scale differently to input dimension size and number of samples. However, EH is on average slightly slower than the remaining models.

6.2 Primary Research Questions

This section synthesises the findings of this thesis in relation to the primary research questions. Drawing on the results established through the sub-research questions and the subsequent analysis, it evaluates to what extent the proposed methods meet the stated criteria for projection quality (C1), scalability (C5), ease of use (C3), and interpretability. Each primary research question is addressed in turn, with the experimental outcomes and qualitative observations consolidated into concise conclusions.

6.2.1 Projection Quality, Scalability, and Ease of Use

With respect to projection quality, the two strongest performing PAT variants are EH and HCE. While HCE can achieve very high scores across projection quality metrics, it suffers from stability issues that limit its reliability in practice. In contrast, EH exhibits consistent behaviour across most datasets. Nevertheless, EH, as well as the other PAT variants, struggle on datasets that are either extremely sparse or characterised by very high input dimensionality.

Scalability further differentiates these models. HCE is substantially slower than the other PAT variants and is practically unusable when compared not only to its architectural peers but also to NNP. This performance gap places clear practical limits on its applicability, despite its strong projection quality under favourable conditions.

These findings allow the first primary research question to be answered affirmatively. It is possible to design an attention-based neural projection method that marginally outperforms state-of-the-art deep-learning-based projection methods in matching the projection quality of traditional techniques. Across a wide range of datasets, PAT models demonstrate a superior ability, albeit by a small margin, to reconstruct fine-grained local structure when compared to NNP. Specifically, PAT more accurately recovers subcluster structure in HAR, preserves fine local detail in spambase, produces substantially less diffuse projections for the already diffuse t-SNE embeddings of CNAE-9, more faithfully reconstructs the global outline of CIFAR-10, and achieves improved cluster separation on both MNIST and fMNIST.

While t-SNE still handily outperforms both models on M_{TN}, surprisingly, PAT often outperforms it on M_{DC}, which NNP rarely does.

An additional advantage of PAT models is their ability to outperform NNP's peak projection quality while requiring fewer training samples. This makes them particularly attractive in scenarios where training data is limited.

While the relative performance difference between NNP and PAT depends on the dataset, the overall trend indicates that when projection quality is the primary objective, EH is the most suitable default choice. Its inference speed is only marginally slower than that of NNP, while consistently achieving higher-resolution projections. For datasets that are sparse or have very

high dimensionality, Pre-EH is the preferred alternative, as it offers improved robustness and faster execution than EH under these conditions.

Importantly, these gains in projection quality are achieved without a substantial loss in ease of use. From a practical perspective, PAT models are, at worst, only marginally less user-friendly than NNP. Both approaches rely on standard neural network training pipelines and provide reasonable default parameter settings that perform well across diverse datasets. Although they employ different loss functions and learning rate schedulers, PAT can be treated as a near drop-in replacement for NNP. While EH may not be universally applicable, switching to Pre-EH requires only a minor configuration change.

The primary trade-off introduced by PAT lies in scalability. These models have a substantially larger number of trainable parameters, and attention mechanisms inherently scale polynomially with input size. Although modern hardware mitigates these costs to some extent, training times are still increased by a factor of approximately two to eight, depending on the training regime. Resulting training times that exceed one minute may be considered unacceptable in many practical settings, particularly when the observed improvements in projection quality are modest for some datasets.

In conclusion, while the use of PAT models is not discouraged, their scalability characteristics place them in a different practical category than NNP. They are less suitable for applications that require near-instantaneous results. When projection quality is the dominant concern, a workflow based on EH is recommended. For datasets that are highly sparse or extremely high-dimensional, Pre-EH is a more appropriate choice. If scalability with respect to the number of training samples is the primary requirement, NNP remains the most viable option, although Pre-EH may serve as a competitive alternative. In cases of uncertainty, EH, Pre-EH, and NNP should be compared using both visual inspection and projection quality metrics, after which the best performing model can be selected.

6.2.2 Interpretability through Attention Visualisation

The results obtained from visualising attention patterns, both at a global and a local level, are inconclusive with respect to interpretability. While clear and consistent explanatory patterns could not be definitively established, the observed behaviour nonetheless motivated a working theory regarding the internal functioning of PAT models.

Despite the ambiguity of the observed attention patterns, the second primary research question can be answered affirmatively. It is possible to design a method for visualising attention patterns in projection use cases for analysis. Global attention behaviour can be identified by constructing per-head, per-layer histograms aggregated over all samples. Local attention behaviour can be examined by visualising the learned projections and encoding attention strength through visual emphasis, while samples receiving attention weights below a defined threshold, set to the reciprocal of the number of samples, are

visually deemphasised.

Although the attention patterns produced by PAT models do not yield conclusive interpretability insights, they may hold the key to understanding how attention-based models could be made more performant than NNP in terms of projection quality. The attention pattern visualisation has already partially inspired one future research idea in subsection 6.4.1.

On the whole, the proposed visualisation approach is itself effective and broadly applicable. The method is reusable, generalises across all models within the PAT class, and provides an intuitive means of exploring both global and local attention behaviour. As such, it constitutes a practical tool for analysis for continued research on projection with attention.

6.3 Discussion

This section reflects on the practical and methodological considerations of this thesis. It examines the limitations encountered during model development and evaluation, including hardware constraints, scalability challenges, and dataset-specific issues. The discussion also contextualises the observed performance of PAT models relative to NNP and t-SNE, providing insights into when and why each approach may be preferred.

6.3.1 Hardware Limitations

A significant portion of this research was conducted on outdated and unreliable hardware, which had a substantial influence on both methodological and practical decisions. These constraints directly informed several design choices, most notably the selection of batch sizes, dataset sizes, and the overall configuration of the training pipeline. In addition, hardware limitations shaped the parameter selection strategy, as the available resources prevented a faster and more exhaustive exploration of the parameter space. With access to more capable hardware, it would have been feasible to search this space both more deeply and more efficiently.

Hardware constraints also affected the evaluation procedure. Due to time limitations, each model was evaluated using only two runs per dataset. As a result, the statistical robustness of conclusions regarding model stability is limited. While this evaluation provides indicative trends, it does not allow for strong claims about variance or reliability across runs. To partially mitigate this issue, stability problems that occurred outside the formal evaluation phase were explicitly noted and incorporated into the qualitative assessment where possible. Nonetheless, these limitations should be taken into account when interpreting the reported results.

6.3.2 Scalability Issues

An initial assumption underlying this work was that t-SNE is substantially slower than NNP. Given that attention mechanisms scale polynomially, it was never considered realistic for PAT models to match the scalability of NNP. Consequently, the design objective was to achieve performance that is faster than t-SNE, while allowing PAT to be slower than NNP within a limited and acceptable margin.

During the scalability experiments, these assumptions were partially challenged. While prior work improving NNP reports inference speeds that reach parity with t-SNE at approximately 5000 samples [13], the results obtained in this thesis indicate that multi-core implementations of t-SNE can, in fact, be faster than NNP. That is, at least up to the maximum dataset size evaluated, which was 13000 samples. Although NNP retains the important advantage of out-of-sample support, one of its commonly cited motivations, namely superior speed, appears to be overstated or at least highly dependent on implementation and hardware configuration. It remains true, however, that NNP scales approximately linearly with the number of samples, so it must eventually be faster than the polynomial t-SNE.

Under the scalability criteria defined at the outset of this thesis, PAT models do not meet the stated requirements. Nevertheless, these findings suggest that the scalability criteria themselves warrant reconsideration. As a result, future research should re-evaluate the dataset size categories proposed by Espadoto et al. [11], and ensure that any updated categorisation is reflected in the selection and construction of benchmark datasets used for evaluating projection methods.

6.3.3 Basing Parameters Exclusively on MNIST

Although MEnc variants perform reasonably well across the evaluated datasets, basing the estimation of globally optimal hyperparameters exclusively on MNIST likely resulted in a degree of over-specialisation. This focus may have limited the achievable performance on other datasets, effectively leaving projection quality and stability improvements unexploited. In particular, hyperparameters related to the learning rate and its scheduling are likely to admit more robust, dataset-agnostic values than those identified through tuning on MNIST alone.

As a consequence, the reported performance of the MEnc models on the remaining datasets may underestimate their true potential. With a more diverse and representative hyperparameter optimisation strategy, it is plausible that these models could achieve higher projection quality and improved stability across a broader range of data characteristics.

6.3.4 Limiting to t-SNE as Ground Truth

Restricting the evaluation to t-SNE projections as ground truth limits the assessment of genericity for PAT models. This choice was primarily motivated by practical considerations. Focusing on a single projection method significantly reduced development time and was necessary given the available hardware and time constraints. While t-SNE provides a widely used and well-understood reference, this restriction prevents conclusions from being drawn about how well PAT generalises to approximating projections produced by other traditional DR techniques, such as UMAP.

As a result, the findings of this thesis primarily characterise performance relative to t-SNE, rather than projection methods in general. Broadening the set of reference techniques would enable a more thorough evaluation of the flexibility and robustness of PAT. This would clarify whether the observed improvements extend beyond this specific projection technique.

6.3.5 Limitations of Attention Weights in Interpretability and Explainability

Visualising attention weights provides insight into which input elements are being attended to, but it does not convey how these attention patterns influence the resulting projections. In other words, attention-based visualisations omit the contribution of the *value* matrix, leaving a methodological gap that affects all attention visualisation algorithms. Consequently, such visualisations can indicate where the model focuses its attention but cannot explain the precise effect of this focus on the output.

Research exists that addresses this limitation by visualising alternative aspects of transformer models. For instance, Visbert [58] visualises the hidden states from each encoder block to provide a more complete understanding of model behaviour. While exploring these approaches is beyond the scope of this thesis due to time constraints, they represent promising directions for future work aimed at improving the interpretability and explainability of PAT and related attention-based models.

6.4 Future Research

The following section outlines potential directions for further research on PAT models. The ideas are presented roughly in order of increasing complexity, beginning with relatively straightforward modifications such as adjusting batch sizes, and progressing to more ambitious extensions such as unified encoder-decoder frameworks and foundational multi-dataset models. Research on self- and semi-supervised PAT models is not included here, as it is already an active area of investigation.

6.4.1 Batch Sizes and Scalability

Batching has a significant impact on the scalability of PAT models. In the current implementation, the batch size for training is relatively small. Investigating the use of larger batch sizes could potentially reduce training time and improve projection quality. Additionally, larger batches might influence the learned attention patterns, which could reveal new insights into how the model encodes local and global structures.

For inference, determining the ideal number of samples to project at once is an open question. Preliminary experiments suggested that projecting all samples simultaneously outperforms using the training batch size. However, for very large datasets, alternative strategies could improve efficiency. For example, projecting three batches of 2000 samples each is faster than projecting a single batch of 9000 samples, and could help scale inference more linearly. Future research should investigate the trade-offs between batch size, projection quality, and runtime performance to identify optimal strategies for both training and inference.

6.4.2 Relative Positional Embeddings

Transformers rely on positional embeddings to encode the order of input elements. Classic absolute positional embeddings, as used in Attention is All You Need[4], are not suitable for projection tasks because they assume a one-directional sequence. For our PAT models, relative positional embeddings are required. Recent research has explored various relative positional embedding methods, including Rotary Position Embedding (RoPE) [59], a popular relative positional embedding method in the domain of natural language processing. RoPE builds on earlier foundational works, including Shaw et al.'s self-attention with relative position representations [60], Transformer-XL [61], XLNet [62], the text-to-text transformer by Raffel et al. [63], the enhanced relative position embeddings proposed by Huang et al. [64], and TUPE [65].

In computer vision, relative positional embeddings have been adapted to image data, for example, in the Swin Transformer [66], iRPE methods [67], and 2D adaptations of RoPE [68], [69], including head-wise adaptive extensions for fine-grained image generation [70].

For 3D and point cloud data, methods such as Point Transformer V3 [71], Stratified Transformer [72], and IBT [73] leverage relative positional encoding to capture both local and global spatial structure. Investigating these techniques may allow us to reduce the size of PAT models while retaining performance.

6.4.3 Downsampling data and Minimising Attention Layers

Our current research shows that Pre-EH, which applies a linear layer to downsample high-dimensional data before entering the MEnc, can achieve projections that are similar to or even better than the other PAT models. It also has better genericity (C4), as it works on sparse and high-dimensional data.

This observation suggests that incorporating additional linear layers at the start of the network, or further downsampling the input data, could allow us to reduce the number of attention layers required. Minimising attention layers in this manner may substantially improve the runtime and scalability of PAT models without compromising projection quality. Future research could explore optimal downsampling strategies and quantify how few attention layers are needed to maintain both stable and accurate projections across diverse datasets.

6.4.4 Inverse Projection

A natural extension of PAT models is to explore inverse projection, where the goal is to reconstruct high-dimensional data from low-dimensional embeddings. Initial research could focus on implementing standard inverse projection using PAT to evaluate feasibility and performance.

A more advanced approach would be to train a unified projection and inverse projection model in an autoencoder-type framework, not unlike Transformer-DR[48]. We propose the following modification: a multi-objective loss function could be used to simultaneously mimic the traditional DR method at the encoder output while reconstructing the original high-dimensional data at the decoder output. Including the reconstruction objective may provide additional supervision, which could improve robustness, stability, and generalisation of the learned projections. Future research could investigate the benefits of this dual-objective training for both projection quality and out-of-sample reconstruction.

6.4.5 Decoders for Dimensionality Reduction

Transformer decoders typically attend to both the input sequence and the previously decoded or ground truth output sequence. We propose designing an encoder-decoder transformer model for projection, although a decoder-only architecture, as used in the GPT family of models, could also be explored.

In this framework, the decoder would be trained using both the low-dimensional embeddings and the corresponding high-dimensional ground truth when available. Decoders attend to both the encoder output and previous outputs. Instead of reconstructing samples sequentially, the decoder can process all samples simultaneously while employing a masking mechanism to handle unknown high-dimensional points.

Analogous to masked self-attention in language models [4], the model would attend only to known high-dimensional data while ignoring missing values. This approach enables the model to leverage available ground truth information while still learning to infer missing projections in a structured and consistent manner. Future research could explore the performance of encoder-decoder and decoder-only approaches and their impact on projection quality.

6.4.6 Foundational Dimensionality Reduction Model

A particularly interesting direction for future research is to develop a foundational Dimensionality Reduction model through multi-dataset learning. In this approach, each dataset would be projected using learnable parameters specific to that dataset, mapping to a shared intermediate dimensionality. These intermediate embeddings would then be input to a shared attention-based model that projects all datasets to 2D.

This setup would allow for dataset-specific adaptations while maintaining a shared projection backbone. Such a framework could enable a unified, foundational model for dimensionality reduction across multiple datasets. Previous work by Espadato et al. [13] demonstrated that transfer learning, i.e., training on one dataset and then continuing on another, was effective for NNP. Extending this concept to PAT models could provide a natural continuation and improve generalisability across diverse datasets.

Acronyms

M_c Continuity. 11, 12, 36, 41–47, 49, 51, 53

M_t Trustworthiness. 11, 12, 36, 41–49, 51, 53, 56

M_{DC} Distance Consistency. 1, 13, 36, 41–47, 49–54, 68

M_{NH} Neighbourhood Hit. 12, 36, 41–51, 53, 56

M_P Procrustes Error. 13, 36, 41–45, 47, 49, 51–53, 55

M_{TN} True Neighbors. 1, 12, 36, 41–56, 67, 68

M_σ Scale-Normalized Stress. 12, 36, 41–45, 47, 49, 51, 53, 55

M_r Pearson Correlation. 14, 36, 41–47, 49–51, 53, 55, 56

BERT Bidirectional Encoder Representations from Transformers. 21

CIFAR-10 Canadian Institute For Advanced Research. 3–5, 14–16, 28, 29, 31, 32, 37, 40, 50–52, 67, 68, 99

CNAE-9 National Classification of Economic Activities (Classificação Nacional de Atividades Econômicas). 3, 4, 6, 14, 17, 28, 29, 31, 37, 40, 52–54, 67, 68, 101

DL Deep Learning. 8

DR Dimensionality Reduction. 1–5, 7–11, 17, 18, 20, 23, 25, 26, 72, 74, 75

EH EarlyHeavyEncoder. 4, 34–37, 40–50, 53–59, 66–69

fMNIST Fashion-MNIST. 3–5, 14, 15, 28, 29, 31, 32, 40, 43–45, 68, 93

GPT Generative Pre-trained Transformer. 21, 74

HAR Human Action Recognition. 3–5, 14, 16, 28, 29, 31, 40, 48–50, 68, 97

HCE HighCapEverywhereEncoder. 4, 34, 41, 42, 44, 46–50, 53, 58, 59, 66–68

kNNP K-Nearest Neighbours Neural Network Projection. 10, 25

LB LateBoostEncoder. 34, 41, 42, 44, 46, 47, 49, 53, 59, 66

MB ModeBalancedEncoder. 34, 41, 42, 44, 47, 49, 53, 58, 59, 66

MEnc Modified-Encoder. 5, 30, 32–34, 37, 50, 58, 66, 71, 74

ML Machine Learning. 7, 8, 10

MNIST modified National Institute of Standards and Technology Database. 2, 4–6, 14–16, 25, 26, 28–32, 34, 35, 37, 38, 40–43, 54, 55, 59–61, 64, 66, 68, 71

MSE Mean Squared Error. 32

NLP Natural Language Processing. 17, 18, 27

NNP Neural Network Projection. 1, 3–5, 7, 10, 11, 23, 26, 30, 31, 34–37, 40–59, 66–71, 75, 76

PAT Projection with ATtention. 1, 2, 4, 25–27, 30, 32–37, 40, 41, 43, 45, 46, 48, 50, 52, 53, 55–59, 65–75

PCA Principal Component Analysis. 7–9

PQ1 Can we design an attention-based neural architecture for projection that matches the projection quality (C1) of traditional projection methods better than state-of-the-art deep-learning-based methods for projections without sacrificing their scalability (C2) and ease of use (C3)?. 5, 66

PQ2 Can we design a method for visualising attention patterns in projection usecases for analysis?. 5

Pre-EH PreprocessedEHEncoder. 4, 34, 37, 46–54, 58, 59, 66, 67, 69, 74

RNN Recurrent Neural Network. 17, 20

RoPE Rotary Position Embedding. 73

SHaRP Shape-Regularized Multidimensional Projections. 11

SQ1 How does projecting via W^O within the multi-head attention mechanism compare to using a modified transformer encoder in terms of projection quality (C1)?. 5, 66, 67

SQ2 How does the number of layers in the Attention-Only or Modified-Encoder projection model impact the overall performance (i.e., projection quality (C1), out-of-sample support (C5), and scalability (C2))?. 5, 66, 67

SQ3 What is the effect of varying the number of attention heads on the projection quality (C1) and out-of-sample support (C5) of the model?. 5, 66, 67

SQ4 How do the attention-related hyperparameters d_k, d_v, and the feed-forward dimension d_{ff} affect the projection quality (C1) of attention-based projection methods?. 5, 67

SQ5 What is the impact of dataset size and dimensionality on the scalability (C2) of Attention-based models?. 5, 67

SSNP Self-Supervised Network Projection. 10, 11, 23

t-SNE t-Distributed Stochastic Neighbour Embedding. 1, 3–5, 7–11, 18, 23, 25, 26, 29–31, 34, 36, 40–54, 56, 59, 66–68, 70–72

UMAP Uniform Manifold Approximation and Projection for Dimension Reduction. 1, 3, 7–11, 25, 72

VIG Visualisation and Graphics Group. 1, 65

ViT Vision Transformer. 23

Bibliography

[1] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008, ISSN: 1533-7928. Accessed: Feb. 24, 2025. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.

[2] M. Espadoto, N. S. T. Hirata, and A. C. Telea, *Deep Learning Multidimensional Projections*, arXiv:1902.07958 [cs], Feb. 2019. DOI: 10.48550/arXiv.1902.07958. Accessed: Jan. 17, 2025. [Online]. Available: http://arxiv.org/abs/1902.07958.

[3] L. McInnes, J. Healy, and J. Melville, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, arXiv:1802.03426 [stat], Sep. 2020. DOI: 10.48550/arXiv.1802.03426. Accessed: Feb. 25, 2025. [Online]. Available: http://arxiv.org/abs/1802.03426.

[4] A. Vaswani et al., "Attention is All you Need", in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. Accessed: Feb. 22, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805 [cs], May 2019. DOI: 10.48550/arXiv.1810.04805. Accessed: Feb. 24, 2025. [Online]. Available: http://arxiv.org/abs/1810.04805.

[6] P. E. Rauber, S. G. Fadel, A. X. Falcão, and A. C. Telea, "Visualizing the Hidden Activity of Artificial Neural Networks", *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 101–110, Jan. 2017, Conference Name: IEEE Transactions on Visualization and Computer Graphics, ISSN: 1941-0506. DOI: 10.1109/TVCG.2016.2598838. Accessed: Mar. 14, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7539329.

[7] B. C. Benato, A. C. Telea, and A. X. Falcão, "Semi-Supervised Learning with Interactive Label Propagation Guided by Feature Space Projections", in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, ISSN: 2377-5416, Jan. 2019, pp. 392–399. DOI: 10.1109/SIBGRAPI.2018.00057. Accessed: Mar. 14, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8614354.

[8] B. C. Benato, J. F. Gomes, A. C. Telea, and A. X. Falcão, "Semi-automatic data annotation guided by feature space projection", en, *Pattern Recognition*, vol. 109, p. 107 612, Aug. 2020, ISSN: 00313203.

DOI: 10.1016/j.patcog.2020.107612. Accessed: Mar. 14, 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320320304155.

[9] D. Rumelhart, G. Hinton, and R. Williams, "Learning Internal Representations by Error Propagation", en, Book Title: Readings in Cognitive Science, Elsevier, 1988, pp. 399–421, ISBN: 978-1-4832-1446-7. DOI: 10.1016/B978-1-4832-1446-7.50035-2. Accessed: Feb. 24, 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9781483214467500352.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", en, *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1127647. Accessed: Feb. 19, 2025. [Online]. Available: https://www.science.org/doi/10.1126/science.1127647.

[11] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea, "Toward a Quantitative Survey of Dimension Reduction Techniques", *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2153–2173, Mar. 2021, Conference Name: IEEE Transactions on Visualization and Computer Graphics, ISSN: 1941-0506. DOI: 10.1109/TVCG.2019.2944182. Accessed: Feb. 22, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/8851280.

[12] M. Espadoto, F. C. M. Rodrigues, N. S. T. Hirata, and R. Hirata, "Deep Learning Inverse Multidimensional Projections", en, 2019.

[13] M. Espadoto, N. Hirata, A. Falcão, and A. Telea, "Improving Neural Network-based Multidimensional Projections:" en, in *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Valletta, Malta: SCITEPRESS - Science and Technology Publications, 2020, pp. 29–41, ISBN: 978-989-758-402-2. DOI: 10.5220/0008877200290041. Accessed: Feb. 18, 2025. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0008877200290041.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, Publisher: Institute of Electrical and Electronics Engineers. DOI: 10.1109/5.726791. Accessed: Feb. 27, 2025. [Online]. Available: https://hal.science/hal-03926082.

[15] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms*, Publication Title: arXiv e-prints ADS Bibcode: 2017arXiv170807747X, Aug. 2017. DOI: 10.48550/arXiv.1708.07747. Accessed: Feb. 27, 2025. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2017arXiv170807747X.

[16] C. Smet, *Projection with Attention*, en, Jan. 2026. [Online]. Available: https://github.com/Casper-Smet/projection-with-attention.

[17] A. Telea, "Beyond the Third Dimension: How Multidimensional Projections and Machine Learning Can Help Each Other:" en, in *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2023, pp. 5–16, ISBN: 978-989-758-634-7. DOI: 10.5220/0011926400003417. Accessed: Feb. 19, 2025. [Online]. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0011926400003417.

[18] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[19] T. S. Modrakowski, M. Espadoto, A. X. Falcão, N. S. T. Hirata, and A. Telea, "Improving Deep Learning Projections by Neighborhood Analysis", en, in *Computer Vision, Imaging and Computer Graphics Theory and Applications*, K. Bouatouch et al., Eds., Cham: Springer International Publishing, 2022, pp. 127–152, ISBN: 978-3-030-94893-1. DOI: 10.1007/978-3-030-94893-1_6.

[20] M. Espadoto, N. Hirata, and A. Telea, "Self-supervised Dimensionality Reduction with Neural Networks and Pseudo-labeling:" en, in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, SCITEPRESS - Science and Technology Publications, 2021, pp. 27–37, ISBN: 978-989-758-488-6. DOI: 10.5220/0010184800270037. Accessed: Feb. 19, 2025. [Online]. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010184800270037.

[21] Y. Kim, A. C. Telea, S. C. Trager, and J. Btm Roerdink, "Visual cluster separation using high-dimensional sharpened dimensionality reduction", en, *Information Visualization*, vol. 21, no. 3, pp. 197–219, Jul. 2022, ISSN: 1473-8716, 1473-8724. DOI: 10.1177/14738716221086589. Accessed: Feb. 19, 2025. [Online]. Available: https://journals.sagepub.com/doi/10.1177/14738716221086589.

[22] A. Machado, A. Telea, and M. Behrisch, "ShaRP: Shape-Regularized Multidimensional Projections", *EuroVis Workshop on Visual Analytics (EuroVA)*, pp. 1–6, 2023, arXiv:2306.00554 [cs]. DOI: 10.2312/eurova.20231088. Accessed: Feb. 19, 2025. [Online]. Available: http://arxiv.org/abs/2306.00554.

[23] H. Jeon et al., *ZADU: A Python Library for Evaluating the Reliability of Dimensionality Reduction Embeddings*, arXiv:2308.00282 [cs], Aug. 2023. DOI: 10.48550/arXiv.2308.00282. Accessed: Mar. 14, 2025. [Online]. Available: http://arxiv.org/abs/2308.00282.

[24] A. Machado, M. Behrisch, and A. Telea, "Necessary but not Sufficient: Limitations of Projection Quality Metrics", en, *Computer Graphics Forum*, vol. 44, no. 3, e70101, 2025, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.11 ISSN: 1467-8659. DOI: 10.1111/cgf.70101. Accessed: Nov. 18, 2025.

[Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.70101.

[25] J. Venna and S. Kaski, "Local multidimensional scaling", en, *Neural Networks*, vol. 19, no. 6-7, pp. 889–899, Jul. 2006, ISSN: 08936080. DOI: 10.1016/j.neunet.2006.05.014. Accessed: Mar. 14, 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0893608006000724.

[26] K. Smelser, J. Miller, and S. Kobourov, *"Normalized Stress" is Not Normalized: How to Interpret Stress Correctly*, arXiv:2408.07724 [cs] version: 1, Aug. 2024. DOI: 10.48550/arXiv.2408.07724. Accessed: Sep. 8, 2025. [Online]. Available: http://arxiv.org/abs/2408.07724.

[27] A. Machado, M. Behrisch, and A. Telea, "Extensible TensorFlow Implementations of Projection Quality Metrics", en, *VisGap - The Gap between Visualization Research and Visualization Software*, 2025, Artwork Size: 8 pages Edition: 1160 ISBN: 9783038682899 Publisher: The Eurographics Association. DOI: 10.2312/VISGAP.20251160. Accessed: Nov. 19, 2025. [Online]. Available: https://diglib.eg.org/handle/10.2312/visgap20251160.

[28] F. Paulovich, L. Nonato, R. Minghim, and H. Levkowitz, "Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping", en, *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564–575, May 2008, ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.70443. Accessed: Mar. 14, 2025. [Online]. Available: http://ieeexplore.ieee.org/document/4378370/.

[29] R. M. Martins, D. B. Coimbra, R. Minghim, and A. C. Telea, "Visual analysis of dimensionality reduction quality for parameterized projections", *Computers & Graphics*, vol. 41, pp. 26–42, Jun. 2014, ISSN: 0097-8493. DOI: 10.1016/j.cag.2014.01.006. Accessed: Nov. 18, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849314000235.

[30] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan, "Selecting good views of high-dimensional data using class consistency", in *Computer graphics forum*, Issue: 3, vol. 28, Wiley Online Library, 2009, pp. 831–838. Accessed: Nov. 18, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01467.x.

[31] Y. Goldberg and Y. Ritov, "Local procrustes for manifold embedding: A measure of embedding quality and embedding algorithms", en, *Machine Learning*, vol. 77, no. 1, pp. 1–25, Oct. 2009, ISSN: 1573-0565. DOI: 10.1007/s10994-009-5107-9. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1007/s10994-009-5107-9.

[32] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1098–1107, Dec. 2005, ISSN: 1941-0492. DOI: 10.110

9/TSMCB.2005.850151. Accessed: Nov. 18, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1542257.

[33] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images", 2009, Publisher: Toronto, ON, Canada. Accessed: Nov. 17, 2025. [Online]. Available: http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf.

[34] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine", in *Ambient Assisted Living and Home Care*, D. Hutchison et al., Eds., vol. 7657, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 216–223, ISBN: 978-3-642-35394-9 978-3-642-35395-6. DOI: 10.1007/978-3-642-35395-6_30. Accessed: Nov. 17, 2025. [Online]. Available: http://link.springer.com/10.1007/978-3-642-35395-6_30.

[35] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, *Spambase dataset*, 1999.

[36] P. M. Ciarelli and E. Oliveira, "Agglomeration and Elimination of Terms for Dimensionality Reduction", en, in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, Pisa, Italy: IEEE, 2009, pp. 547–552, ISBN: 978-1-4244-4735-0. DOI: 10.1109/ISDA.2009.9. Accessed: Nov. 17, 2025. [Online]. Available: http://ieeexplore.ieee.org/document/5364970/.

[37] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd. 2025. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[38] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014. Accessed: Mar. 10, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

[39] K. Cho et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, arXiv:1406.1078 [cs], Sep. 2014. DOI: 10.48550/arXiv.1406.1078. Accessed: Mar. 10, 2025. [Online]. Available: http://arxiv.org/abs/1406.1078.

[40] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", *ArXiv*, vol. 1409, Sep. 2014.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", 2016, pp. 770–778. Accessed: Mar. 12, 2025. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

[42] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, arXiv:1607.06450 [stat], Jul. 2016. DOI: 10.48550/arXiv.1607.06450. Accessed:

Mar. 12, 2025. [Online]. Available: http://arxiv.org/abs/1607.06450.

[43] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd. 2025, Online manuscript released August 24, 2025. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[44] A. Vaswani et al., "Tensor2Tensor for Neural Machine Translation", in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, C. Cherry and G. Neubig, Eds., Boston, MA: Association for Machine Translation in the Americas, Mar. 2018, pp. 193–199. Accessed: Dec. 5, 2025. [Online]. Available: https://aclanthology.org/W18-1819/.

[45] J. Vig, *A Multiscale Visualization of Attention in the Transformer Model*, arXiv:1906.05714 [cs], Jun. 2019. DOI: 10.48550/arXiv.1906.05714. Accessed: Dec. 3, 2025. [Online]. Available: http://arxiv.org/abs/1906.05714.

[46] J. Vig, *Jessevig/bertviz*, original-date: 2018-12-16T16:50:42Z, Dec. 2025. Accessed: Dec. 5, 2025. [Online]. Available: https://github.com/jessevig/bertviz.

[47] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners", en, Feb. 2019.

[48] R. Ran, T. Gao, and B. Fang, *Transformer-based dimensionality reduction*, arXiv:2210.08288 [cs], Oct. 2022. DOI: 10.48550/arXiv.2210.08288. Accessed: Feb. 24, 2025. [Online]. Available: http://arxiv.org/abs/2210.08288.

[49] A. Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, arXiv:2010.11929 [cs], Jun. 2021. DOI: 10.48550/arXiv.2010.11929. Accessed: Dec. 12, 2025. [Online]. Available: http://arxiv.org/abs/2010.11929.

[50] C. R. Harris et al., "Array programming with NumPy", *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[51] D. Ulyanov, *Multicore-TSNE*, 2016. [Online]. Available: https://github.com/%20DmitryUlyanov/Multicore-TSNE.

[52] A. Machado, *ShaRP*, 2024. [Online]. Available: https://github.com/amreis/ShaRP.

[53] I. Loshchilov and F. Hutter, *SGDR: Stochastic Gradient Descent with Warm Restarts*, arXiv:1608.03983 [cs], May 2017. DOI: 10.48550/arXiv.1608.03983. Accessed: Nov. 28, 2025. [Online]. Available: http://arxiv.org/abs/1608.03983.

[54] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.

6980. Accessed: Nov. 28, 2025. [Online]. Available: http://arxiv.org/abs/1412.6980.

[55] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, arXiv:1912.01703 [cs], Dec. 2019. DOI: 10.48550/arXiv.1912.01703. Accessed: Nov. 28, 2025. [Online]. Available: http://arxiv.org/abs/1912.01703.

[56] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*, arXiv:2205.14135 [cs], Jun. 2022. DOI: 10.48550/arXiv.2205.14135. Accessed: Dec. 2, 2025. [Online]. Available: http://arxiv.org/abs/2205.14135.

[57] C. Smet, *Memory-Optimised Projection Quality Metrics*, original-date: 2025-12-05T15:31:51Z, Dec. 2025. Accessed: Dec. 5, 2025. [Online]. Available: https://github.com/Casper-Smet/pytorch-projection-qm.

[58] B. v. Aken, B. Winter, A. Löser, and F. A. Gers, *VisBERT: Hidden-State Visualizations for Transformers*, arXiv:2011.04507 [cs], Nov. 2020. DOI: 10.48550/arXiv.2011.04507. Accessed: Dec. 3, 2025. [Online]. Available: http://arxiv.org/abs/2011.04507.

[59] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, *RoFormer: Enhanced Transformer with Rotary Position Embedding*, arXiv:2104.09864 [cs], Nov. 2023. DOI: 10.48550/arXiv.2104.09864. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2104.09864.

[60] P. Shaw, J. Uszkoreit, and A. Vaswani, *Self-Attention with Relative Position Representations*, arXiv:1803.02155 [cs], Apr. 2018. DOI: 10.48550/arXiv.1803.02155. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/1803.02155.

[61] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context", in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. DOI: 10.18653/v1/P19-1285. Accessed: Dec. 15, 2025. [Online]. Available: https://aclanthology.org/P19-1285/.

[62] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding", in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. Accessed: Dec. 15, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[63] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020, ISSN: 1533-7928. Accessed:

Dec. 15, 2025. [Online]. Available: http://jmlr.org/papers/v21/20-074.html.

[64] Z. Huang, D. Liang, P. Xu, and B. Xiang, *Improve Transformer Models with Better Relative Position Embeddings*, arXiv:2009.13658 [cs], Sep. 2020. DOI: 10.48550/arXiv.2009.13658. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2009.13658.

[65] G. Ke, D. He, and T.-Y. Liu, *Rethinking Positional Encoding in Language Pre-training*, arXiv:2006.15595 [cs], Mar. 2021. DOI: 10.48550/arXiv.2006.15595. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2006.15595.

[66] Z. Liu et al., *Swin Transformer V2: Scaling Up Capacity and Resolution*, arXiv:2111.09883 [cs], Apr. 2022. DOI: 10.48550/arXiv.2111.09883. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2111.09883.

[67] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, *Rethinking and Improving Relative Position Encoding for Vision Transformer*, arXiv:2107.14222 [cs], Jul. 2021. DOI: 10.48550/arXiv.2107.14222. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2107.14222.

[68] B. Heo, S. Park, D. Han, and S. Yun, *Rotary Position Embedding for Vision Transformer*, arXiv:2403.13298 [cs], Jul. 2024. DOI: 10.48550/arXiv.2403.13298. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2403.13298.

[69] B. Heo, S. Park, D. Han, and S. Yun, "Rotary Position Embedding for Vision Transformer", en, in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., vol. 15068, Series Title: Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2025, pp. 289–305, ISBN: 978-3-031-72683-5 978-3-031-72684-2. DOI: 10.1007/978-3-031-72684-2_17. Accessed: Dec. 15, 2025. [Online]. Available: https://link.springer.com/10.1007/978-3-031-72684-2_17.

[70] J. Li, B. Chen, H. Li, Z. Dong, J. Wang, and S. Zhu, *Head-wise Adaptive Rotary Positional Encoding for Fine-Grained Image Generation*, arXiv:2510.10489 [cs], Oct. 2025. DOI: 10.48550/arXiv.2510.10489. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2510.10489.

[71] X. Wu et al., *Point Transformer V3: Simpler, Faster, Stronger*, arXiv:2312.10035 [cs] version: 1, Dec. 2023. DOI: 10.48550/arXiv.2312.10035. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2312.10035.

[72] X. Lai et al., *Stratified Transformer for 3D Point Cloud Segmentation*, arXiv:2203.14508 [cs], Mar. 2022. DOI: 10.48550/arXiv.2203.14508. Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2203.14508.

[73] Z. Li, P. Gao, H. Yuan, R. Wei, and M. Paul, *Exploiting Inductive Bias in Transformer for Point Cloud Classification and Segmentation*, arXiv:2304.14124 [cs], Apr. 2023. DOI: 10.48550/arXiv.2304.14124.

Accessed: Dec. 15, 2025. [Online]. Available: http://arxiv.org/abs/2304.14124.

A. Ethics and Privacy Quick Scan

Response Summary:

Section 1. Research projects involving human participants

P1. Does your project involve human participants? This includes for example use of observation, (online) surveys, interviews, tests, focus groups, and workshops where human participants provide information or data to inform the research. If you are only using existing data sets or publicly available data (e.g. from X, Reddit) without directly recruiting participants, please answer no.

- No

Section 2. Data protection, handling, and storage

The General Data Protection Regulation imposes several obligations for the use of **personal data** (defined as any information relating to an identified or identifiable living person) or including the use of personal data in research.

D1. Are you gathering or using personal data (defined as any information relating to an identified or identifiable living person)?

- No

Section 3. Research that may cause harm

Research may cause harm to participants, researchers, the university, or society. This includes when technology has dual-use, and you investigate an innocent use, but your results could be used by others in a harmful way. If you are unsure regarding possible harm to the university or society, please discuss your concerns with the Research Support Office.

H1. Does your project give rise to a realistic risk to the national security of any country?

- No

H2. Does your project give rise to a realistic risk of aiding human rights abuses in any country?

- No

H3. Does your project (and its data) give rise to a realistic risk of damaging the University's reputation? (E.g., bad press coverage, public protest.)

- No

H4. Does your project (and in particular its data) give rise to an increased risk of attack (cyber- or otherwise) against the University? (E.g., from pressure groups.)

- No

H5. Is the data likely to contain material that is indecent, offensive, defamatory, threatening, discriminatory, or extremist?

- No

H6. Does your project give rise to a realistic risk of harm to the researchers?

- No

H7. Is there a realistic risk of any participant experiencing physical or psychological harm or discomfort?

- No

H8. Is there a realistic risk of any participant experiencing a detriment to their interests as a result of participation?

- No

H9. Is there a realistic risk of other types of negative externalities?

- No

Section 4. Conflicts of interest

C1. Is there any potential conflict of interest (e.g. between research funder and researchers or participants and researchers) that may potentially affect the research outcome or the dissemination of research findings?

- No

C2. Is there a direct hierarchical relationship between researchers and participants?

- No

Section 5. Your information.

This last section collects data about you and your project so that we can register that you completed the Ethics and Privacy Quick Scan, sent you (and your supervisor/course coordinator) a summary of what you filled out, and follow up where a fuller ethics review and/or privacy assessment is needed. For details of our legal basis for using personal data and the rights you have over your data please see the University's privacy information. Please see the guidance on the ICS Ethics and Privacy website on what happens on submission.

Z0. Which is your main department?

- Information and Computing Science

Z1. Your full name:

Casper Willem Smet

Z2. Your email address:

casper.smet@gmail.com

Z3. In what context will you conduct this research?

- As a student for my master thesis, supervised by::
  Telea, A.C.

Z5. Master programme for which you are doing the thesis

- Artificial Intelligence

Z6. Email of the course coordinator or supervisor (so that we can inform them that you filled this out and provide them with a summary):

a.c.telea@uu.nl

Z7. Email of the moderator (as provided by the coordinator of your thesis project):

coordinator-ai-master@uu.nl

Z8. Title of the research project/study for which you filled out this Quick Scan:

Pay Attention to your Neighbours: Leveraging Attention in Dimensionality Reduction

Z9. Summary of what you intend to investigate and how you will investigate this (200 words max):

In this project i will develop a transformer-based approach for learning projection techniques like T-SNE and their inverse projections. Unlike previous neural methods, which process samples individually and rely on network weights to encode distribution, this method treats the dataset as a sequence, enabling samples to explicitly attend to their neighbours.
This approach aims to produce cleaner, more generalizable data representations and improved projection quality, particularly on unseen data. Validation will involve comparing this method against standard metrics and benchmarks from prior research. The project will deliver a Python module implementing the model and supporting tools to ensure replicability and foster further experimentation.

Z10. In case you encountered warnings in the survey, does supervisor already have ethical approval for a research line that fully covers your project?

- Not applicable

Scoring

- Privacy: 0
- Ethics: 0

B. Global and Local Attention Patterns on Other Datasets

B.1 fMNIST

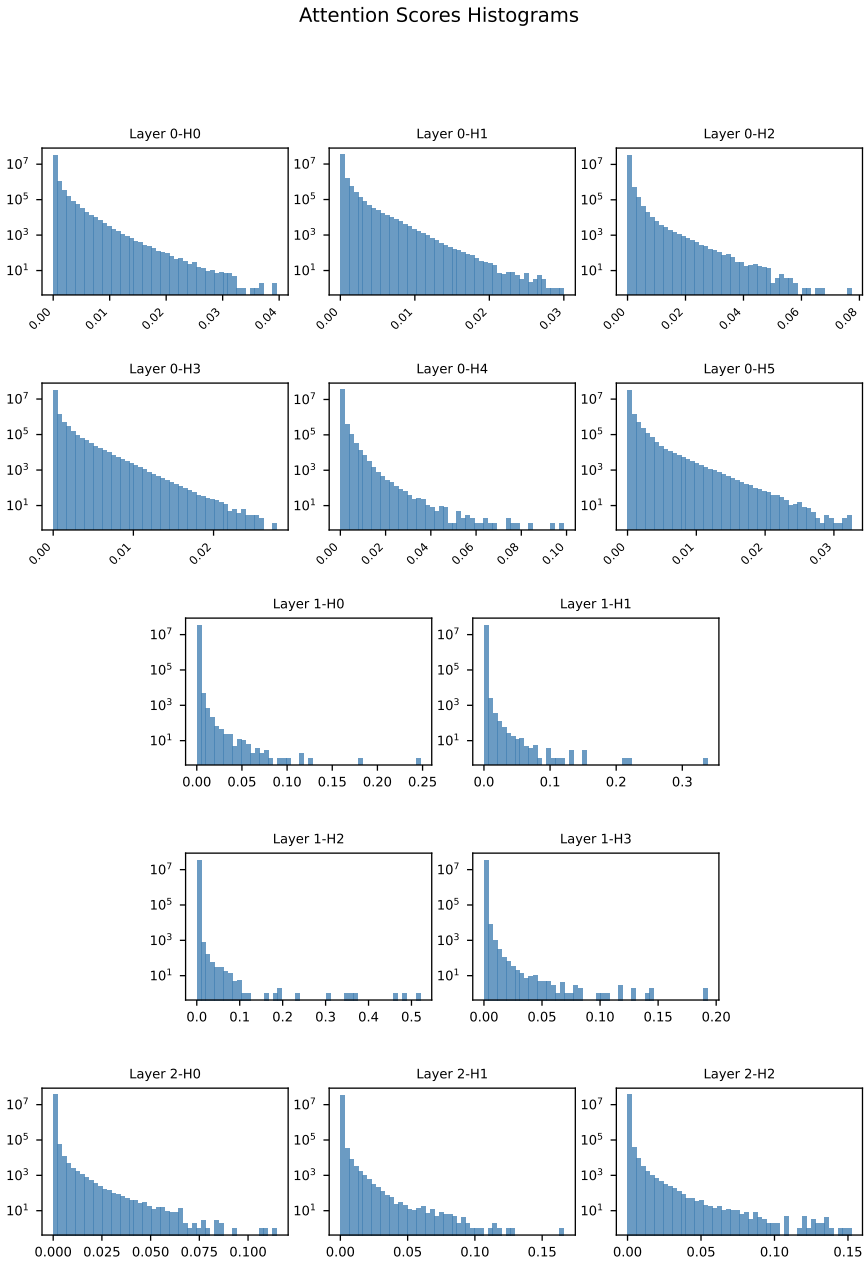
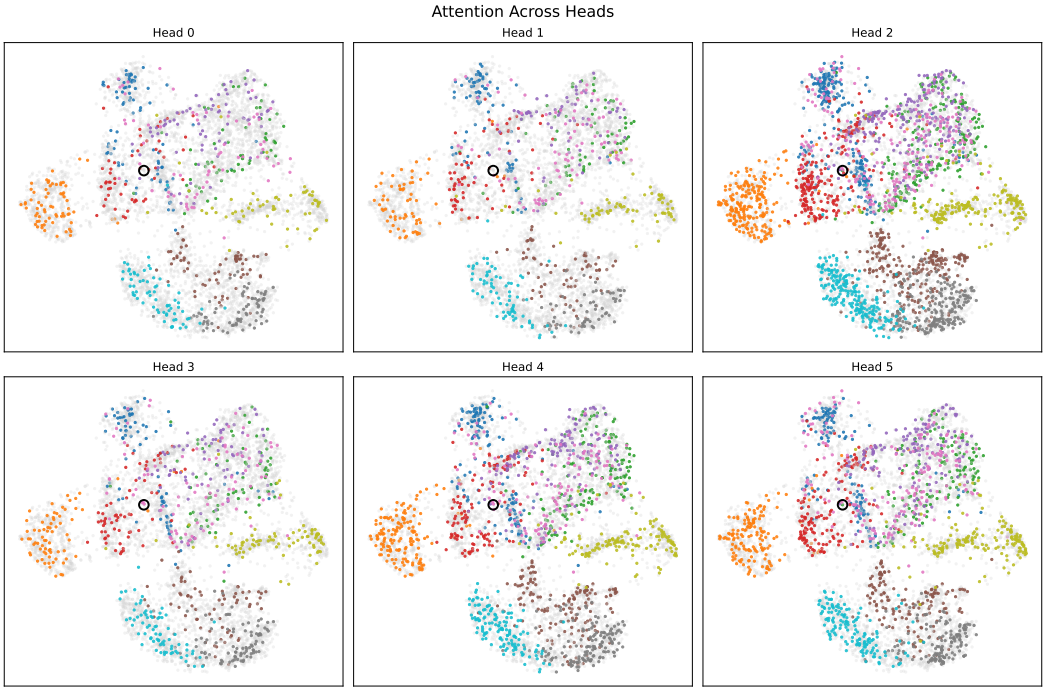
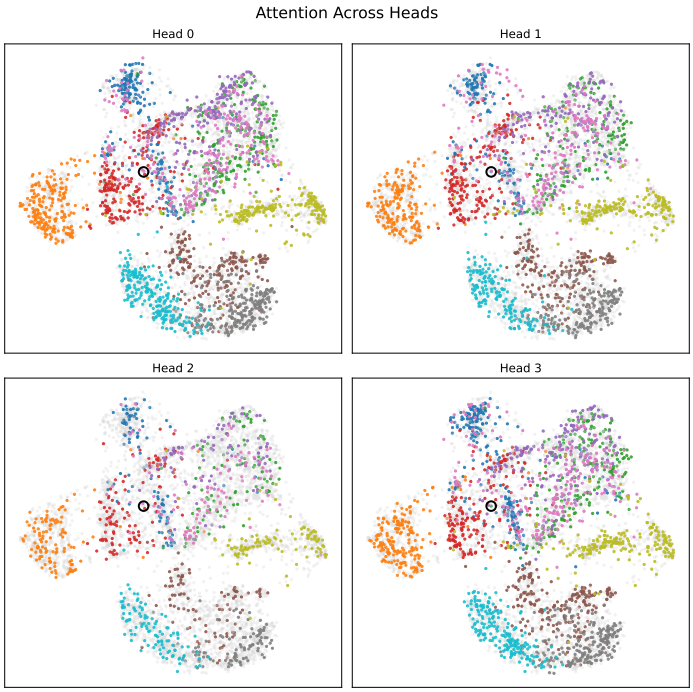


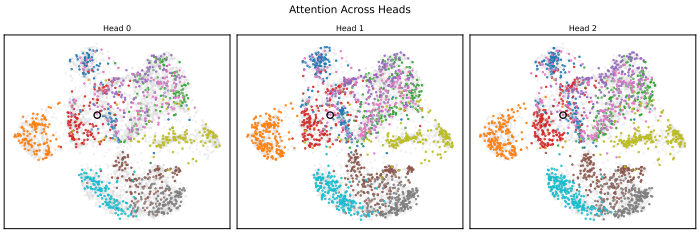
Figure B.1: Attention weight histograms per head per layer for fMNIST, log-93 scaled. H1 is head 1, H2 is head2, etc.



(a) Layer one.



(b) Layer two.



(c) Layer three.

Figure B.2: Attention weights for the first, second, and third layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out.

B.2 Spambase

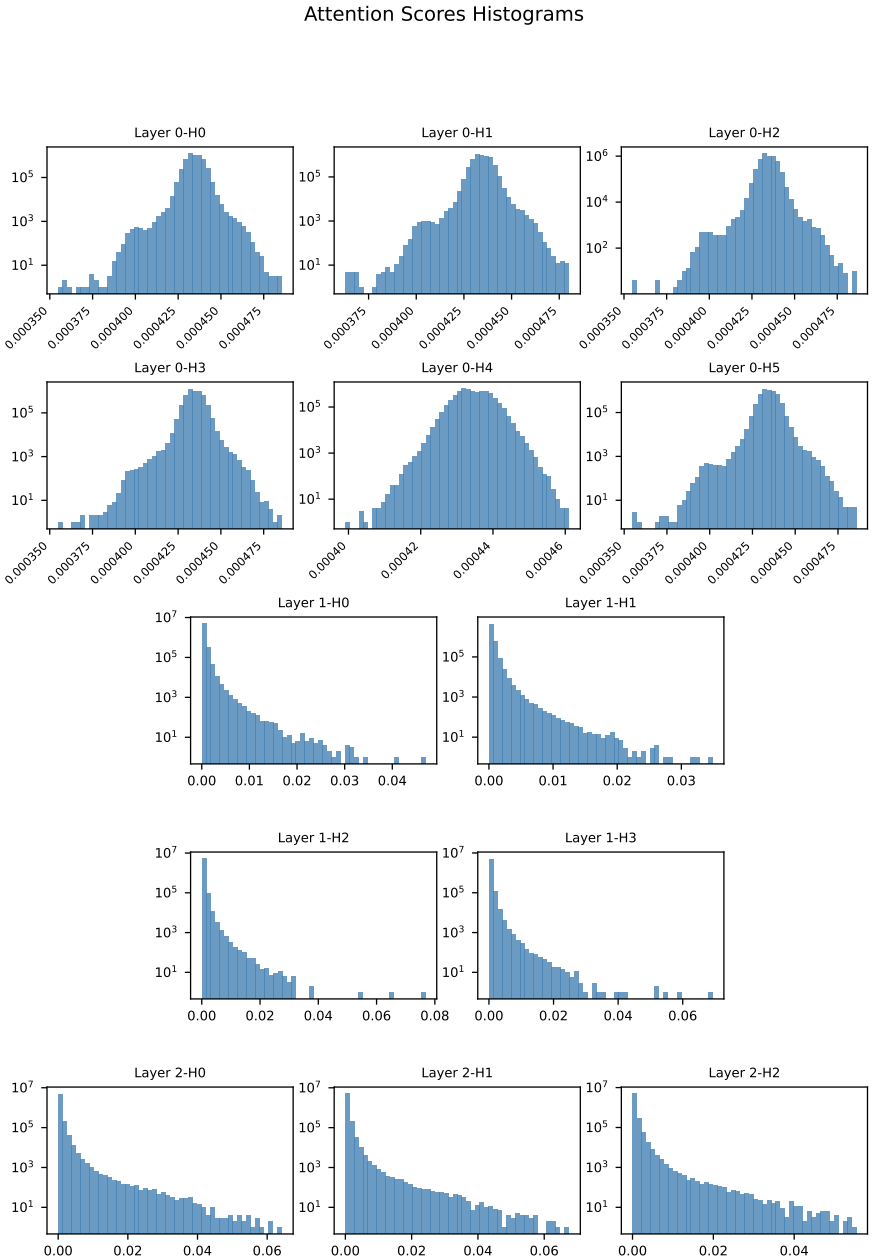
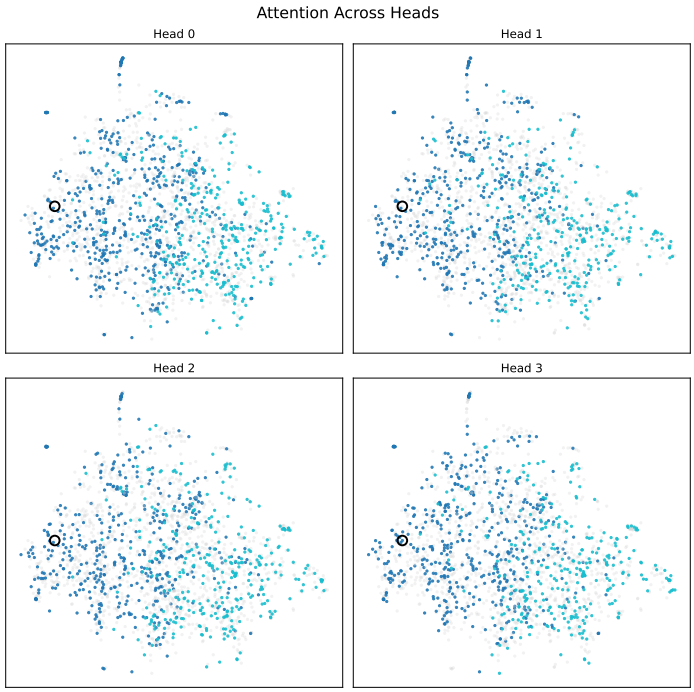
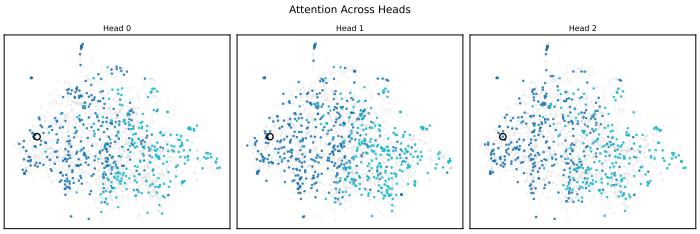


Figure B.3: Attention weight histograms per head per layer for Spambase, log-scaled. H1 is head 1, H2 is head2, etc.

(a) Layer one.



(b) Layer two.



(c) Layer three.

Figure B.4: Attention weights for the first, second, and third layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out.

B.3 HAR

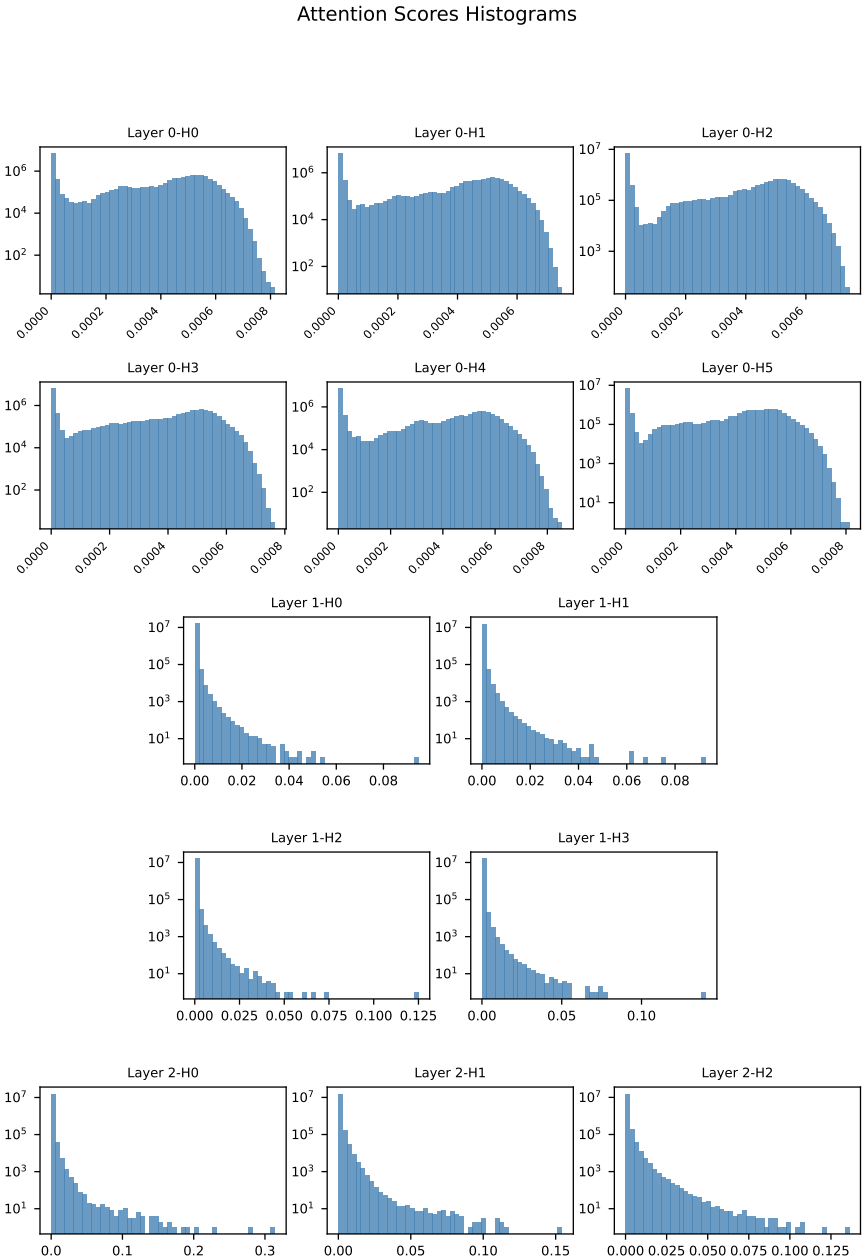
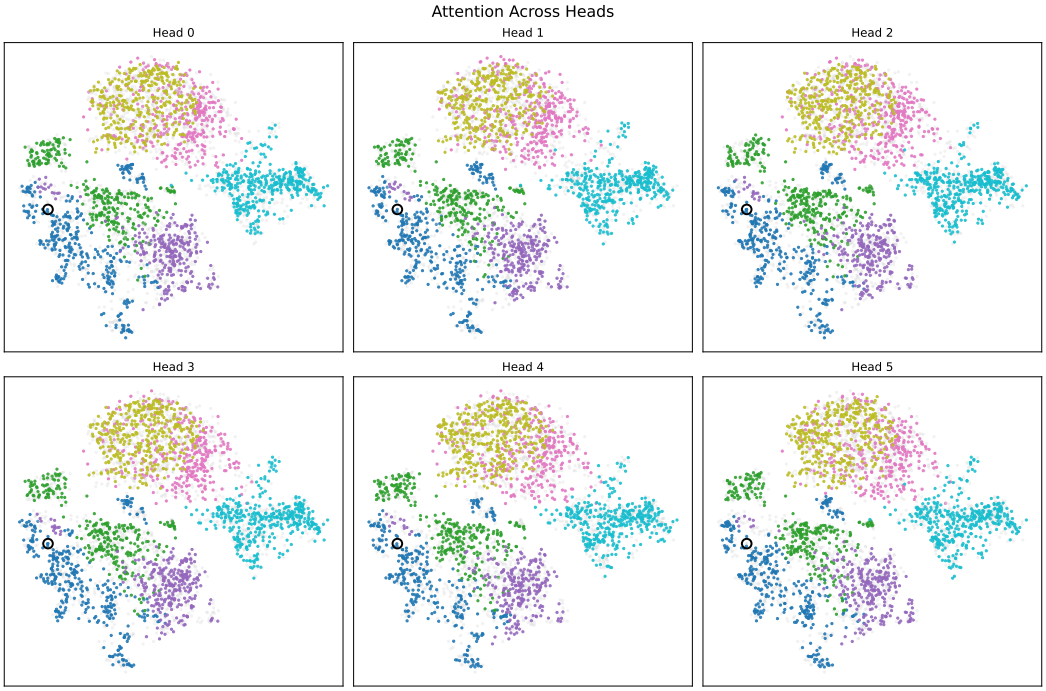
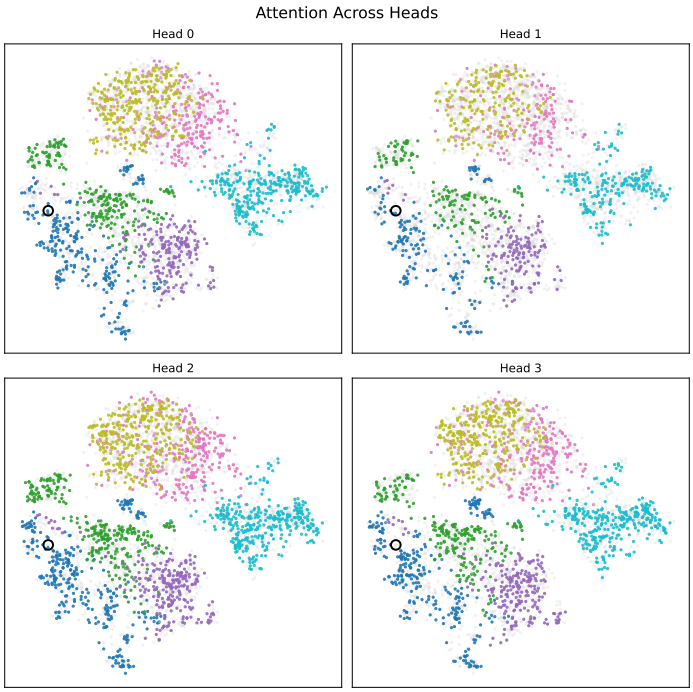


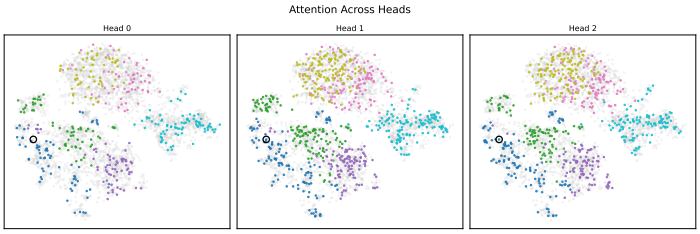
Figure B.5: Attention weight histograms per head per layer for HAR, log-scaled. H1 is head 1, H2 is head2, etc.



(a) Layer one.



(b) Layer two.



(c) Layer three.

Figure B.6: Attention weights for the first, second, and third layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out.

B.4 CIFAR-10

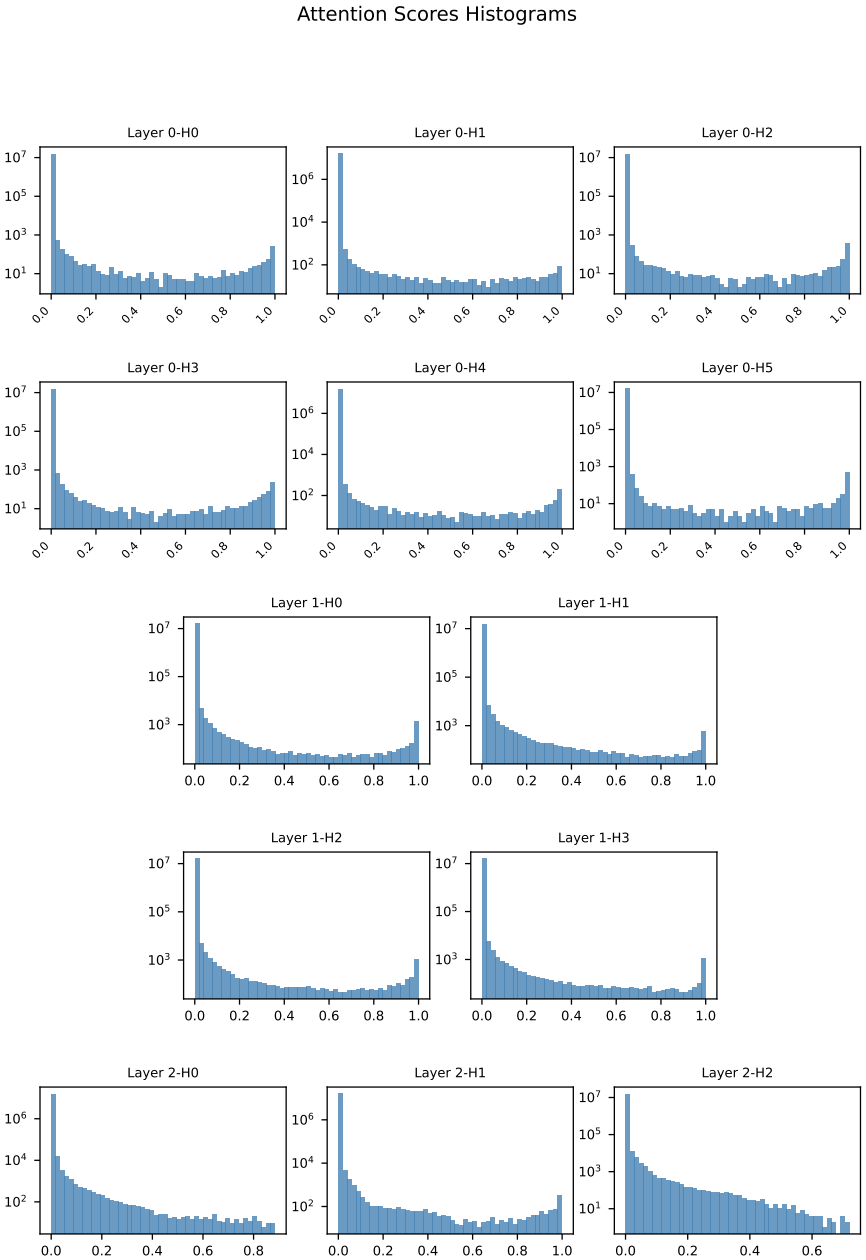
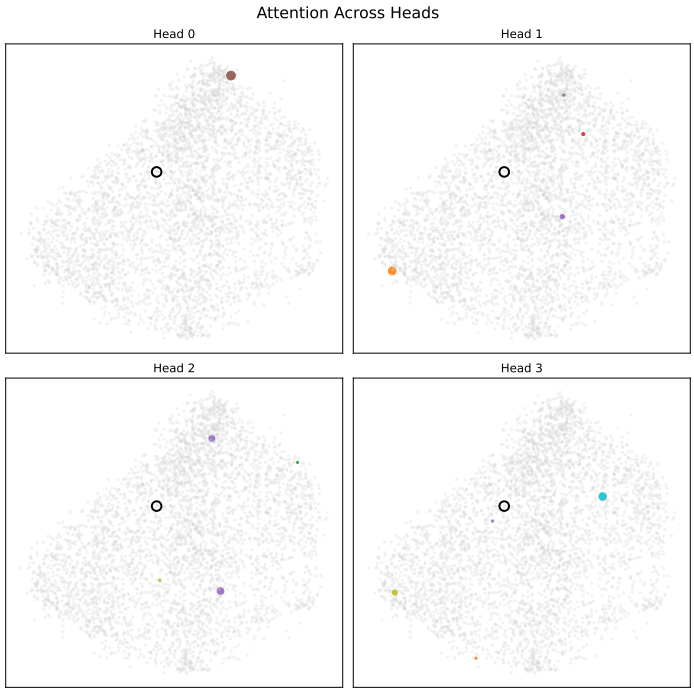
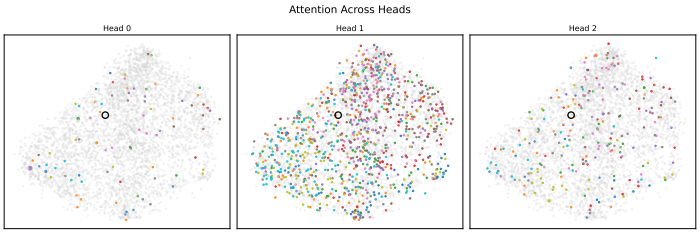


Figure B.7: Attention weight histograms per head per layer for CIFAR-10, log-scaled. H1 is head 1, H2 is head2, etc.

(a) Layer one.



(b) Layer two.



(c) Layer three.

Figure B.8: Attention weights for the first, second, and third layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out.

B.5 CNAE-9

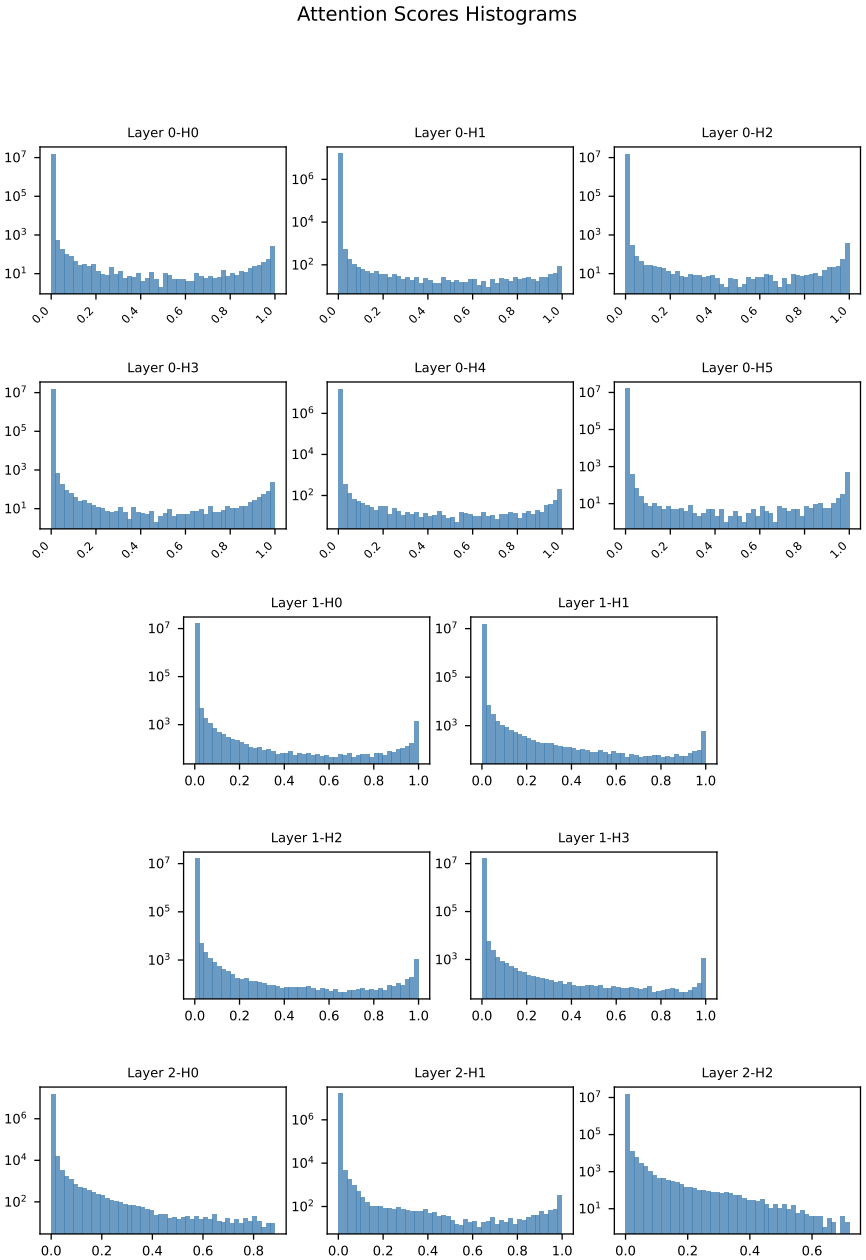
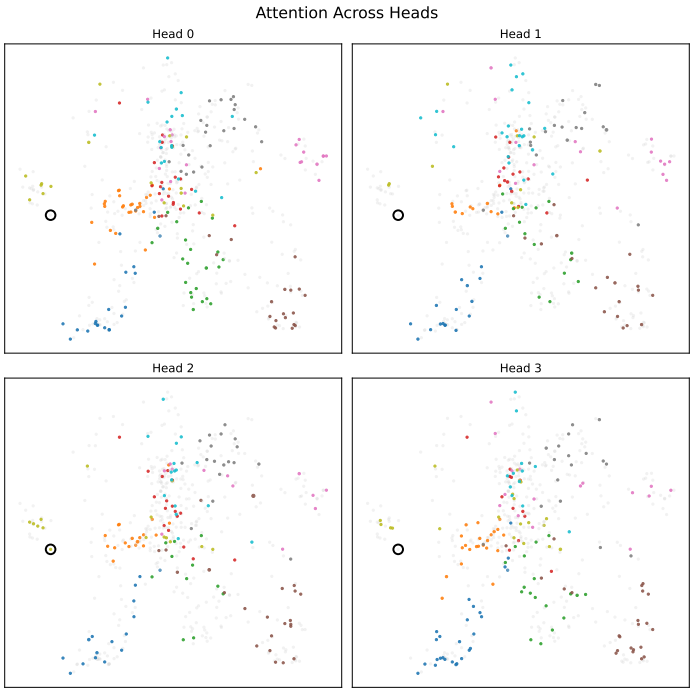
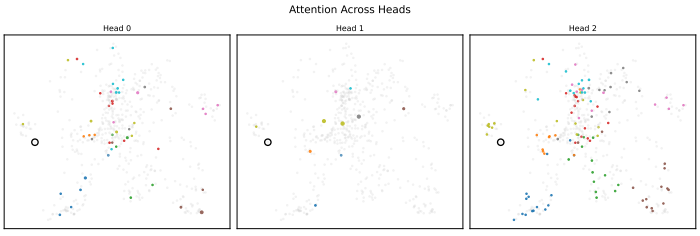


Figure B.9: Attention weight histograms per head per layer for CNAE-9, log-scaled. H1 is head 1, H2 is head2, etc.

(a) Layer one.



(b) Layer two.



(c) Layer three.

Figure B.10: Attention weights for the first, second, and third layer visualised for the sample circled in black. Samples are sized according to total weight, and samples with a weight bellow the threshold are greyed out.