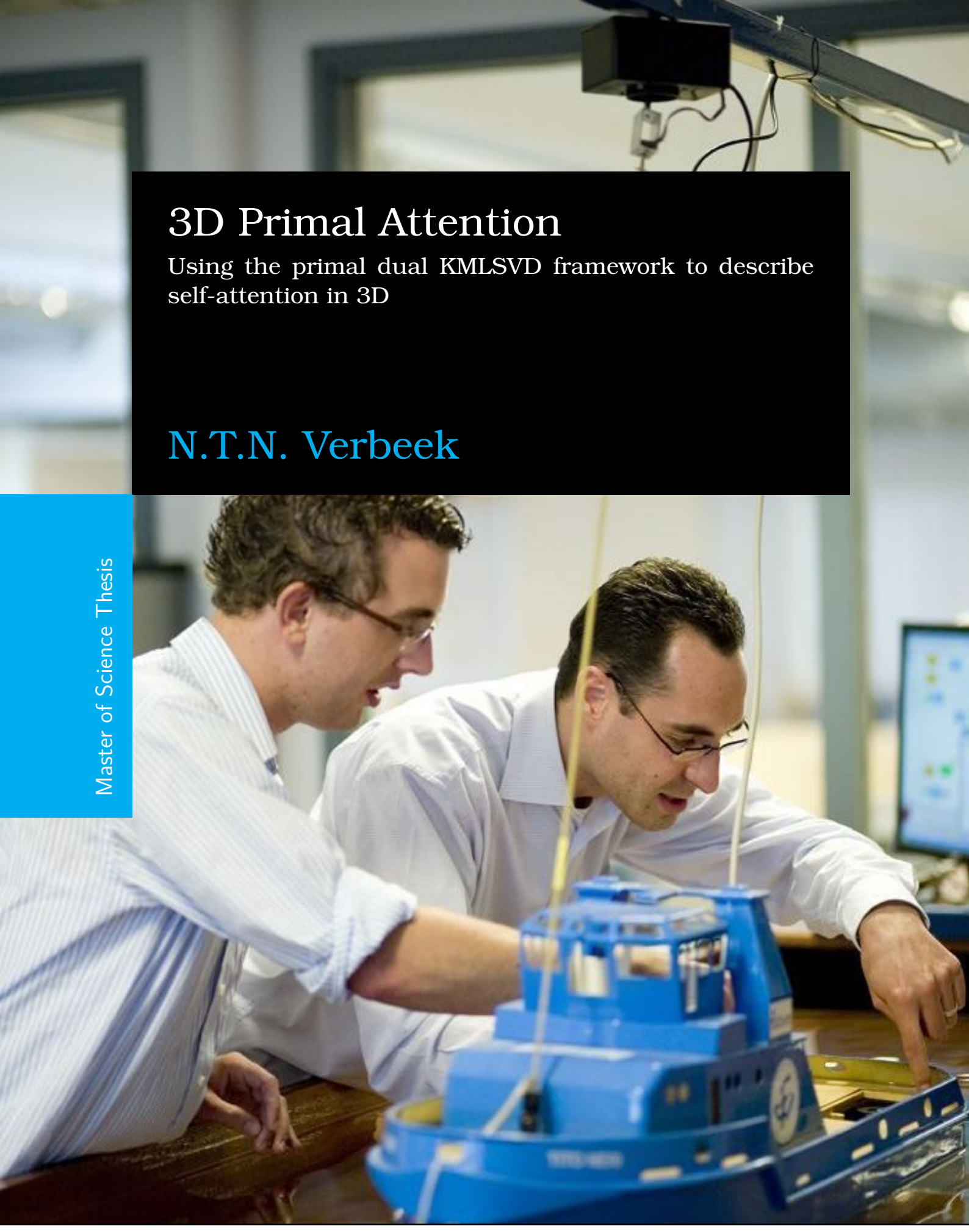


3D Primal Attention

Using the primal dual KMLSVD framework to describe self-attention in 3D

N.T.N. Verbeek

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

3D Primal Attention

Using the primal dual KMLSVD framework to describe self-attention in 3D

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

N.T.N. Verbeek

November 14, 2025

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

The self-attention mechanisms play a crucial role in multiple applications, for example modern large language models (LLMs), but their growing adoption has led to rapidly increasing energy, water, economic, and hardware demands[25][20][5]. This thesis examines the application of the Primal-Dual Kernel Multi-linear Singular Value Decomposition (KMLSVD) framework as introduced by Wesel and Batselier[3] on the self-attention mechanism[24]. The Primal-Dual KMLSVD attention framework makes three-dimensional self-attention possible enabling a more information rich representation, possibly increasing accuracy, computation time and/or a decrease in energy and hardware requirements. Furthermore, the Primal formulation does not compute the attention tensor, significantly decreasing the computational and time complexity. Therefore, Primal-Dual KMLSVD attention could play a major role in green AI applications.

Three tests are performed on 10 different timeseries datasets in order to: i) find the most accurate Primal KMLSVD attention variant, ii) compare Primal to Dual KMLSVD attention and iii) compare the Primal-Dual KMLSVD attention framework to primal attention[4] and canonical attention[24]. The results of these tests prove that Primal-Dual KMLSVD can define self-attention in 3D but, as of writing this thesis, the current used formulations are too inefficient time wise to be a valid improvement or alternative to self-attention. Furthermore, small scale tests suggest that Primal-Dual KMLSVD might not even be required to define self attention in 3D. However, as no experiments were performed on higher-order (3D) datasets, the potential of this framework for such problems remains an open question.

Table of Contents

Preface

This thesis "3D Primal attention, Using the Primal-Dual KMLSVD framework to describe self-attention in 3D" was written as part of the master Systems and Control at 3ME TU Delft. The writing of this thesis spanned from 18 November 2024 to 14 October 2025. This thesis is a final product of my studies at delft. This thesis is meant for any student interested in green AI or self-attention. The energy consumption of current commercial and popular self-attention applications is unsustainable. The hope is that by researching this topic, we find a possible solution to this problem.

Acknowledgments

I would like to thank my supervisor, professor Kim Batselier for helping me throughout the process of writing my thesis. Without his input and feedback, this thesis would not have been possible. I would also like to thank the committee members Albert Saiapin and Amin Sharifi Kolarijani for taking time out of their schedules to read my rather sizable thesis and be present during my defense.
I would also like to thank my fellow student Tom Lijding, whose help and input improved the efficiency of my code and gave me good ideas to implement. Without his help, it would not have been possible to have such a streamlined, fast, and (relatively) neat code. Finally, I would like to thank my parents for helping me with the writing process of this thesis.

Chapter 1

Introduction

Self attention is a relatively new and powerful machine learning concept that is able to efficiently encode and learn contextual relationships within sequences of data[24]. Self-attention is commonly used in applications like Large Language Models (LLM)s (e.g., ChatGPT, Claude etc.) and computer vision based tasks like medical imaging, image recognition and image generation[17][27][28]. Self-attention, as defined by Vaswani et al.[24], is inherently two-dimensional and computationally heavy for large datasets. The two-dimensional nature of canonical self-attention possibly limits its expressive power. Additionally, the implementation of self-attention on especially LLMs requires large data centers [15]. These centers use disproportional amounts of energy and water for training and cooling[25][20]. The LLM's GPT-3 for instance released over 500 metric tons of carbon during the training of the model[1]. The aforementioned LLMs also come with a high and ever-increasing monetary cost to train and maintain[5]. GPT-4 for instance cost a staggering $40M for a single training run[5]. Besides the energy use, high and rising monetary cost and greenhouse gas emissions, self-attention suffers from performance drops as the sequence length of the data it processes increases[8]. These challenges highlight the need for a more efficient and expressive self-attention formulation. To address these structure and computational limitations, this thesis introduces a novel three-dimensional self-attention mechanism based on the primal-dual Kernel Multi-linear Singular Value Decomposition (KMLSVD) framework. Unlike conventional self-attention definitions which stay in 2D [24][4][12], the proposed method operates in 3D, enabling a richer and more compact representation of contextual dependencies. This richer and more compact representation has the potential to reduce computational complexity, improve the scalability to high dimensional data, and alleviate the environmental burden associated with large-scale training. It could also open the door for novel transformer designs made specifically for 3D datasets, thereby significantly increasing the model's performance on said 3D datasets. Furthermore, this formulation offers flexibility in computing 3D self-attention, where the primal formulation can help reduce the impact of sequence length on model performance. However, before investigating the implementation and design of these transformers and promises, it is crucial to determine whether Primal-Dual KMLSVD attention actually works as a self-attention mechanism in the first place. This last task is undertaken in this thesis.
This thesis aims to answer the central research question:

- "Can 3D Primal-Dual KMLSVD attention actually function as a self-attention mechanism?".

Besides this main question, this thesis also tries to answer the sub-questions:

- "Can Primal-Dual KMLSVD attention be used as an improvement or alternative to self-attention?"
- "Is the Primal or Dual KMLSVD attention formulation more favorable in terms of computational efficiency and predictive performance?"
- "Can alternative primal KMLSVD attention formulations and feature maps improve training efficiency and predictive accuracy compared to the baseline approach?".

This thesis will proceed as follows. In chapter 2 the required background information is discussed. Chapter 3 introduces the Primal-Dual KMLSVD attention framework and presents some possible modifications to the Primal KMLSVD attention. In chapter 4, the test results of three tests are shown, and their findings discussed along with some smaller scale tests. Finally, in chapter 5 a conclusion and discussion based on the test results are made along with possible future research ideas.

Chapter 2

Background

This chapter covers the existing background information that is required to understand Primal-Dual Kernel Multi-linear Singular Value Decomposition (KMLSVD) self-attention. Commonly used concepts in this thesis are explained in section 2-1. The Singular Value Decomposition (SVD) and its primal-dual (re-)formulation will be discussed in section 2-2. The Multi-linear Singular Value Decomposition (MLSVD) and its primal-dual (re-)formulation will be explained in section 2-3. The self-attention mechanism is introduced in section 2-4. In section 2-5 the concept of using primal-dual Kernel Singular Value Decomposition (KSVD), as introduced by Chen et al.[4], to represent and improve self-attention is introduced. Finally, in section 2-6 all the theoretical components of this chapter are used to define an implementation method of primal attention.
Throughout the thesis, the following notation is used:

- Lower case variables x denote a vector.
- Upper case variables X denote a matrix.
- Calligraphic upper case variables $\mathcal{X}$ denote a tensor (see section 2-1 for what a tensor is).
- $X_{(n)}$ denotes the mode-n-unfolding of tensor $\mathcal{X}$ (see section 2-1 for what a mode-n-unfolding is).
- $\mathcal{A} \times_n B$ denotes a mode-n-product between the $\mathcal{A}$ tensor and the B matrix (see section 2-1 for what a mode-n-product is).

2-1 Terms and Definitions

Throughout this thesis, various definitions and terms are used, which are defined and explained in this section. This section can be skipped if the reader has basic knowledge of tensors, tensor networks, and tensor diagrams.

First off, a **tensor** represents any multidimensional array or data structure like a vector (1D) or a matrix (2D). A tensor can have any number of dimensions, e.g., anything ranging from $t \in \mathbb{R}^N$ to $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times ... \times N_\infty}$ is a tensor. A good example of a tensor is an RGB (red, green, blue) color image $\mathcal{T}_{RGB} \in \mathbb{R}^{N_y \times N_x \times 3}$. N_x and N_y are the image pixel width and length respectively (i.e., image resolution). Due to it being an RGB image, each pixel is encoded by a vector $t_{RGB} \in \mathbb{R}^3$ that encodes the amount of red, blue and green in said pixel.
The **order** of a tensor is the same as the dimensionality of the tensor. So, a tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times ... \times N_D}$ has order D.
A **mode** is a specific dimension of the tensor. For example, the first, second and third mode of the tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ are of sizes N_1, N_2 and N_3 respectively. In the example of the RGB image tensor $\mathcal{T}_{RGB}$, the first, second and third mode are N_y, N_x and 3 respectively.
A **mode-n-matricization** refers to different ways to flatten a tensor into a matrix along the n'th mode. For example, given a tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, taking the mode-2-matricization results in a matrix $T_{(2)} \in \mathbb{R}^{N_2 \times N_1 N_3}$. See figure 2-1 for a visual example of the different ways to apply a mode-n-matricization on a 3D tensor.
A **mode-n-product** denotes different ways to compute a tensor-matrix multiplication. In practice, a mode-n-product involves multiplying a matrix with the transposed mode-n-matricization of the tensor and them permuting and reshaping the product back into a tensor. For example, given tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and matrix $A \in \mathbb{R}^{M_1 \times N_1}$, the mode-2-product between $\mathcal{T}$ and A (see equation 2-1) will follow these steps. First, take the mode-2-matricization of the tensor: $T_{(2)} \in \mathbb{R}^{N_2 \times N_1 N_2}$. Then, compute $AT_{(2)}^\top$, and finally permute and reshape the result into $\mathbb{R}^{N_1 \times M_1 \times N_3}$. Note that the second mode (the mode along which the multiplication took place) has now changed.

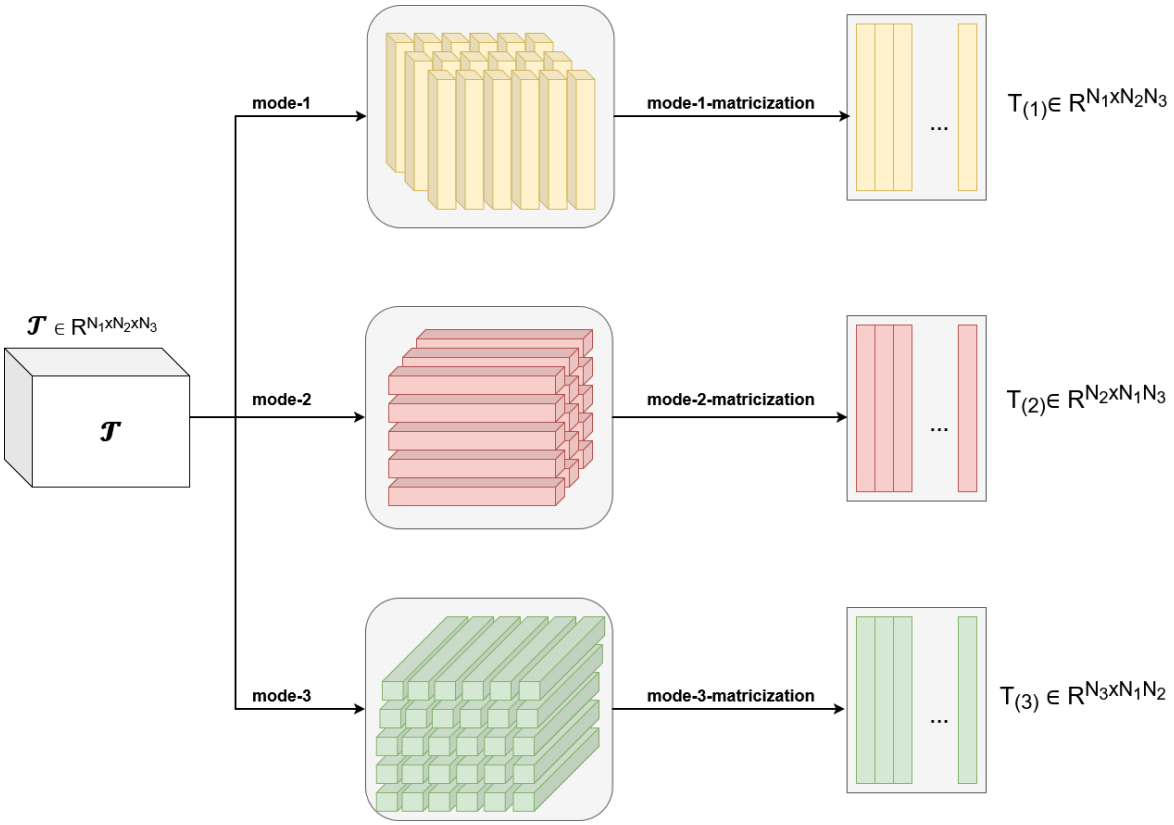


Figure 2-1: Visual representation of a mode 1, mode 2 and mode 3 matricization of a 3D tensor. This image is heavily inspired by C.Zetai and L.Clifford [6]

A **Tensor Network Diagram (TND)** is a visual representation of tensors and tensor multiplications.

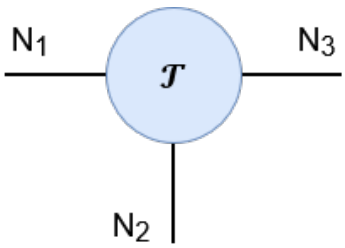


Figure 2-2: Tensor network representation of a 3D tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$

The "arms" in figure 2-2 correspond to the different dimensions. Each arm represents a separate dimension and as such, the amount of arms connected to a tensor determine the dimensionality (e.g., 3 arms means a 3D tensor, 6 arms means a 6D tensor etc.). Besides representing dimensionality, the connection of two or more arms from different tensor represent a dimensional contraction or a mode-n-multiplication. Figure 2-3 shows the mode-2-multiplication of a 3D tensor $T \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ with a matrix $A \in \mathbb{R}^{M \times N_2}$.

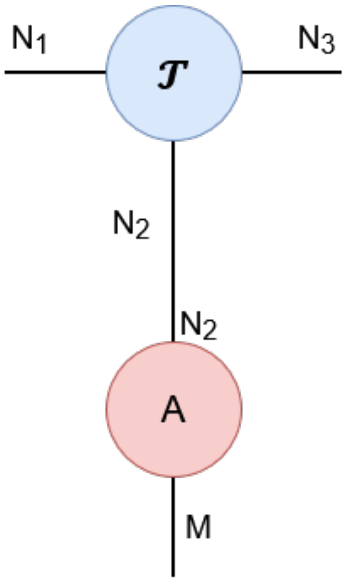


Figure 2-3: Tensor diagram of a mode-2-multiplication between $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $A \in \mathbb{R}^{M \times N_2}$

Note the interconnection of the arm corresponding to the 2nd mode of the $\mathcal{T}$ tensor with the A matrix arm. Also note that when two arms are interconnected, these dimensions "disappear" or are contracted. These tensor network diagrams can also be expressed mathematically. The equations corresponding to figure 2-3 is as follows.

$$\mathcal{T} \times_2 A, \tag{2-1}$$

where $\times_2$ denotes the mode-2-multiplication.

A feature map and kernel function are important tools used in machine learning applications to improve accuracy and expressive power. A feature map is an explicit mapping from the input space to a (possibly higher-dimensional) feature space. An example of a feature-map is shown in equation 2-2. There, $X \in \mathbb{R}^{N \times M}$ is the input matrix and $\Phi(X)$ is the feature map.

$$\Phi(X) = \begin{bmatrix} X \\ XX^\top \end{bmatrix} \tag{2-2}$$

A kernel function is a function that computes the inner product of two feature vectors in feature space, without the explicit computing of the mapping Φ. The kernel allows the algorithm to implicitly work in the (possibly higher-dimensional) feature space. An example of a kernel function is the polynomial kernel(2-3) where $x \in \mathbb{R}^M$ and $x' \in \mathbb{R}^M$ are two separate entries of the $X \in \mathbb{R}^{N \times M}$ dataset.

$$k(x, x') = (x^\top x' + 1)^2 \tag{2-3}$$

2-2 Primal-dual (K)SVD

The Singular Value Decomposition (SVD) is a linear matrix decomposition method that decomposes a given matrix $X \in \mathbb{R}^{N \times M}$ into a positive diagonal core matrix $S \in \mathbb{R}^{R \times R}$ and

orthogonal factor matrices $U \in \mathbb{R}^{N\times R}$, $V \in \mathbb{R}^{M\times R}$ that admit the following relation

$$\begin{aligned} X &= USV^\top \\ SVD(X) &= [U, S, V] \end{aligned} \tag{2-4}$$

where R denotes the rank of the given matrix X[23]. The U, S and V matrices have the following properties with $I \in \mathbb{R}^{R\times R}$:

- $S := \text{diag}(\sigma_1, ..., \sigma_R) \quad \text{with } \sigma_n \geq 0, \quad \forall n \in (0, ..., R)$
- $U^\top U = I \quad$ (orthogonality),
- $V^\top V = I \quad$ (orthogonality).

The SVD maximizes the captured variance of the matrix that is being decomposed. This means that the SVD allows for an approximate reconstruction of the original dataset matrix X with reduced rank, while preserving as much of its variance or original information as possible. By selecting the first $n \leq R$ singular values and the corresponding singular column vectors of U and row vectors of $V^\top$ a low-rank approximation of matrix X (X_K in equation (2-5)) can be made

$$X_K = \sum_{n=1}^{K} \sigma_n \mathbf{u}_n \mathbf{v}_n^\top. \tag{2-5}$$

This truncated reconstruction retains the directions of maximum variance in the data, since the singular values σ_n are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R$. Each term $\sigma_n \mathbf{u}_n \mathbf{v}_n^\top$ contributes successively less to the overall variance of X. This ability to approximate a matrix using a low-rank reconstruction can be very significant, especially in low-rank matrices. A low-rank matrix can, with the help of SVD, be entirely described using fewer elements than before, easing up on the memory or computational requirements[19].
SVD can be defined with Lanczos' Decomposition Theorem[9].

Theorem 2.1 (Lanczos decomposition for matrices)**.** *Let $X \in \mathbb{R}^{N\times M}$ be a rank-R matrix. Consider the solution (U, V, S) satisfying:*

$$\begin{aligned} XV &= US, \\ X^\top U &= VS, \end{aligned} \tag{2-6}$$

where $S \in \mathbb{R}^{R\times R}$ is a positive diagonal matrix. Then, the matrices $U \in \mathbb{R}^{N\times R}$ and $V \in \mathbb{R}^{M\times R}$ are orthogonal, and X admits the decomposition:

$$X = USV^\top. \tag{2-7}$$

In other words, the SVD matrices U, S and V can be computed by solving the constrained equation 2-6.

As mentioned previously, SVD can be recast into a primal-dual optimization problem following the LS-SVM framework[22][10]. The resulting primal-dual SVD formulation on a given matrix $X \in \mathbb{R}^{N\times M}$ has the following form. Let $W_1 \in \mathbb{R}^{M\times R}$ and $W_2 \in \mathbb{R}^{N\times R}$ denote learnable weight matrices, $E_1 \in \mathbb{R}^{N\times R}$ and $E_2 \in \mathbb{R}^{N\times R}$ denote equality conditions and let $S \in \mathbb{R}^{R\times R}$

and $C \in \mathbb{R}^{M \times N}$ be given

$$\begin{aligned} & \textbf{primal:} \\ \max_{W_1, W_2, E_1, E_2} \quad & \mathcal{J}(W_1, W_2, E_1, E_2) := \\ & \frac{1}{2} \sum_{d=1}^{2} \mathrm{Tr}(E_d S^{-1} E_d^\top) - \mathrm{Tr}(W_2^\top C^\top W_1) \\ \text{subject to:} \quad & E_1 = XCW_2, \\ & E_2 = X^\top C^\top W_1, \end{aligned} \tag{2-8}$$

(2-9)

with R denoting the rank of the matrix X and "Tr" denoting the trace operation. In this formulation, E_1 and E_2 ensure the weight matrices W_1 and W_2 maximize the variance on the dataset X.
The dual expression of the primal is as follows:

$$\begin{aligned} \textbf{dual:} \\ XCXU_2 = XU_2 = U_1 S, \\ X^\top C^\top X^\top U_1 = X^\top U_1 = U_2 S, \end{aligned} \tag{2-10}$$

with $U_1 \in \mathbb{R}^{N \times R}$, $U_2 \in \mathbb{R}^{M \times R}$ both orthogonal matrices. For a complete derivation of the dual, see Suykens [23]. The dual expression(2-10) exactly corresponds to Lanczos' decomposition theorem 2.1 if the following properties hold:

1. the S matrix of equation 2-10 is a positive diagonal matrix. In other words, the S matrix of equation 2-10 should be exactly equal to the S matrix of the SVD decomposition.
2. If $XCX = X$ holds. This is a more trivial condition, seeing as the choice of C is entirely free and can specifically be computed in order for $XCX = X$ to hold.

If these two conditions hold, the dual exactly corresponds to Lanczos' Decomposition Theorem 2.1 meaning that the dual formulation(2-10) is the same as the "classic" definition of SVD. This also means that the primal formulation(2-8) is a definition of the SVD. After all, the primal and the dual are two different ways of solving the same problem.
It is important to note that the orthogonality conditions of U_1 and U_2 do not have to be actively enforced if the above two properties hold. The orthogonality property is merely a result of the dual now being the same as Lanczos' decomposition theorem.

Non-Linearization

The strength of the primal-dual formulation of SVD lies in its capacity for non-linear extension. In the primal this can be done by applying non-linear feature maps $\Phi_1 \in \mathbb{R}^{N_1 \times M_1}$ and $\Phi_2 \in \mathbb{R}^{N_2 \times M_2}$ on the dataset $X \in \mathbb{R}^{N \times M}$. In the dual, this translates to applying the SVD non-linear Kernel matrix $K \in \mathbb{R}^{N_1 \times N_2}$ instead of X, making it a Kernel Singular Value Decomposition (KSVD). This non-linearization effectively transforms SVD into its non-linear

counterpart "Kernel SVD (KSVD)". The resulting primal-dual KSVD formulation has the following form

$$\textbf{primal:} \tag{2-11}$$

$$\begin{aligned} \max_{W_1,W_2,E_1,E_2} \quad & \mathcal{J}(W_1,W_2,E_1,E_2) := \\ & \frac{1}{2}\sum_{d=1}^{2} \mathrm{Tr}(E_d \hat{S}^{-1} E_d^\top) - \mathrm{Tr}(W_2^\top C^\top W_1) \\ \text{subject to:} \quad & E_1 = \Phi_1 C W_2, \\ & E_2 = \Phi_2 C^\top W_1, \end{aligned} \tag{2-12}$$

where $W_1 \in \mathbb{R}^{M_1 \times R}$, $W_2 \in \mathbb{R}^{M_2 \times R}$, $E_1 \in \mathbb{R}^{N_1 \times R}$, $E_2 \in \mathbb{R}^{N_2 \times R}$, $S \in \mathbb{R}^{R \times R}$ and $C \in \mathbb{R}^{M_1 \times M_2}$. Here, R denotes the rank of the kernel matrix $K \in \mathbb{R}^{N_1 \times N_2}$, "Tr" denotes the trace operation and M_1 and M_2 correspond to the dimensionality of the non-linear feature maps Φ_1 and Φ_2 respectively. The two feature maps can have a different dimensionality than the original dataset X.
The dual expression is as follows:

$$\textbf{dual:} \tag{2-13}$$

$$\begin{aligned} \overbrace{\Phi_1 C \Phi_2^\top}^{K} U_2 &= K U_2 = U_1 S, \\ \underbrace{\Phi_2 C^\top \Phi_1^\top}_{K^\top} U_1 &= K^\top U_1 = U_2 S, \end{aligned}$$

$$\text{with:} \quad K = \Phi_1 C \Phi_2^\top. \tag{2-14}$$

where $U_1 \in \mathbb{R}^{N_1 \times R}$ and $U_2 \in \mathbb{R}^{N_2 \times R}$ both are orthogonal matrices. See Suykens [23] for a detailed derivation of the primal-dual KSVD framework.

The dual corresponds to Lanczos' decomposition theorem2.1 if the following relations hold:

1. S is a positive diagonal matrix (i.e., an S matrix corresponding to the SVD decomposition2.1).
2. $\Phi_1 C \Phi_2 = K$

With the above two conditions met, the dual corresponds to theorem 2.1 and as such is a definition of an SVD on a Kernel matrix K instead of the original matrix X. This in turn means that the primal is also a definition of KSVD.
The C-matrix (also known as the compatibility matrix) in the above primal-dual KSVD definition allows for a (possible) mismatch in dimensionality of the two $\Phi_1 \in \mathbb{R}^{N_1 \times M_1}$ and $\Phi_2 \in \mathbb{R}^{N_2 \times M_2}$ feature maps for constructing the kernel matrix $K \in \mathbb{R}^{N_1 \times N_2}$. The C matrix should be designed in such a way that equation 2-14 holds.
Recall that previously it was mentioned that the core S matrix should be the same core S matrix of the SVD decomposition. In the dual, the core matrix S is gained by solving the set

of equations (i.e., by solving Lanczos' Decomposition Theorem 2.1). However, in the primal, the core S matrix cannot be gained by solving the optimization problem and is therefore assumed to be previously known. This however is paradoxical: in order to solve the primal KSVD formulation the S-matrix should be known beforehand however, the S-matrix can only be gained by solving the KSVD. This creates a "chicken or the egg" problem, in order to solve the primal formulation S should be known, but S can only be known if the SVD is solved. The "chicken and egg" problem can be addressed by also considering the core S matrix as an optimization matrix/variable[4].
The E_1 and E_2 matrices can also be expressed in either the dual or the primal.

$$\begin{aligned} &\textbf{primal:} && \textbf{dual:} \\ &E_1 = \Phi_1 C W_2 && E_1 = \overbrace{\Phi_1 C \Phi_2^\top}^{K} U_1 = K U_1 \\ &E_2 = \Phi_2 C^\top W_1 && E_2 = \overbrace{\Phi_2 C^\top \Phi_1^\top}^{K^\top} U_2 = K^\top U_2 \end{aligned} \tag{2-15}$$

It should be noted that these are two different expressions for the same matrices. Although these primal-dual expressions may initially appear redundant, they play a crucial role in explaining primal attention, a more computationally optimal way of describing self-attention. See Section 2-5 for how the E_1 and E_2 matrices are used to describe primal attention.

2-3 Primal-dual (K)MLSVD

MLSVD is the multidimensional variant of the SVD. Where SVD is applied on matrices, MLSVD can be applied on any higher order tensor. MLSVD decomposes an N-dimensional tensor $\mathbf{X} \in \mathbb{R}^{N_1 \times \cdots \times N_N}$ into a core tensor $\mathbf{S} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ and N semi-orthogonal factor matrices $U_n \in \mathbb{R}^{N_n \times R_n}$ which admit the following decomposition

$$\begin{aligned} \mathcal{X} &= U_1 \times_1 \mathcal{S} \times_2 ... \times_N U_N, \\ \text{MLSVD}(\mathcal{X}) &= [U_1, U_2, ..., U_N, \mathcal{S}] \end{aligned} \tag{2-16}$$

where R_n denotes the rank of the mode-n-unfolding of the tensor $\mathcal{X}$. $\mathcal{S}$ has the following property

$$S_{(n)} S_{(n)}^\top := \text{positive diagonal matrix}, \quad \text{for all } n \in \{1, 2, \ldots, N\} \tag{2-17}$$

In this thesis, only the three-dimensional version of primal-dual KMLSVD is used. As such, MLSVD will be referred to in a three-dimensional setting in this thesis even though primal-dual KMLSVD is not limited to 3D.
The 3D MLSVD can be defined with the *Generalized Lanczos' Decomposition Theorem* as introduced by Wesel and Batselier[3]:

Theorem 2.2 (Generalized Lanczos decomposition theorem)**.** *An arbitrary rank-*(R_1, R_2, R_3) *tensor* $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ *can be written in MLSVD form, i.e., as in Equation*(2-16) *with core tensor* $\mathcal{S} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ *and semi-orthogonal factor matrices* $U_1 \in \mathbb{R}^{N_1 \times R_1}$, $U_2 \in \mathbb{R}^{N_2 \times R_2}$, *and*

$U_3 \in \mathbb{R}^{N_3 \times R_3}$ *defined by the following set of equations:*

$$\begin{aligned} U_1 S_{(1)} &= X_{(1)}(U_3 \otimes U_2), \\ U_2 S_{(2)} &= X_{(2)}(U_3 \otimes U_1), \\ U_3 S_{(3)} &= X_{(3)}(U_2 \otimes U_1), \end{aligned} \tag{2-18}$$

with the additional constraint that $S_{(1)}S_{(1)}^\top$, $S_{(2)}S_{(2)}^\top$, *and* $S_{(3)}S_{(3)}^\top$ *are positive diagonal matrices.*

MLSVD for the 3D case can be recast into a primal-dual formulation and non-linearized just like primal dual KSVD. The primal dual formulation of MLSVD on a Kernel tensor $\mathcal{K} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ (KMLSVD) is defined as follows. Let

- $W_1 \in \mathbb{R}^{N_1 \times R_1}$, $W_2 \in \mathbb{R}^{N_2 \times R_2}$ and $W_3 \in \mathbb{R}^{N_3 \times R_3}$ be learnable weights matrices.
- $E_1 \in \mathbb{R}^{N_1 \times R_1}$, $E_2 \in \mathbb{R}^{N_2 \times R_2}$, $E_3 \in \mathbb{R}^{N_3 \times R_3}$ denote equality condition matrices.
- $\Phi_1 \in \mathbb{R}^{N_1 \times M_1}$, $\Phi_2 \in \mathbb{R}^{N_2 \times M_2}$, $\Phi_3 \in \mathbb{R}^{N_3 \times M_3}$ be given or pre-computed feature matrices.
- tensors $\mathcal{C} \in \mathbb{R}^{M_1 \times M_2 \times M_3}$ and $\mathcal{S} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ be given.

Primal (2-19)

$$\begin{aligned} \max_{W_1,W_2,W_3,E_1,E_2,E_3} \quad & \mathcal{J}(W_1, W_2, W_3, E_1, E_2, E_3) := \\ & \frac{1}{2}\sum_{d=1}^{3} \operatorname{Tr}\left(E_d (S_{(d)} S_{(d)}^\top)^{-1} E_d^\top\right) \\ & - 2\,\mathrm{vec}(\mathcal{C})^\top (W_3 \otimes W_2 \otimes W_1)\mathrm{vec}(\mathcal{S}) \\ & + \frac{1}{2}\mathrm{vec}(\mathcal{C})^\top \left(\Phi_3^\top \Phi_3 \otimes \Phi_2^\top \Phi_2 \otimes \Phi_1^\top \Phi_1\right) \mathrm{vec}(\mathcal{C}) \\ \text{subject to:} \quad & E_1 = \Phi_1 C_{(1)} (W_3 \otimes W_2) S_{(1)}^\top, \\ & E_2 = \Phi_2 C_{(2)} (W_3 \otimes W_1) S_{(2)}^\top, \\ & E_3 = \Phi_3 C_{(3)} (W_2 \otimes W_1) S_{(3)}^\top. \end{aligned}$$

where

- M_1, M_2, M_3 denote the dimensionality of the first, second and third feature map Φ_1, Φ_2 and Φ_3 respectively.
- N_1, N_2, N_3 denote the size of the first, second and third mode respectively of the kernel tensor $\mathcal{K}$
- R_1, R_2, R_3 denote the rank of the mode-1, mode-2 and mode-3 unfolding respectively of the kernel tensor $\mathcal{K}$

The E_n equality conditions of the primal KMLSVD serve the same function as that of the primal KSVD, maximizing the captured variance.
The dual expression is as follows, where $K_{(n)}$ and $S_{(n)}$ denotes the mode-n-unfolding of the $\mathcal{K}$ tensor and the $\mathcal{S}$ tensor respectively

$$\begin{aligned} &\textbf{Dual}: \\ &U_1 S_{(1)} = K_{(1)}(U_3 \otimes U_2), \\ &U_2 S_{(2)} = K_{(2)}(U_3 \otimes U_1), \\ &U_3 S_{(3)} = K_{(3)}(U_2 \otimes U_1), \end{aligned} \tag{2-20}$$

With $U_1 \in \mathbb{R}^{N_1 \times R_1}$, $U_1 \in \mathbb{R}^{N_1 \times R_1}$ and $U_3 \in \mathbb{R}^{N_3 \times R_3}$. For an exact reconstruction of the dual(2-20) from the primal(2-19), see the appendix.
Linear KMLSVD (or MLSVD) can be attained by setting $\Phi_1 = X_{(1)}$, $\Phi_2 = X_{(2)}$ and $\Phi_3 = X_{(3)}$ in the **primal** and $\mathcal{K} = \mathcal{X}$ in the **dual**. This is, however, not useful seeing as $M_n \gg N_n$ will always hold and as such, solving the dual is always the less computationally complex solution.
The dual(2-20) exactly corresponds to theorem 2.2 if the following relations hold:

1. Equation 2-21 holds.
2. $\mathcal{S}$ has the same properties as a MLSVD-$\mathcal{S}$ tensor (see theorem 2.2).

If the above relations hold, the dual(2-20) is a definition of the MLSVD on a Kernel tensor $K \in \mathbb{R}^{N_1 \times N_2 \times N_3}$. This in turn means that the primal is also a definition of MLSVD on a Kernel tensor. Additionally, the above two relations holding also ensure that U_1, U_2 and U_3 are indeed orthogonal.
Much like with primal-dual KSVD, the $\mathcal{C}$ tensor ensures that the Kernel tensor $\mathcal{K}$ can be constructed from the three feature maps Φ_1, Φ_2, Φ_3 even when those feature maps have different dimensionalities. A different way of seeing the $\mathcal{C}$ tensor is as a "mapping" of how the three feature-maps should be combined. The C tensor should be designed in such a way that the following holds:

$$K = \Phi_1 \times_1 C \times_2 \Phi_2 \times_3 \Phi_3 \tag{2-21}$$

or equivalently

$$\text{vec}(K) = (\Phi_3 \otimes \Phi_2 \otimes \Phi_1)\text{vec}(C). \tag{2-22}$$

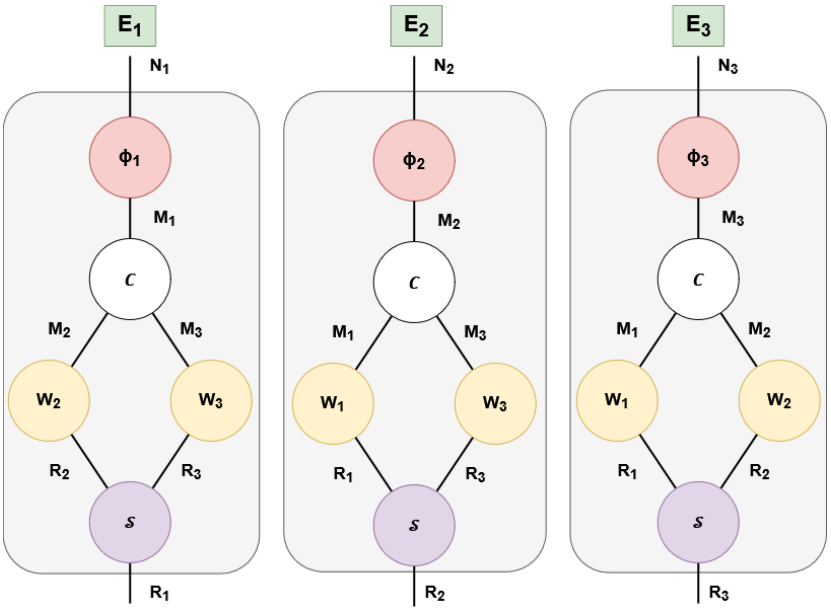
The question of why even using the primal over the dual might naturally arise. According to Batselier et al.[3], if $N \ll M$ holds, the dual is computationally more favorable and if $N \gg M$ holds, the primal is more computationall favorable where $N := \text{Max}(N_1, N_2, N_3)$ and $N := \text{Max}(M_1, M_2, M_3)$. This allows for some flexibility in solving the KMLSVD. This decrease in computational complexity of the primal if $M \ll N$ is also seen back from tests, where using the dual KMLSVD formulation lead to significantly longer computation times (see sections 4-2-1 and 4-2-2).
The primal formulation of KMLSVD suffers from the same "chicken and egg" problem as primal KSVD. The core $\mathcal{S}$ tensor of KMLSVD is assumed to be known in the primal formulation. However, the core $\mathcal{S}$ tensor can only be known if the KMLSVD is solved in the dual (i.e., the "canonical" way of solving the KMLSVD). This problem was addressed for primal KSVD by Chen et al.[4] by setting the core S matrix as an optimization variable. Whether

setting $\mathcal{S}$ to a learnable parameter tensor also works for the primal KMLSVD formulation has not been researched in literature and is tested in this thesis.
The E_1, E_2 and E_3 matrices can now also be expressed in primal or dual formulation.

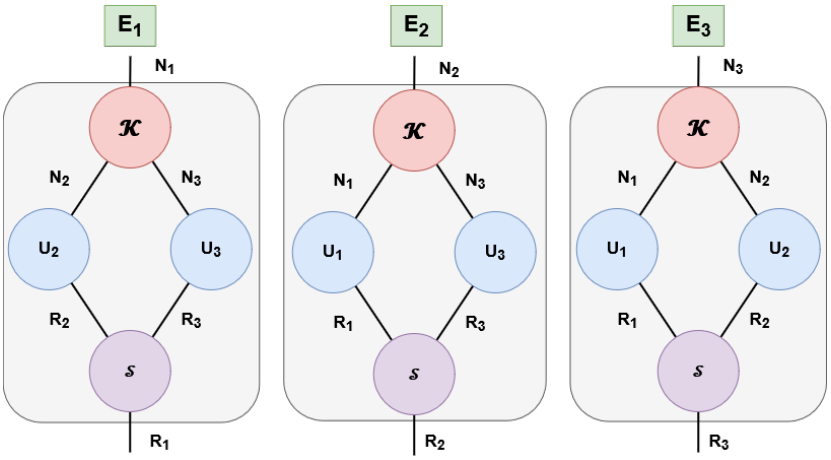
primal:	**dual:**
$E_1 = \Phi_1 C_{(1)} (W_3 \otimes W_2) S_{(1)}^\top$	$E_1 = K_{(1)} (U_3 \otimes U_2) S_{(1)}^\top$
$E_2 = \Phi_2 C_{(2)} (W_3 \otimes W_1) S_{(2)}^\top$	$E_2 = K_{(2)} (U_3 \otimes U_1) S_{(2)}^\top$
$E_3 = \Phi_3 C_{(3)} (W_2 \otimes W_1) S_{(3)}^\top$	$E_3 = K_{(3)} (U_2 \otimes U_1) S_{(3)}^\top$

(2-23)

The corresponding tensor diagrams are given in figure 2-4



(a) Primal tensor network representation of E_n matrices



(b) Dual tensor network representation of E_n matrices

Figure 2-4: E_n matrices represented in tensor networks for primal (a) and dual (b) formulation

The primal and dual expressions of these matrices are crucial for defining self attention in 3D. See chapter 3-1 for an in-depth overview of how these matrices are used to define 3D self attention.

2-4 Self-attention

In the following section, quite a lot of symbols for different dimensionalities are used. A simple table with an overview of the dimensionalities and their corresponding symbols are given below for the sake of readability.

Symbol	Description
N	Sequence length of the input
M	Embedding dimension of the input
d_q	Embedding dimension of the queries
d_k	Embedding dimension of the keys
d_v	Embedding dimension of the values
R	Low-rank approximation of input embedding M
M_n	Embedding dimension of the n-th feature-map Φ_n

Table 2-1: Overview of dimensionalities used in this section

Self-attention is a relatively new mechanism in machine learning, used in various applications—including large language models such as ChatGPT[26]. It is designed to capture dependencies and relationships within sequences of input data. In the context of large language models like ChatGPT, for example, self-attention captures the relationships between words within an input segment, allowing the model to understand each word in the context of the others. In this section, the self-attention framework, as defined by Vaswani et al.[24] is explained. For the rest of the thesis, this self-attention framework will be referred to as "canonical self-attention".
Consider a data sequence $X \in \mathbb{R}^{N\times M}$ where N denotes the sequence length and M represents the dimensionality of each input vector (i.e., the data has N number of input vectors each of length M). With this data sequence, self attention first computes *queries*: $Q(X) \in \mathbb{R}^{N\times d_q}$, *keys*: $K(X) \in \mathbb{R}^{N\times d_k}$ and *values*: $V(X) \in \mathbb{R}^{N\times d_v}$

$$Q(X) = X\hat{W}_q^\top \quad K(X) = X\hat{W}_k^\top \quad V(X) = X\hat{W}_v^\top \tag{2-24}$$

with $\hat{W}_q \in \mathbb{R}^{d_q\times M}$, $\hat{W}_k \in \mathbb{R}^{d_k\times M}$ and $\hat{W}_v \in \mathbb{R}^{d_v\times M}$. The d_q, d_k and d_v are the dimensionality of the queries, keys and values respectively. These can differ from the dimensionality of the dataset M, but within the context of this thesis the following relation will hold:

$$d_q = d_k = d_v = M. \tag{2-25}$$

After computing the queries, keys and values(2-24), self attention computes the so-called attention score matrix $A \in \mathbb{R}^{N\times N}$(2-26) [4]

$$A = \frac{Q(X)K(X)^\top}{\sqrt{M}}. \tag{2-26}$$

where each entry of A is a dot product between a vector of the queries $Q(X)$ and a vector of the keys $K(X)$. Additionally, the attention score is divided by $\sqrt{M}$ (i.e., the dimensionality of the queries and keys) as a normalization step. This is done to prevent entries of the attention score matrix to become overly large, which in turn stabilizes the learning process[24].
This attention score is then non-linearized with an activation function, usually with the *softmax* activation function[24].

$$f_{\text{softmax}}(A)_i = \frac{e^{y_i}}{\sum_{j=1}^{\top} e^{y_j}}. \tag{2-27}$$

Applying the softmax to A has several effects. First, it makes the values in each row lie closer in value to each other, further stabilizing the learning process. Second, applying softmax ensures that all entries are now positive, avoiding the issue of negative values and their detrimental effect on the learning process. Finally, the softmax causes each row of the matrix to sum to 1. This, in turn, means that the resulting matrix can be interpreted as a probability distribution over keys for each query. In this sense, the model assigns weight, or a degree of attention, to every key entry based on how relevant it is to the current query entry.
The softmax is not the only activation function that can be used[16] so a generic activation function f_{active} will be used for this section. The output of self-attention $Y_{\text{att}} \in \mathbb{R}^{N \times d_v}$ is computed by passing the attention score A through an activation function f_{active} and multiplying it with the values $V(X)$.

$$Y_{\text{att}} = f_{\text{active}}\left(\frac{Q(X)^\top K(X)}{\sqrt{d_k}}\right) V(X). \tag{2-28}$$

The underlying logic of the self-attention output(2-28) is that it uses the non-linearized attention score(2-27) to encode context into the output Y_{att}. The non-linearized attention scores are used to weight and aggregate the values $V(X)$, the values containing the actual content or features of the input sequence. This way, the final output of self-attention reflects both the contextual structure of the sequence and the underlying data content. The self-attention output can basically be viewed as a context weighed and influenced version of the input X.

Transformers

In order to properly use the self-attention mechanism, a Transformer is necessary[24]. The self-attention mechanism on its own is not complete enough to establish a full input-output relation. A Transformer is essentially that: a deep learning model built around self-attention mechanisms, extended with additional layers such as feed-forward networks and normalization for example[4], to generate a full and coherent output from the input data. Transformers for example add position encoding[24] which is vital for learning. Transformers can be seen as the entire package, it generates predictions (or other types of outputs based on what is required) based on the data. Self-attention can be seen as (vital) building stone of transformers, giving them the ability to capture complex dependencies across the entire input sequence and thus form the foundation for their effectiveness.

Multi-headed self-attention

In a lot of modern transformers and in this thesis, multi-headed self-attention is used. Multi-headed self-attention is a key mechanism in transformer models that allow them to focus on different aspects of a sequence simultaneously[24][13]. Instead of using a single attention-mechanism to process the entire input sequence $X \in \mathbb{R}^{N \times M}$, multiple attention mechanisms with a lower dimensional output are applied on the dataset X. The output of these self-attention mechanisms are then combined into a final output. The idea behind multi-headed self-attention is that, by applying multiple lower-dimensional self-attention mechanisms, a richer representation can be learned, enabling the model to capture different types of relationships and contextual cues within the data more effectively.

See figure 2-5 for a visual representation of how the dataset is divided into multiple heads. In this figure, each color of the original dataset corresponds to one element of the input sequence, and the blocks with the same color represent the individual dimensions/variables of that element's representation. Thus, the original dataset can be seen as a sequence of length 4, with each element expressed in 12 variables. Each attention mechanism produces a standard self-attention output (2-28), which preserves the original sequence length but represents each element in a lower-dimensional embedding than the input dataset. Each separate attention mechanism is also referred to as a head.

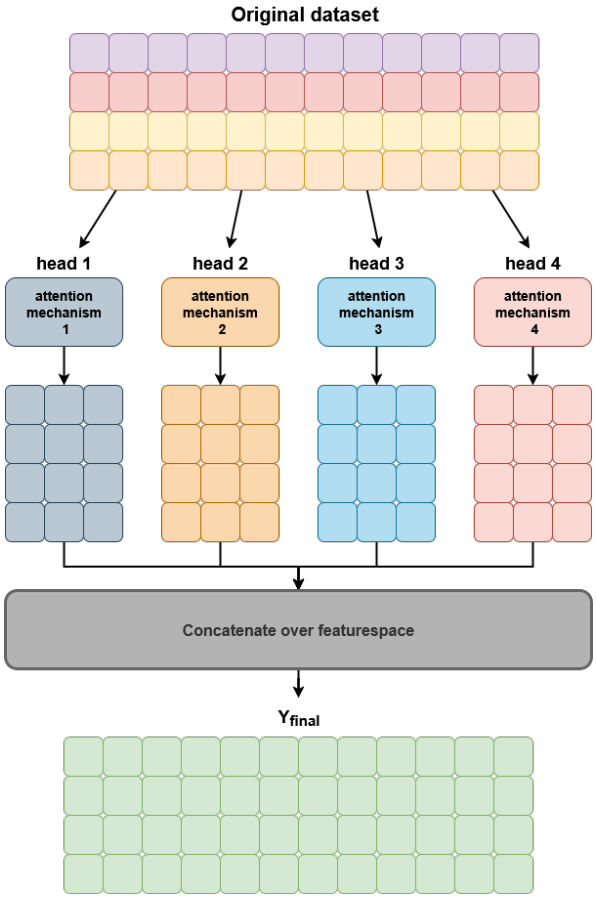


Figure 2-5: Visual representation of how multi-headed attention works. Note that each head processes the entire sequence to a lower dimensional representation. These lower dimensional outputs are then concatenated in a linear layer.

In practice, multi-headed attention works and is applied as follows. Instead of initializing the query, key and value weights as $M \times M$ weight matrices, they are initialized as: $W_q \in \mathbb{R}^{h \times M \times \frac{M}{h}}, W_k \in \mathbb{R}^{h \times M \times \frac{M}{h}}, W_v \in \mathbb{R}^{h \times M \times \frac{M}{h}}$ where h denotes the amount of heads. This can alternatively be viewed as creating h separate $W_q^{(i)} \in \mathbb{R}^{M \times \frac{M}{h}}, W_k^{(i)} \in \mathbb{R}^{M \times \frac{M}{h}}$ and $W_v^{(i)} \in \mathbb{R}^{M \times \frac{M}{h}}$ instances with $i \in [1, .., h]$. Usually (and in within the context of this thesis) the amount of parameters per head is kept equal meaning that each head projects the original dataset dimension of M to $\frac{M}{h}$. With these h separate weight matrices, h separate queries, keys and values are computed $(Q(X)^{(i)} \in \mathbb{R}^{N \times \frac{M}{h}}, K(X)^{(i)} \in \mathbb{R}^{N \times \frac{M}{h}}, V(X)^{(i)} \in \mathbb{R}^{N \times \frac{M}{h}})$.

Those separate queries, keys and values are then used to compute h separate self attention outputs(2-28). These outputs are finally concatenated into one large output. Note that multi-headed attention does not increase or decrease the amount of learnable parameters of the self-attention mechanism itself. The addition of a linear concatenation layer however does increase the total amount of learnable parameters.

2-5 Using primal-dual KSVD to describe self-attention

As stated in the introduction of this chapter, self-attention can be described using the primal-dual KSVD framework. Such an alternative description simplifies the computational complexity and leads to improved performance and accuracy of the model [4].
The non-linearized attention score(2-27) can alternatively be viewed as a non-linear kernel matrix or function[4]. The output of the self attention mechanism can then be re-formulated to

$$\begin{aligned} Y_{\text{att}} &= \hat{K}_{\text{att}}(X)V(X) \\ \text{with } \hat{K}_{\text{att}} &= f_{\text{active}}\left(\frac{Q(X)^\top K(X)}{\sqrt{M}}\right). \end{aligned} \tag{2-29}$$

Note that when the re-written self attention output(2-29) is very similar to the dual KSVD expression of the E_1 matrix(2-15). In fact, the self-attention output is exactly the same as the dual expression for E_1 if the values $V(X)$ are equal to the left singular vectors of the SVD decomposition of $K_{\text{att}}(X)$(2-29)[4]. So if equation 2-30 holds, $E_1 = Y_{\text{att}}$ also holds.

$$V(X) = U_1 \tag{2-30}$$

In other words: self attention can entirely be described by solving the SVD(2.1) on the self-attention kernel matrix(2-29) and computing the E_1 matrix(2-15). Moreover, the computation of the self-attention kernel(2-29) can be circumvented by solving the KSVD in the primal, which leads to a significant decrease in computation time[4]. This can be done because the primal and dual formulation of the E_1 matrix are two different ways to describe the same thing (i.e., E_1).
The authors who originally introduced the concept of primal attention (Chen et al. [4]) use a slightly modified primal KSVD formulation (2-11) to describe self attention with primal KSVD.
Given feature maps $\Phi_1 \in \mathbb{R}^{N\times M_2}$, $\Phi_2 \in \mathbb{R}^{N\times M_2}$, let the following definition be a modification of equation 2-11 and let $E_1 \in \mathbb{R}^{N\times R}$ and $E_2 \in \mathbb{R}^{N\times R}$ additionally serve as outputs for this specific primal KSVD formulation

$$\begin{aligned} & \textbf{primal:} \\ \max_{W_1,W_2,E_1,E_2,S} \quad & J_{\text{KSVD}}(W_1,W_2,E_1,E_2,S) := \\ & \frac{1}{2}\sum_{d=1}^{2}\text{Tr}(E_d\hat{S}^{-1}E_d^\top) - \text{Tr}(W_2^\top C^\top W_1) \\ \text{subject to:} \quad & E_1 = \Phi_1 C W_{2|x}, \\ & E_2 = \Phi_2 C^\top W_{1|x}, \end{aligned} \tag{2-31}$$

with $W_{1|x} \in \mathbb{R}^{M\times R}$, $W_{2|x} \in \mathbb{R}^{M\times R}$, $W_1 \in \mathbb{R}^{\hat{N}\times R}$ and $W_2 \in \mathbb{R}^{\hat{N}\times R}$ where $R \leq \min(N, M)$ denotes the rank of the kernel matrix K and $\hat{N} \leq N$. The main difference between primal KSVD formulation used by Chen et al. [4] and the primal KSVD definition as introduced by Suykens et al. [10] is the substitution of W_1 and W_2 with $W_{1|x}$ and $W_{1|x}$ respectively. These represent *data-dependent* versions of the primal KSVD weight matrices W_1 and W_2 and are computed as follows

$$W_{1|x} = f(X)^\top W_1 \qquad W_{2|x} = f(X)^\top W_2 \tag{2-32}$$

where $f(X) \in \mathbb{R}^{M\times\hat{N}}$ denotes a possible sub-sampling of the original dataset X. This sub-sampling is done in order to still process large sequence datasets. It is not exactly clear why this is done, but the current theory is that using data dependent weights mimics or approximates the logic of the values(2-24) which are computed by multiplying the dataset X with a weight matrix $\hat{W}v$ which alternatively can be viewed as generating a data dependent weight matrix which linearly depends on the dataset X. Since the values can be seen as data dependent weights and are directly used in the output of self attention, similar logic should apply to the output of primal attention. Note that the data dependent weight matrices are not used in the cost function itself. This is due to the fact that the $\mathrm{Tr}(W_2^\top C^\top W_1)$ term is used to regulate the weight matrices to prevent overly large values[4][10].
R represents the rank of the self-attention kernel matrix (see equation (2-29)). However, in the primal formulation of KSVD, the kernel matrix is not directly computed, so the exact rank R cannot directly be determined. Instead, R can be used as a potentially low-rank approximation of the original embedding dimension M. This could lead to a more efficient representation of the primal attention mechanism, decreasing the computation time and requirements (energy, cooling etc.).
The choice of feature maps Φ_1 and Φ_2 cannot be arbitrary and should incorporate both non-linearity and the queries and keys[4]. Chen et al. [4], for instance, propose using cosine similarity as the basis for designing the feature maps:

$$\Phi_1 = \frac{Q(X)}{||Q(x)||_2} \qquad \Phi_2 = \frac{K(X)}{||K(x)||_2} \tag{2-33}$$

Another idea is to try and mimic the softmax self attention kernel(2-29) with the feature maps such that the following relation holds:

$$f_{softmax}\left(\frac{Q(X)^\top K(X)}{\sqrt{(d_k)}}\right) \approx \Phi_1 C \Phi_2^\top. \tag{2-34}$$

This can be done with the positive orthogonal random feature map[16].

$$\Phi_1 = \frac{1}{\sqrt{M}}\exp\left(-\tfrac{\|q\|^2}{2}\right)\begin{bmatrix}\exp\left(w_1^\top q\right)\\ \exp\left(w_2^\top q\right)\\ \vdots\\ \exp\left(w_m^\top q\right)\end{bmatrix}, \quad \Phi_2 = \frac{1}{\sqrt{M}}\exp\left(-\tfrac{\|k\|^2}{2}\right)\begin{bmatrix}\exp\left(w_1^\top k\right)\\ \exp\left(w_2^\top k\right)\\ \vdots\\ \exp\left(w_m^\top k\right)\end{bmatrix}, \tag{2-35}$$

with $m \leq M$, $w \in \mathbb{R}^m$ a vector generated from the standard normal distribution and $q \in \mathbb{R}^M$ and $k \in \mathbb{R}^M$ vector entries of the queries and keys $Q(X)$ and $K(X)$. Another possible

approach is to normalize the queries and keys by their Frobenius norm, as shown below in equation 2-36.

$$\Phi_1 = \frac{Q(X)}{||Q(X)||_F} \qquad \Phi_2 = \frac{K(X)}{||K(X)||_F} \tag{2-36}$$

The motivation for this choice comes from Saratchandran et al. [12], who argue that the strong performance of the self-attention mechanism arises from an implicit regularization of the attention score matrix. However, since the attention scores are not directly computed in the primal formulation, their Frobenius norm cannot be directly regulated. By using these feature maps, the hope is that there is still some implicit regularization of the Frobenius norm.
In section 2-2, the equality conditions E_1 and E_2 were introduced to maximize the variance. In primal attention, they serve as an output that describes the self-attention mechanism. If the notation of Chen et al.[4] is adapted, E_1 and E_2 have the following primal and dual formulations,

primal:	**dual:**
$E_1 = \Phi_1 C W_{2\|x},$	$E_1 = \overbrace{\Phi_1 C \Phi_2^\top}^{\hat{K}_{\text{att}}} U_1 = \hat{K}_{\text{att}} U_1,$
$E_2 = \Phi_2 C^\top W_{1\|x},$	$E_2 = \overbrace{\Phi_2 C^\top \Phi_1^\top}^{\hat{K}_{\text{att}}^\top} U_2 = \hat{K}_{\text{att}}^\top U_2,$

(2-37)

recall that the C-matrix was primarily meant as a way to circumvent a dimensional mismatch when multiplying the feature maps with the weights. Since no dimensional mismatch is present between Φ_1 and $W_{2|x}$ and between Φ_2 and $W_{1|x}$ the C-matrix can be set to a square identity matrix. This in turn means that the C-matrix is not necessary and the E_1,E_2 equations simplify to

primal:	**dual:**
$E_1 = \Phi_1 W_{2\|x},$	$E_1 = \overbrace{\Phi_1 \Phi_2^\top}^{\hat{K}_{\text{att}}} U_1 = \hat{K}_{\text{att}} U_1,$
$E_2 = \Phi_2 W_{1\|x},$	$E_2 = \overbrace{\Phi_2 \Phi_1^\top}^{\hat{K}_{\text{att}}^\top} U_2 = \hat{K}_{\text{att}}^\top U_2,$

(2-38)

if the C-matrix is indeed set to be equal to the identity matrix.
Besides the ability to circumvent the kernel matrix computation, using the primal dual KSVD framework also grants "extra" information or data on the self-attention output in the form of the E_2 matrix which is otherwise not present in standard self-attention[4]. In practice, this extra output E_2 is concatenated with E_1 and then passed through a linear, fully connected layer, generating a more information rich output. So not only does primal self attention lead to a faster self-attention mechanism, it also grants extra information and possibly better accuracy.
The primal formulation is computationally more favorable than the dual if the sequence length N is larger than the dimensional embedding M. To clarify this, consider the computation of the kernel matrix in standard self-attention. Let $X \in \mathbb{R}^{N \times M}$ denote a dataset where N

is the sequence length and M is the dimensionality of each input vector. In canonical self-attention, the resulting kernel matrix has the following dimensionality $\hat{K}_{\text{att}} \in \mathbb{R}^{N \times N}$. Each of the N^2 entries in the kernel matrix $\hat{K}_{\text{att}}$ are computed as a dot product between two vectors of dimension M. Since the full matrix contains N^2 such entries, the total computational complexity of constructing the kernel matrix $\hat{K}_{\text{att}}$ can blow up. In fact, the computational complexity of the kernel matrix $\hat{K}_{\text{att}}$ is $\mathcal{O}(N^2M)$ and the computational complexity of the self attention output is $\mathcal{O}(N^2M)$[18] bringing the total computational complexity to $\mathcal{O}(2N^2M)$. The computational complexity of the simplified primal KSVD output (E_1) is $\mathcal{O}(2NMR)$. So note that if $M < N$ holds, the primal is more computationally favorable and if $M > N$ holds, the dual is more favorable.
Recall that in primal KSVD suffers from the "chicken and egg" problem of the S-matrix as discussed in section 2-2. This problem is solved by Chen et al. [4] by also setting the S-matrix as a learnable parameter. In order to enforce that the S-matrix does indeed adhere to the positive diagonal conditions, Chen initializes it as a positive diagonal S-matrix where only the diagonal are learnable parameters.

2-6 Implementing Primal Attention

Having discussed the various theoretical components of primal attention, this section details how these parts are used in an implementation.
Before delving into the implementation of the learning process of a neural network should be explained. Neural networks learn through a two-step process: i) estimations and ii) corrections. First, a neural network makes an estimation based on the input data. Then, it evaluates how far off this prediction is using a loss function, which quantifies the error. The network uses this error to adjust its internal parameters via backpropagation [11] to improve future predictions. The loss function plays a crucial role in the network's ability to properly "learn" from the data as it defines what the error means for a given task and will be referred to as the *task related loss* in this paper. A good example of a commonly used task related loss is the Mean Squared Error Loss, which measures how far predictions $\hat{y}_i$ are from the actual targets y_i,

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{2-39}$$

with N in equation 2-39 denoting the total number of samples.
When using the primal KMLSVD formulation to define self attention, the task related loss is not sufficient for a good learning process[4]. Chen et al. propose a modified loss function to solve this problem where L denotes the original task related loss (e.g., Mean Squared Error Loss(2-39)) and J_{KSVD} denotes the primal attention KSVD cost function from equation (2-31)

$$\min \quad J_{total} := L + \eta J_{\text{KSVD}}^2. \tag{2-40}$$

The underlying idea behind this new loss function is twofold. First, the J_{KSVD}^2 term ensures that the overall cost function and optimization problem remains a primal KMLSVD formulation. It is squared to enforce an optimal cost of 0. In effect, the J_{KSVD}^2 term acts as a regularizer that enforces the solution to be a primal KSVD solution. The η is a tunable hyperparameter that determines how strong the J_{KSVD} influences the overall cost function

J_{total}. The task related loss L ensures the model also learns from the data itself. To summarize, J_{KSVD} is a regularization term to enforce KSVD structure and L enforces learning from the data itself.
Finally, applying primal attention on a given dataset $X \in \mathbb{R}^{N \times M}$ (N entries, each entry having M variables) involves three steps: i) initialization, ii) forward step and iii) loss calculation and backpropagation.
Initialization: Before the learning process starts, primal attention initializes the following weight matrices $W_1 \in \mathbb{R}^{N \times M}, W_2 \in \mathbb{R}^{N \times M}, W_q^{M \times M}, W_k^{M \times M}$ and diagonal matrix $S \in \mathbb{R}^{R \times R}$. Here N is the sequence length ,M is the embedding size of the primal attention mechanism and $R \leq M$ should hold.
Forward step: The forward step is divided in 4 parts:

1. The queries $Q(X)$ and keys $K(X)$ are computed using equation 2-24.
2. Then $Q(X)$ and $K(X)$ are used to compute the feature matrices Φ_1 and Φ_2 (e.g., by using equation 2-33).
3. Subsequently, these feature matrices are used to compute the $E_1 \in \mathbb{R}^{N \times R}$ and $E_2 \in \mathbb{R}^{N \times R}$ matrices using equation 2-38.
4. The E_1 and E_2 matrices are then concatenated and used as the output of the forward step along with the weights W_1 and W_2, the S matrix and the feature maps Φ_1 and Φ_2.

W_1, W_2, S, Φ_1 and Φ_2 are explicitly required for calculating part of the loss function, specifically the J_{KSVD} term in equation 2-31; however, they are not part of the predictive output of the model and are only used in computing the loss. The E_1 and E_2 are the actual predictive outputs of the model.
Loss calculation and backpropagation: The entire output from the previous step is used to compute the loss (2-40) that is subsequently used in backpropagation to update the internal variables (i.e., W_1, W_2, W_q, W_k, S). Note however that the cost function does not calculate any variables apart from the loss itself.

Chapter 3

Primal KMLSVD self-attention

In this chapter, the central contribution of this thesis is introduced: a 3D self-attention formulation based on the primal-dual Kernel Multi-linear Singular Value Decomposition (KMLSVD) framework. Unlike existing (re)formulations of self-attention that either manipulate 3D data to better fit the standard self-attention mechanism[21] or still inherently work in 2D [4][12], the proposed mechanism operates directly in 3D, making the 3D structure a fundamental part of the formulation itself. In addition to this formulation, several alternative variants based on the primal KMLSVD attention are proposed that aim to improve training efficiency and model accuracy. In section 3-1 a general overview of the Primal-Dual KMLSVD framework that describes 3D self-attention is introduced and described. Section 3-2 is a more in-depth analysis of the different modifications of primal KMLSVD attention. In section 3-3 an overview of how to actually implement primal and dual KMLSVD is given. Finally, section 3-4 describes how the Primal-Dual KMLSVD attention framework is used to describe multi-headed self-attention.
Due to the different self-attention mechanisms used, a table is given below 3-1 showing the 3 main self-attention mechanisms used, and the names used to refer to them in this thesis.

Self-Attention Type	**Name in thesis**
Self attention as defined by Vaswani et al. [24]	Canonical self-attention
Self attention described using the Primal KSVD framework as defined by Chen et al. [4]	Primal Attention
Self-attention described using the Primal-Dual KMLSVD framework (the topic of this thesis)	Primal-Dual KMLSVD attention

Table 3-1: Types of self-attention mechanisms and their explanations.

Additionally, a lot of matrices and tensors are defined and used throughout this chapter. For an easier reading experience, a table of these matrices and tensors is given.

matrix or tensor	Description	dimensionality
W_q	Weight matrix from query projection	$\mathbb{R}^{M\times M}$
W_k	Weight matrix from key projection	$\mathbb{R}^{M\times M}$
W_f	Weight matrix from fiber projection	$\mathbb{R}^{M\times M}$
W_1, W_2, W_3	Weight matrices from the primal KMLSVD formulation	$\mathbb{R}^{M\times R}$
$W_{(1\|x)}, W_{(2\|x)}, W_{(3\|x)}$	Data-dependent versions of the primal KMLSVD weight matrices W_1, W_2, W_3	$\mathbb{R}^{M\times R}$
$\mathcal{A}$	3D attention score tensor used in Dual KMLSVD attention	$\mathbb{R}^{N\times N\times N}$
$\mathcal{K}$	3D attention kernel tensor used in Dual KMLSVD attention	$\mathbb{R}^{N\times N\times N}$
Φ_1, Φ_2, Φ_3	Feature maps used in Primal KMLSVD attention	$\mathbb{R}^{N\times M}$
$Q(X), K(X), F(X)$	Queries, Keys, and Fibers used to compute the feature maps Φ_1, Φ_2, Φ_3 respectively	$\mathbb{R}^{N\times M}$

Table 3-2: Overview of the various tensors, their descriptions and dimensionalities used in this chapter. Note that the (data-dependent) KMLSVD weight matrices all share the same shape. While this is not necessary in the primal KMLSVD formulation, it is a deliberate design choice. Additionally, the feature maps are also the same shape due to the same deliberate design choice.

3-1 3D-KMLSVD self-attention

Up until now it has been shown that i) Kernel Singular Value Decomposition (KSVD) can be rewritten into a primal dual 2-2 framework, ii) this primal dual KSVD formulation can be generalized to the 3D (Kernel)Multi-linear Singular Value Decomposition (MLSVD) case 2-3, and iii) how the primal dual KSVD formulation could be used to describe the self-attention mechanism by applying the Singular Value Decomposition (SVD) on the (non-linearized) attention score Kernel matrix and computing E_1(2-28).
Describing the self-attention output with the primal-dual KSVD framework could also be generalized to the 3D primal-dual KMLSVD framework, after all the MLSVD is just a multidimensional version of the SVD. The practical implication of this are i) establishing a self-attention framework that works in three dimensions, and ii) this framework could significantly reduce time and computational complexity by using the primal KMLSVD formulation. This 3D framework could offer a more information-rich self-attention mechanism that could improve the maximum possible accuracy attainable or improve the efficiency of the learning process (i.e., less learning steps needed to get similar results to canonical self-attention). Additionally, on three-dimensional datasets or datasets with inherent three-dimensional relationships, KMLSVD attention could improve the accuracy compared to canonical self-attention. However, before these claims can be investigated, Primal-Dual KMLSVD attention first needs to be proven to work in practice. This thesis will verify the validity of using primal KMLSVD attention for describing self-attention and if the Primal-Dual KMLSVD attention framework can be used as an improved version of canonical self-attention. The next section explains the concept and implementation of the actual KMLSVD attention.

3-1-1 3D-KMLSVD self-attention in the Dual

A straightforward way of understanding 3D self-attention is by first viewing it from the Dual KMLSVD formulation perspective. The goal is to mimic 2D self-attention as closely as possible. To do this, a 3D attention score tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$ is made from a dataset $X \in \mathbb{R}^{N \times M}$ with N entries, each of those entries having an embedding of M variables. Then, the KMLSVD will be applied on the non-linearized 3D attention score tensor and the E_1, E_2 and E_3 matrices are computed in the dual(2-23) using the U_1, U_2 and U_3 matrices attained from the MLSVD.
The 3D attention score tensor is created as follows. First, the queries $Q(X) \in \mathbb{R}^{N \times d_q}$, keys $K(X) \in \mathbb{R}^{N \times d_k}$ and fibers $F(X) \in \mathbb{R}^{N \times d_f}$ are computed

$$Q(X) = XW_q^\top \quad K(X) = XW_k^\top \quad F(X) = XW_f^\top \tag{3-1}$$

where $W_q \in \mathbb{R}^{d_q \times M}$, $W_k \in \mathbb{R}^{d_k \times M}$ and $W_f \in \mathbb{R}^{d_f \times M}$ are learnable weight matrices. In the context of this paper, the following relation is set

$$d_q = d_k = d_f = M. \tag{3-2}$$

These queries, keys, and fibers are then used to compute the 3D attention score $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$ as follows, with $\mathcal{C} \in \mathbb{R}^{M \times M \times M}$ being an identity tensor

$$\mathcal{A} = Q(X) \times_1 \mathcal{C} \times_2 K(X) \times_3 F(X) \tag{3-3}$$

with corresponding tensor diagram notation in figure 3-1. The red square in figure 3-1 is the attention score tensor $\mathcal{A}$. Note that $\mathcal{C}$ is not limited to an identity tensor and, in theory, can be any real valued tensor. However, to keep things as simple and straightforward as possible, it has been set as an identity tensor. This choice also has a conceptual motivation: when $\mathcal{C}$ is an identity tensor, each entry of the 3D attention score $\mathcal{A}$ corresponds to a dot product of a singular query, key, and fiber vector. This mirrors the logic of the standard (2D) self-attention score matrix, where each entry is computed as a dot product between a query and key vector. Recall that in the 2D case, the attention score matrix could be seen as a collection of data point comparisons (each entry of A being a comparison of two separate data points). With the $\mathcal{C}$ tensor being an identity tensor, now each entry of the attention score tensor $\mathcal{A}$ is a comparison of three separate data points.

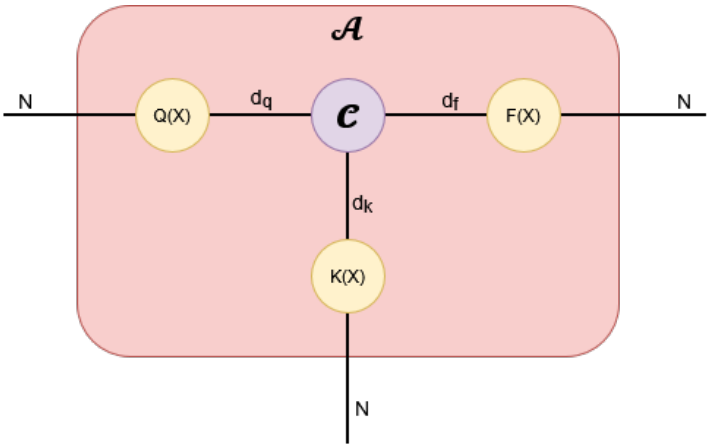


Figure 3-1: Tensor diagram notation of 3D Attention score $\mathcal{A} \in \mathbb{R}^{N\times N\times N}$. In this figure d_q,d_k and d_f are not set to be M for a more complete notation however, within this paper $d_q = d_q = d_f = M$ is always assumed.

The main idea is to replicate the 2D attention score matrix with this 3D attention score tensor. Each entry of the 3D attention score tensor $\mathcal{A}$ is a "3D dot product" of a $Q(X), K(X)$ and $F(X)$ vector.

$$\mathcal{K}_{ijk} = \sum_{m=1}^{M} q(m)^i k(m)^j f(m)^k \tag{3-4}$$

with $q^i \in \mathbb{R}^M, k^j \in \mathbb{R}^M$ and $f^k \in \mathbb{R}^M$ vectors from the $Q(X), K(X)$ and $F(X)$ (3-1) matrices, respectively. The superscripts i, j, k denote which entry of the queries, keys, and fibers matrix is taken and m denotes the m'th entry of the vector itself. This is comparable to standard 2D self-attention, where each entry in the attention score matrix(2-26) represents a dot product between a vector of $Q(X)$ and $K(X)$.
Just as with standard self-attention, the 3D attention score $\mathcal{A}$ is non-linearized. This non-linearization can be done with any non-linear activation function (as with canonical self-attention). The added difficulty is now defining a 3D activation function. An example of this is a modified version of the softmax function(2-27) called the "cubemax". This is a generalization of the softmax to higher dimensions, as defined below:

$$\begin{aligned} &\text{Given: } \mathcal{Z} \in \mathbb{R}^{U\times V\times W}, \text{ let } Z = \mathcal{Z}_i \in \mathbb{R}^{V\times W} \\ &f_{\text{cubemax}}(Z_{i,j}) = \frac{e^{Z_{i,j}}}{\sum_{u=1}^{U}\sum_{w=1}^{W} e^{Z_{w,u}}} \end{aligned} \tag{3-5}$$

Here $f_{cubemax}(Z_{i,j})$ is taken over the first and third modes (i.e., along the U and V axes) of the $\mathcal{Z}$ tensor. In other words, cubemax performs a softmax operation across a 2D slice of the 3D tensor, similar to how softmax is computed along a single axis in the 2D case. In figure 3-2 a visual representation is given on how the cubemax is applied.

The 3D self-attention Kernel $\mathcal{K} \in \mathbb{R}^{N\times N\times N}$ is computed with a 3D activation function $f_{3D,active}$ like the aforementioned cubemax function (3-5).

$$\mathcal{K} = f_{3D,active}\left(\frac{Q(X) \times_1 \mathcal{C} \times_2 K(X) \times_3 F(X)}{M}\right). \tag{3-6}$$

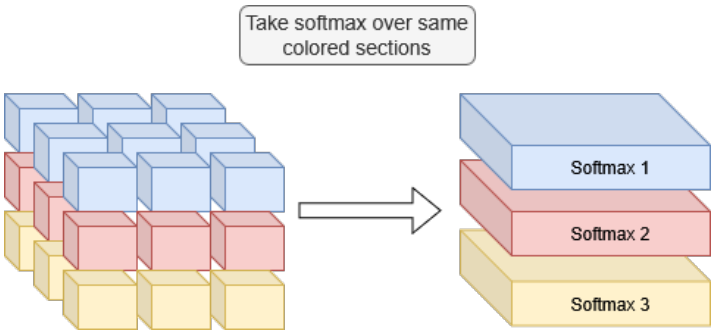


Figure 3-2: Graphical notation on how the Cubemax is applied on a simple $\mathbb{R}^{3\times3\times3}$ tensor. Each block in the left image signifies an entry in the tensor. Note that the softmax is applied on each y-z "slice".

In this case, M is used (as opposed to the square root of M in equation 2-26) to normalize the entries of the attention score tensor. This is done because the attention score tensor has an additional dimension when compared to the classical attention score matrix (2-26). This means that each entry in the attention score tensor can become significantly larger due to it being a result of a dot product of three vectors.
Finally, the MLSVD of $\mathcal{K}$ is computed in the dual(2-20) and the E_1, E_2 and E_3 scores are computed with their dual formulation(see figure 2-4 and equation (2-23)) and concatenated.

$$Y_{out} = \text{concat}\left(E_1, E_2, E_3\right) \tag{3-7}$$

Note that 3D self-attention using the Dual KMLSVD framework will be referred to as Dual KMLSVD attention from this section onward. Primal-Dual KMLSVD attention is self-attention defined using either the Primal or the Dual KMLSVD formulation. See section 3-1-2 for an explanation on what Primal KMLSVD attention is.

3-1-2 3D KMLSVD self-attention in the Primal

As one might expect, if computing the self-attention kernel is already computationally intensive in the two-dimensional case, extending it to three dimensions can be even more demanding. Let alone computing the MLSVD of said tensor. The computational complexity of just computing the 3D KMLSVD kernel tensor is, $\mathcal{O}(N^3M^2)$ which already has a quadratic dependence on the feature space **and** a cubic dependence on the input sequence.
This is precisely where the primal KMLSVD formulation(2-19) becomes valuable. Unlike the dual KMLSVD formulation, the primal KMLSVD formulation circumvents the computation of the kernel tensor $\mathcal{K}$ altogether, offering a potentially more tractable alternative for practical implementations.
Several different modifications of the standard primal KMLSVD attention are further discussed in section 3-2. The most straightforward version, referred to here as the "standard Primal KMLSVD attention", is directly derived from the primal KMLSVD formulation(2-19). In this version, the E_1, E_2 and E_3(2-23) matrices additionally serve as the output for the standard Primal KMLSVD attention mechanism. Furthermore, the $\mathcal{S}$ tensor is treated as a learnable parameter in order to solve the "chicken and egg" problem described in section 2-3. It is important to note that *all* Primal KMLSVD attention modifications discussed in

section 3-2 apply some modification on the standard Primal KMLSVD attention framework introduced in this section.
Note that in this formulation, there is no method to enforce the property of $S_{(n)}S_{(n)}^T$ being a positive diagonal matrix. Strangely enough, testing has found that this formulation is sufficient in describing self-attention even though the condition of $S_{(n)}S_{(n)}^T$ being positive diagonal is not explicitly enforced. However, enforcing the positive diagonal condition (see section 3-3 on two possible enforcement strategies) did increase the accuracy on some datasets during tests.
The output of Primal KMLSVD attention is the same as that of the Dual KMLSVD attention: concatenating the E_1, E_2 and E_3 matrices(3-7). The only difference being how the E_n matrices are computed (i.e., is the primal or the dual formulation used to compute the E_n matrices in equation 2-23).
As for the choice of the feature maps Φ_1, Φ_2 and Φ_3, three possibilities have been considered for this thesis. The i) cosine similarity feature map (2-33), ii) Frobenius norm feature map (3-8) and the iii) softmax approximation feature map (2-35). The Cosine feature-map was chosen due to it being used in primal-attention tests[4]. The Frobenius norm feature map was chosen due to Saratchandran et al. suggesting that regulation of the Frobenius norm is what makes self-attention perform so good[12]. The softmax approximation feature map was chosen due to that feature-map being the closest feature map (as of writing this thesis) that describes the cubemax(3-5) function. It should be noted that the softmax approximation feature map, when applied to Primal KMLSVD attention, does not accurately represent the cubemax function/kernel. Nevertheless, it was still considered for the tests.

$$\frac{Q(X)}{||Q(X)||_F} \quad \frac{K(X)}{||K(X)||_F} \quad \frac{F(X)}{||F(X)||_F} \tag{3-8}$$

Note that 3D self-attention using the Primal KMLSVD framework will be referred to as Primal KMLSVD attention from this section onward.

3-2 Variations of the Primal KMLSVD attention framework

In the previous sections, the basic primal KMLSVD self-attention was defined and explained. In this section, some possible modifications and why those modifications could improve the performance are discussed.
The two main modifications looked at in this paper are i) making the Primal KMLSVD weights W_1, W_2 and W_3 data dependent just like with primal attention (see section 2-5) and ii) making the $\mathcal{C}$-tensor a learnable parameter instead of a pre-defined identity tensor.
Data dependent weight matrices: First, making the weights data dependent is not as straightforward as one might expect. As mentioned in section 2-5 the weights W_n are multiplied by a linear transform $f(X)$ of the dataset (see equation 2-32) in the 2D case. Translating this to the 3D case would mean that the outputs and equality conditions E_1, E_2, E_3(2-23) of primal KMLSVD attention would have the following equation

$$\begin{aligned} E_1 &= \Phi_1 C_{(1)}(W_{3|x} \otimes W_{2|x})S_{(1)}^\top, \\ E_2 &= \Phi_2 C_{(2)}(W_{3|x} \otimes W_{1|x})S_{(2)}^\top, \\ E_3 &= \Phi_3 C_{(3)}(W_{2|x} \otimes W_{1|x})S_{(3)}^\top, \end{aligned} \tag{3-9}$$

with

$$W_{1|x} = f(x)^T W_1 \quad W_{2|x} = f(x)^T W_2 \quad W_{3|x} = f(x)^T W_3. \tag{3-10}$$

The primal KMLSVD attention formulation, as defined in section 3-1-2, stays otherwise unchanged.

The non-data-dependent versions of the weight matrices are used in the cost function because that term serves as a regularization, ensuring the values do not grow too large. However, initial testing has proven that this setup causes the primal KMLSVD self-attention mechanism to break down. It was unable to learn from datasets and never exceeded its initial accuracy during tests. It is not exactly clear why learning breaks down with this set up, but the current theory is that, due to the presence of a Kronecker product between two data dependent weight matrices, there is a quadratic dependency on the dataset X. This quadratic dependency breaks down the learning process. In order to test this, a new formulation of this data dependent weight matrices was made which linearly incorporates the dataset X. This new formulation comes down to replacing the $\mathcal{C}$ tensor with $\mathcal{X}_T \in \mathbb{R}^{N\times N\times M}$ which is a tensorized version of the dataset $X \in \mathbb{R}^{N\times M}$ where X lies on the diagonal of $\mathcal{X}_T$ such that the following relations hold

$$\mathcal{X}_T(n,:,n) = X(n,:) \qquad \text{for } \forall n \in \{1,\cdots,N\}, \tag{3-11}$$

$$\mathcal{X}_T(l,:) = 0 \qquad \text{for } \forall l \neq n. \tag{3-12}$$

See figure 3-3 for the tensor diagram of the E_1, E_2 and E_3 matrix outputs.

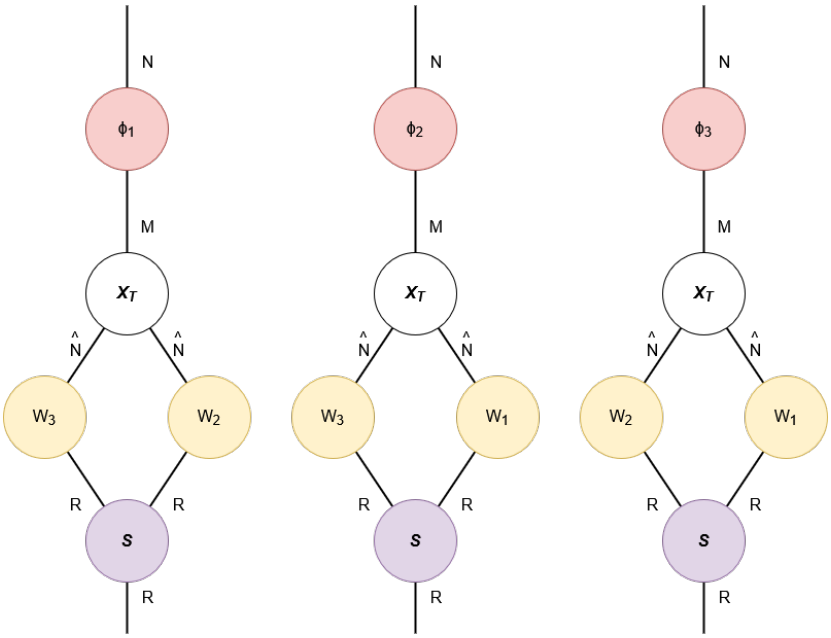


Figure 3-3: Tensor diagram notation of E_1, E_2 and E_3 for the data dependent primal KMLSVD attention setup. In comparison to figure 2-4a, the $\mathcal{C}$ tensor and its unfoldings have been replaced by the mode-3-unfolding of $\mathcal{X}_\mathcal{T}, X_{(3)}$

Note that for each E_1, E_2 and E_3 score, the mode-3-unfolding of $\mathcal{X}_T$ is used. This is done by design due in order to prevent a dimensional mismatch.

learnable $\mathcal{C}$ tensor: Second the $\mathcal{C}$-tensor can be set as a learnable parameter instead of initialized as an identity tensor. In this thesis, two main ways of making the $\mathcal{C}$-tensor learnable are implemented: i) initializing the entire $\mathcal{C}$-tensor as a learnable parameter or ii) only initializing the hyper diagonal of the $\mathcal{C}$-tensor as a learnable parameter. The reasoning behind option i is that a more optimal way of combining and comparing the queries, keys and fibers might be learned by setting the $\mathcal{C}$ tensor as a learnable parameter.
learnable vectorized $\mathcal{C}$ tensor: Due to the potential large size of the $\mathcal{C}$ tensor, setting the entire tensor as a learnable parameter could significantly slow down the learning process. In order to prevent this, only the diagonal of the $\mathcal{C}$ tensor could be set to be learnable parameters. This way, there are only M additional learnable parameters as opposed to M^3.

As both methods (i.e., data dependent weight matrices and learnable $\mathcal{C}$-tensor) change how the $\mathcal{C}$-tensor is defined, both methods cannot be used at the same time. That is why in the rest of this paper, these two methods will never be used at the same time. This means that there are, in total, 4 separate primal attention methods

Primal KMLSVD attention type	**Reference symbol or name**
Standard, as defined in section 3-1-2	standard
data dependent	$W_{n\|x}$
learnable $\mathcal{C}$ tensor	$\mathcal{C}_{\text{learn}}$
learnable and diagonalized $\mathcal{C}$ tensor	$\mathcal{C}_{\text{learn,diag}}$

Table 3-3: Types of Primal KMLSVD attention and their reference symbols or names.

3-3 Implementing Primal KMLSVD Attention

Having discussed the various theoretical components of using the primal dual KMLSVD formulation to describe self-attention in 3D, this section discusses how these parts are used in an implementation. In this section, it is assumed that the reader is familiar with the learning process of neural networks and the applications of primal attention as described in section 2-5.
Much like with primal attention, primal KMLSVD attention will largely follow the same logic. Primal KMLSVD attention will have a forward pass (estimation) and backward pass (corrections) using a custom loss function with a regularization term that ensure the KMLSVD structure is kept and a task based loss term which ensure learning from the dataset. The custom cost function used for primal KMLSVD is as follows where L denotes the task related loss (see section 2-6 for what a task related loss is), J_{KMLSVD} denotes the primal KMLSVD function (2-19) and $J_{\mathcal{S}}$ denotes a regularization term that enforced that $\mathcal{S}$ adheres to the property that $S_{(n)}S_{(n)}^T$ for each mode-n unfolding are all positive diagonal matrices (i.e., enforcing that $\mathcal{S}$ is indeed a MLSVD core tensor). Both J_{KMLSVD} and $J_{\mathcal{S}}$ are regularization terms that ensure the KMLSVD structure is adhered to.

$$\min \quad J_{total} := L + \eta_1 * J_{KMLSVD} + \eta_2 * J_{\mathcal{S}}. \tag{3-13}$$

J_s is calculated by computing the three $S_{(1)}S_{(1)}^T$, $S_{(2)}S_{(2)}^T$ and $S_{(3)}S_{(3)}^T$ terms, extracting each non-diagonal element, squaring said elements and summing them together as follows

$$J_{\mathcal{S}} := \left(\overline{S_{(1)}S_{(1)}^\top}\right)^2 + \left(\overline{S_{(2)}S_{(2)}^\top}\right)^2 + \left(\overline{S_{(3)}S_{(3)}^\top}\right)^2 \tag{3-14}$$

where the horizontal upper line "$\overline{\ }$" denotes the off diagonal terms only. The off-diagonal terms are squared in order to i) keep the cost positive and ii) have the optimal cost be 0. The underlying idea of adding $J_{\mathcal{S}}$ to the cost function is that it penalizes non-zero off diagonal terms, enforcing a MLSVD structure on the $\mathcal{S}$ tensor.
Note that the constraint of $S_{(n)}S_{(n)}^T$ being a diagonal positive matrix for each mode-n unfolding does not necessarily have to be enforced in the cost function. Another possible method is to initialize $\mathcal{S}$ randomly, then compute the MLSVD on $\mathcal{S}$ and then use the core tensor of the MLSVD decomposition (see equation (2-16)) on $\mathcal{S}$, seeing as this will enforce the positive diagonal matrix constraint. So, instead of $\mathcal{S}$ being used to define Primal KMLSVD attention, $\hat{\mathcal{S}}$ from equation 3-15 is used.

$$MLSVD(\mathcal{S}) = [\hat{U_{1,S}}, \hat{U_{2,S}}, \hat{U_{3,S}}, \hat{\mathcal{S}}] \tag{3-15}$$

This means that the choice of enforcing the MLSVD structure on $\mathcal{S}$ with either the cost function $J_{\mathcal{S}}$ or applying the MLSVD decomposition creates two separate variants of primal KMLSVD attention. One variant, the cost enforced KMLSVD attention, integrates the $J_{\mathcal{S}}$ term in the cost function and uses gradient based updates to get the desired $\mathcal{S}$ structure. The other variant, MLSVD enforced KMLSVD attention, applies the MLSVD on the $\mathcal{S}$ tensor itself to enforces the desired structure. These two aforementioned enforcement strategies are completely separate from the primal KMLSVD methods named in section 3-2 meaning that the 4 methods mentioned in section 3-2 can be either applied on the "cost enforced" or "MLSVD enforced" Primal KMLSVD attention, effectively creating 8 possible primal KMLSVD attention formulations.
The rest of the primal KMLSVD attention implementation are analogous to the primal attention implementation. So, to summarize:

1. Initialize learnable parameters W_q, W_k, W_1, W_2, W_3, $\mathcal{S}$ and/or $W_{1|X}$, $W_{2|X}$, $W_{3|X}$, and $\mathcal{C}$ depending on which version of primal KMLSVD attention is being used.

2. Do a forward step by computing the E_1, E_2 and E_3 matrices and concatenating them. The variables W_1, W_2, W_3, $\mathcal{S}$ and/or $W_{1|X}$, $W_{2|X}$, $W_{3|X}$, and $\mathcal{C}$ are also set as an output seeing as these variables are needed in calculating the loss.

3. Pass this forward step, along with the learnable parameters initialized in the first step, to the loss function 3-13 and computing the loss(3-13).

4. With the help of the computed loss in step 3, do a backward step and update all the learnable parameters. Then repeat steps 2-4 iteratively updating the parameters each time.

The η_1 and η_2 parameters of the cost function (3-13) are (tunable) hyperparameters.

3-4 Multi-headed Primal-Dual KMLSVD attention

Primal and Dual KMLSVD attention can both be adjusted for multi-headed attention. In essence, multi-headed Primal and Dual KMLSVD attention work exactly the same as standard multi-headed attention (see section 2-4 and figure 2-5). Multiple lower dimensional Primal or Dual KMLSVD self-attention mechanisms are applied on the input sequence and their outputs concatenated. Note that, as opposed to canonical self-attention, primal KMLSVD attention's amount of learnable parameter is heavily influenced by the number of heads. In order to understand exactly why, consider how the output of Primal KMLSVD attention is computed (equation 3-7 or figure 2-4a). Recall that the outputs of primal KMLSVD attention are the $E_1 \in \mathbb{R}^{N\times R}, E_2 \in \mathbb{R}^{N\times R}$ and $E_3 \in \mathbb{R}^{N\times R}$ matrices (see equation 2-23 or figure 2-4) with $R \leq M$. In multi-headed Primal KMLSVD attention there are now h separate lower dimensional Primal KMLSVD attention outputs $E_1^{(i)} \in \mathbb{R}^{N\times R_h}, E_2^{(i)} \in \mathbb{R}^{N\times R_h}$ and $E_3^{(i)} \in \mathbb{R}^{N\times R_h}$ with $R_h \leq \frac{M}{h}$, h the amount of chosen heads and $i \in [1, ..h]$.
Zooming in on a singular head. The feature maps $\Phi_1 \in \mathbb{R}^{N\times \frac{M}{h}}, \Phi_2 \in \mathbb{R}^{N\times \frac{M}{h}}$ and $\Phi_3 \in \mathbb{R}^{N\times \frac{M}{h}}$ (2-33,3-8,2-35) are made using the queries, keys and fibers $Q(X)^{(i)} \in \mathbb{R}^{N\times \frac{M}{h}}, K(X)^{(i)} \in \mathbb{R}^{N\times \frac{M}{h}}$ and $F(X)^{(i)} \in \mathbb{R}^{N\times \frac{M}{h}}$. Due to these queries, keys and fibers having a lower dimensional embedding, the feature maps will also have a lower dimensional embedding. Due to this, the $\mathcal{C} \in \mathbb{R}^{\frac{M}{h}\times\frac{M}{h}\times\frac{M}{h}}$ tensor also decreases is size (see figure 2-4a). This has a cascading effect, causing the $W_1 \in \mathbb{R}^{\frac{M}{h}\times\frac{M}{h}}, W_2 \in \mathbb{R}^{\frac{M}{h}\times\frac{M}{h}}, W_3 \in \mathbb{R}^{\frac{M}{h}\times\frac{M}{h}}$ and $\mathcal{S} \in \mathbb{R}^{R_h\times R_h\times R_h}$ tensor with $R_h \leq \frac{M}{h}$ to also decrease in size due to the lower dimensional embedding. This means that per head, the amount of learnable parameters is:

$$\left(\frac{M}{h}\right)^3 + 3\left(\frac{M}{h}\right)^2 + R_h^3. \tag{3-16}$$

A total amount of learnable parameters for the entire multi-headed Primal KMLSVD attention mechanism then is:

$$\frac{M^3}{h^2} + 3\frac{M^2}{h} + hR_h^3 \tag{3-17}$$

Comparing this to the total learnable parameters of Primal KMLSVD attention (see section 3-1-2 for the dimensionalities of the learnable parameters):

$$M^3 + 3M^2 + R^3. \tag{3-18}$$

Note that $R_h \leq R$ holds and therefore, for any number of heads greater than 1, the total amount of learnable parameters decreases.
Not only is the amount of learnable parameters affected by the number of heads, the computational complexity is also affected by the number of heads. In order to illustrate this, a simple example of computing the E_1 matrix, using multi-headed Standard Primal KMLSVD attention is done. In the table below 3-4, the computational complexity of each computation step is shown. The total computational complexity is as follows:

$$\mathcal{O}\left((\frac{M}{h})^2R_h^2 + N(\frac{M}{h})^3 + N(\frac{M}{h})^2R_h^2 + NR_h^3\right).$$

Note that if $R_h \leq R$ is taken into account, the computational complexity *decreases* as the number of heads increases. Whether this decrease in computational complexity affects the performance is not researched in this thesis.

Computation step	**Computational complexity**
$\overbrace{W_3 \otimes W_1}^{X}$	$\mathcal{O}\left(\left(\frac{M}{h}\right)^2 R_h^2\right)$
$\overbrace{\Phi_1 C_{(1)}}^{Y}$	$\mathcal{O}\left(N\left(\frac{M}{h}\right)^3\right)$
$\overbrace{XY}^{Z}$	$\mathcal{O}\left(N\left(\frac{M}{h}\right)^2 R_h^2\right)$
$\left(ZS_{(1)}^T\right)$	$\mathcal{O}\left(NR_h^3\right)$

Table 3-4: Computation steps and their corresponding computational complexities.

Chapter 4

Methods and Results

This chapter describes the different tests and discusses their results. Three separate tests were performed and are discussed in this thesis. The first test compared different primal Kernel Multi-linear Singular Value Decomposition (KMLSVD) attention variants (as described in sections 3-1-2 and 3-2), feature maps (2-33,3-8,2-35) and $\mathcal{S}$ enforcement strategies (see section 3-3) against each other. The second test compares Primal KMLSVD attention against Dual KMLSVD attention to verify if Primal KMLSVD attention really does improve the computation time and whether there is any meaningful accuracy difference between primal or dual KMLSVD attention. The third and final test compares the best performing Primal or Dual KMLSVD attention variant against primal attention as defined by Chen et al.[4] and standard self-attention as described in section 2-4.
Together, these test aim to answer the main question: "Can 3D Primal-Dual KMLSVD attention actually function as a self-attention mechanism?" and the sub-questions: "Can Primal-Dual KMLSVD attention be used as an improvement or alternative to self-attention?", "Is the Primal or Dual KMLSVD attention formulation more favorable in terms of computational efficiency and predictive performance?" and "Can alternative Primal KMLSVD attention formulations and feature maps improve training efficiency and predictive accuracy compared to the baseline approach?".
In section 4-1, the three main tests are described in detail. In section 4-2 the results of these three tests are discussed along with additional findings of smaller scale tests.

4-1 Test Setups

The aforementioned three tests are all performed on the 10 timeseries datasets shown in table 4-1. The sequence lengths, dimensionalities, and the number of classes of the datasets is shown in the table.

Dataset	Classes	Sequence Length	Dimensionality
EthanolConcentration	4	1751	3
FaceDetection	2	62	144
HandWriting	26	152	3
Heartbeat	2	405	61
JapaneseVowels	9	29	12
PemsSF	7	144	963
SelfRegulationSCP1	2	896	6
SelfRegulationSCP2	2	1152	7
SpokenArabicDigits	10	93	13
UWaveGestureLibrary	8	315	3

Table 4-1: Overview of datasets with number of classes, sequence length, and embedding dimension taken from the timeseries classifications website [7].

Each dataset is split up in a train, test and validation split. The models are trained on the train split and, after each complete pass of the train split, tested on the test split. Based on the results of the test split, specific hyperparameters (like η_1 and η_2 from the primal KMLSVD attention cost function (3-13)) are tuned. The validation split serves as a final validation of the model performance. The sizes of the train and test split for the timeseries datasets are dependent on the datasets used (as they come with built-in train and test splits[2]) but the choice for the validation split has been set to be the last 20% of the test split. So, the test split of the original dataset is further divided into 80% actual test split and 20% validation split. See figure 4-1 for a visual representation of how this dataset split is done. Note that different datasets use different train and test split sizes/percentages.

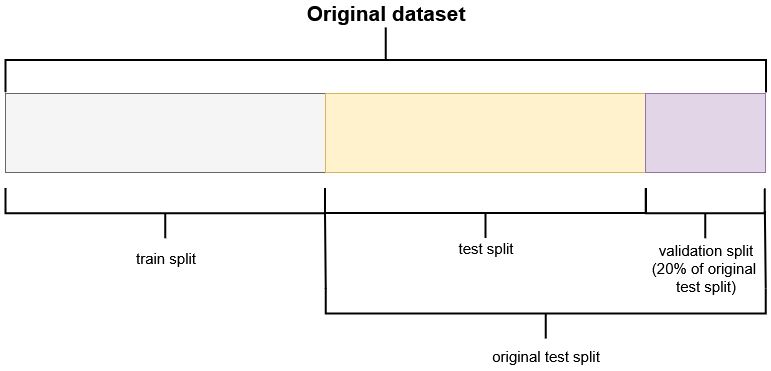


Figure 4-1: Visual representation of dataset splits

Each test is done multiple times in order to catch variability and eliminate randomness from the test results. So each iterations involves completely resetting the model parameters and re-starting the training from scratch. Each test uses the same evaluation criteria: (i) highest average accuracy, (ii) highest overall maximum accuracy, and (iii) training time. An exact description of the three metrics is shown below

1. **Highest average accuracy**: First, the highest accuracy per training run is measured and recorded (one training run being multiple epochs through a dataset and one epoch being an entire pass through the training set). Multiple training runs are performed and then, the average of these highest accuracies is taken.
2. **Highest maximum accuracy**: The overall highest accuracy is taken over all iterations and epochs. Important to note is that no average has been computed here.
3. **training time**: Per epoch, the training time is measured and recorded. The average of these measurements per epoch and training run is then computed.

Highest average accuracy is the most important metric because it gives a good idea on the overall performance. Highest maximum accuracy is in second place because, even though it is heavily influenced by random chance, it could be an indicator on the potential of the model. If, for instance, two models have roughly the same average accuracy, but one model has a way higher maximum accuracy, it could mean that the model with the higher maximum accuracy is overall better but suffers from numerical stability. Training time is set in last place because the Primal-Dual KMLSVD attention modules are coded by an author who is not very proficient in programming. This means that the code is not optimized for speed and some possible redundancies are present, slowing it down further. Furthermore, due to time constraints, different models were trained on different Graphics Processing Unit (video card) (GPU)'s. Although the GPU models were identical, their workloads varied; some GPU's were more heavily utilized at the time of training, which may have led to slower training speeds for certain models. Initially, the plan was to also record the GPU video ram usage and wattage of each self-attention type using built in software. However, these metrics did not seem to change at all per self-attention type. This, paired with the fact that multiple people were using the same GPU invalidated the two metrics. There was not enough time to fix this problem.
Finally, every test mentioned in this thesis makes use of multi-headed attention with 4 heads regardless of the transformer design, dataset, test or self-attention variant. The choice of 4 attention blocks was made because during initial testing, it was shown that 4 attention blocks achieved good results. While more or less blocks are certainly possible, 4 was still chosen.
All tests mentioned in this report are done on the "dcscgpuserver5" supplied by Delft Center for Systems and Control (DCSC). This server has 3 "Nvidia RTX A5000" GPU's.

The rough transformer design is schematically depicted in figure 4-2. A short explanation of what each block is and does is given below:
Input projection + Positional embedding: The input projection projects the input data to the feature-space used by the model and the "Positional Embedding" adds positional information to the input.
Stacked encoder blocks: The "Stacked Encoder blocks" consist of 4 self-attention mechanisms in series and at the end a convolutional layer.

Layer normalization and Classification head: Finally, the "Layer normalization" stabilizes and centers the output and the classification head flattens the sequence embeddings and transforms it into logits (output that encodes the likelihood of each possible class).
Within this transformer, multi-headed self-attention is applied (just as described in section 2-4 and 3-4.

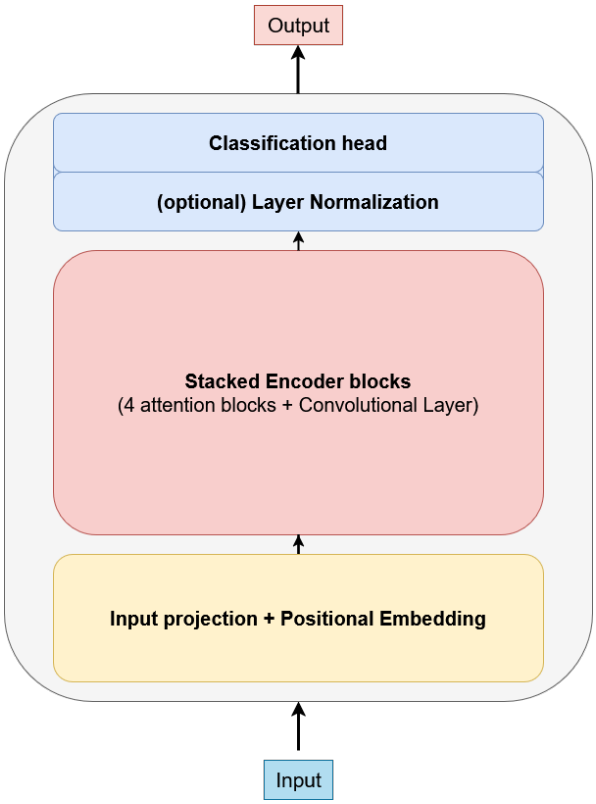


Figure 4-2: Transformer design used for testing timeseries datasets

The above transformer design is taken directly from Chen et al.'s code[4].

4-1-1 Test 1: Finding The Optimal Primal KMLSVD Attention Formulation

In the first test, different (multi-headed) primal KMLSVD attention formulations are compared to each other on 10 timeseries datasets (table 4-1) to see which formulations get the highest accuracy and/or lowest computational time. The first test aims to answer the research questions: "Can Primal KMLSVD self-attention actually function as a 3D self-attention mechanism?" and "Can alternative Primal KMLSVD attention formulations and feature maps improve training efficiency and predictive accuracy compared to the baseline approach?".
In total, 8 different primal KMLSVD attention variants are tested, these 8 arise from the 4 variants discussed in sections 3-2 and 3-1-2 and the 2 different $\mathcal{S}$ structure enforcement strategies mentioned in section 3-3. Seeing as the $\mathcal{S}$ enforcement strategies are entirely separate from the primal KMLSVD attention variants, the primal KMLSVD attention variants can be combined with either one of the $\mathcal{S}$ enforcement strategies creating 8 possible primal

KMLSVD attention formulations.
Each of the 8 attention formulations are applied on two different transformer designs based on figure 4-2. The first transformer design has all of its 4 attention blocks set to the specific primal KMLSVD attention formulation that is being tested and will be referred to as Full Layer Transformer (FLT) in this thesis. The second transformer design has its first 3 attention blocks set to canonical attention and only its last block is primal KMLSVD attention and will be referred to as Last Layer Transformer (LLT). This was done due to Chen et al. finding that primal attention got higher accuracies if only the last attention block of a transformer was set to primal attention and the previous blocks set to canonical attention[4].
Each combination of transformer design, primal KMLSVD formulation and dataset is trained and evaluated across 10 different independent runs. After each run, model parameters are reset to ensure independence. During training, the accuracy of the test set was tracked at every epoch (an epoch being an entire run through the training data set).
For each combination of transformer design, primal KMLSVD formulation and dataset, the cost hyperparameters η_1 and η_2 (3-13) are tuned to get the highest accuracy. This tuning is done by performing a grid search over $\{0.01, 0.1, 1, 5\}$ for both η_1 and η_2 and selecting the pair that yields the highest average accuracy. The chosen values per transformer design, primal KMLSVD formulation and dataset combination are shown in appendix A-1-4.
Finally, to facilitate a fair comparison and to test the expressive power of each primal KMLSVD attention formulation, each formulation is subjected to a "learnable parameter budget". This budget ensures that the amount of learnable parameters in each primal KMLSVD attention variant is roughly the same. An overview of the different self-attention mechanisms and the dimensionalities used within these mechanisms is given below. Note that for this test, the number of heads h has been set to 4.

Standard Primal KMLSVD Attention uses the following learnable parameters $\mathcal{S} \in \mathbb{R}^{h\times\tilde{R}_1\times\tilde{R}_1\times\tilde{R}_1}, W_1 \in \mathbb{R}^{h\times\frac{M_1}{h}\times\tilde{R}_1}, W_2 \in \mathbb{R}^{h\times\frac{M_1}{h}\times\tilde{R}_1}, W_3 \in \mathbb{R}^{h\times\frac{M_1}{h}\times\tilde{R}_1}, W_q \in \mathbb{R}^{h\times\frac{M_1}{h}\times M_1}, W_k \in \mathbb{R}^{h\times\frac{M_1}{h}\times M_1}$ and $W_f \in \mathbb{R}^{h\times\frac{M_1}{h}\times M_1}$ leading to a total amount of learnable parameters of

$$h\tilde{R}_1^3 + 3M_1\tilde{R}_1 + 3M_1^2 \tag{4-1}$$

Primal KMLSVD Attention with data dependent weights uses the following learnable parameters $\mathcal{S} \in \mathbb{R}^{h\times\tilde{R}_2\times\tilde{R}_2\times\tilde{R}_2}, W_1 \in \mathbb{R}^{h\times\hat{N}\times\tilde{R}_2}, W_2 \in \mathbb{R}^{h\times\hat{N}\times\tilde{R}_2}, W_3 \in \mathbb{R}^{h\times\hat{N}\times\tilde{R}_2}, W_q \in \mathbb{R}^{h\times\frac{M_2}{h}\times M_2}, W_k \in \mathbb{R}^{h\times\frac{M_2}{h}\times M_2}$ and $W_f \in \mathbb{R}^{h\times\frac{M_2}{h}\times M_2}$ leading to a total amount of learnable parameters of

$$h\tilde{R}_2^3 + 3h\tilde{N}\tilde{R}_2 + 3M_2^2 \tag{4-2}$$

Primal KMLSVD attention with learnable $\mathcal{C}$-tensor uses the following learnable parameters $\mathcal{S} \in \mathbb{R}^{h\times\tilde{R}_3\times\tilde{R}_3\times\tilde{R}_3}, W_1 \in \mathbb{R}^{h\times\frac{M_3}{h}\times\tilde{R}_3}, W_2 \in \mathbb{R}^{h\times\frac{M_3}{h}\times\tilde{R}_3}, W_3 \in \mathbb{R}^{h\times\frac{M_3}{h}\times\tilde{R}_3}, W_q \in \mathbb{R}^{h\times\frac{M_3}{h}\times M_3}, W_k \in \mathbb{R}^{h\times\frac{M_3}{h}\times M_3}$, $W_f \in \mathbb{R}^{h\times\frac{M_3}{h}\times M_3}$ and $\mathcal{C} \in \mathbb{R}^{h\times\frac{M_3}{h}\times\frac{M_3}{h}\times\frac{M_3}{h}}$ leading to a total amount of learnable parameters of

$$h\tilde{R}_3^3 + 3M_3\tilde{R}_3 + 3M_3^2 + \frac{M_3^3}{h^2} \tag{4-3}$$

Primal KMLSVD attention with learnable and hyper diagonal $\mathcal{C}$-tensor uses the following learnable parameters $\mathcal{S} \in \mathbb{R}^{h\times\tilde{R}_4\times\tilde{R}_4\times\tilde{R}_4}, W_1 \in \mathbb{R}^{h\times\frac{M_4}{h}\times\tilde{R}_4}, W_2 \in \mathbb{R}^{h\times\frac{M_4}{h}\times\tilde{R}_4}, W_3 \in \mathbb{R}^{h\times\frac{M_4}{h}\times\tilde{R}_4}, W_q \in \mathbb{R}^{h\times\frac{M_4}{h}\times M_4}, W_k \in \mathbb{R}^{h\times M_4\times\frac{M_4}{h}}$, $W_f \in \mathbb{R}^{h\times M_4\times\frac{M_4}{h}}$ and $\mathcal{C} \in \mathbb{R}^{h\times\frac{M_4}{h}\times\frac{M_4}{h}\times\frac{M_4}{h}}$ but the tensor $\mathcal{C}$ only has learnable parameters on its diagonal so the total amount of learnable parameters is

$$h\tilde{R}_4^3 + 3M_4\tilde{R}_4 + 3M_4^2 + M_4 \tag{4-4}$$

Note that the following relation between $\tilde{R}$ and M_n hold due to $\tilde{R}_n$ however, due to the fact that the mode-n-unfolding rank $\tilde{R}_n$ is unknown, it has the following relation:

$$\tilde{R}_n \leq \frac{M_n}{h}. \tag{4-5}$$

Below, in table 4-2 an overview of the chosen dimension sizes is given along with table 4-3 giving an overview of the total amount of learnable variables per self-attention design.

Dimensionalities	**Chosen Size**
$M_1, M_2, M_3, M_4, \tilde{R}_1, \tilde{R}_2, \tilde{R}_4$	20
$\tilde{R}_1, \tilde{R}_2, \tilde{R}_4, \hat{N}$	5
$\tilde{R}_3$	3

Table 4-2: List of variables and their chosen sizes

Recall that the number of heads h is set to 4. With this is mind, the final choice of dimensions is shown in table 4-3.

primal KMLSVD variant	**Total number of learnable parameters**
Standard	2000
data dependent weights	2000
learnable $\mathcal{C}$ tensor	1988
learnable and hyper diagonal $\mathcal{C}$ tensor	2020

Table 4-3: The 4 self-attention variants used in Test 1 along with their total amount of variables

The embedding sizes (M_1, M_2, M_3, M_4) were all set to 20 due to preliminary experiments showing that this value provided good accuracy scores across all datasets without compromising too much on computational speed. Additionally, using equal embedding sizes ensures a fairer comparison by controlling for potential effects of differing embedding dimensionalities.

4-1-2 Test 2: Comparing Primal to Dual KMLSVD Attention

In the second test, the best performing primal KMLSVD attention formulation, found in test 1, will be compared to dual KMLSVD attention as described in subsection 3-1-1. For a fair comparison, the amount of learnable variables are also subjected to a learnable parameter

[1] See table 4-2 for an overview of the final choice of dimension sizes.

budget. This test tries to answer the research questions: "Can Dual KMLVD attention actually function as a 3D self-attention mechanism?" and "Is the Primal or Dual KMLSVD attention formulation more favorable in terms of computational efficiency and predictive performance?".
Unfortunately, due to the extremely long training time, this test has not been completed for all datasets. As such, Primal KMLSVD attention is considered to be better than Dual KMLSVD attention due to the significant difference in training time. The preliminary results of the completed datasets are still shown in table A-13.

4-1-3 Test 3: Comparing Primal KMLSVD Attention To Primal Attention And Canonical Self-Attention

The third test aims to compare the "winning" Primal KMLSVD attention formulation, found in test 1 4-2-1, to primal attention[4] and canonical self attention[24]. The goal is to find out whether primal KMLSVD attention can be used as an improvement or alternative to canonical self-attention and primal attention. Seeing as no hyperparameter tuning is necessary, the three attention types are trained on the train *and* test split. The accuracy metrics are now measured on the validation set.
In this test, three transformer models are compared to each other. These three transformer models are identical in design to the transformer model used in test 1 (see figure 4-2) except for the self-attention blocks used within the stacked encoder blocks. The three different self-attention blocks are:

1. The first three blocks are canonical-self attention and the last block is learnable $\mathcal{C}$ tensor Primal KMLSVD attention using the Frobenius norm feature map (see section 4-2-1 for an overview and discussion).

2. The first three attention blocks are canonical self-attention and the last block is primal attention using the cosine similarity feature map.

3. All 4 of the attention blocks are set to canonical attention.

Once again, a learnable parameter budget is applied.
Primal KMLSVD attention with learnable $\mathcal{C}$-tensor see section equation 4-3 and tables 4-2, 4-3.
Primal KSVD attention has the following learnable parameters $W_1 \in \mathbb{R}^{h \times \frac{M_2}{h} \times \tilde{R}_2}, W_2 \in \mathbb{R}^{h \times \frac{M_2}{h} \times \tilde{R}_2}, S \in \mathbb{R}^{h \times \frac{\tilde{R}_2}{h} \times \frac{\tilde{R}_2}{h}}, W_q \in \mathbb{R}^{h \times \frac{M_2}{h} \times M_2}$ and $W_k \in \mathbb{R}^{h \times \frac{M_2}{h} \times M_2}$ with S a diagonal matrix, leading to a total amount of learnable parameters:

$$2M_2\tilde{R}_2 + 2M_2^2 + \tilde{R}_2. \tag{4-6}$$

Canonical self-attention uses the following learnable parameters $W_q \in \mathbb{R}^{h \times \frac{M_3}{h} \times M_3}, W_v \in \mathbb{R}^{h \times \frac{M_3}{h} \times M_3}$ and $W_v \in \mathbb{R}^{h \times \frac{M_3}{h} \times M_3}$ leading to a total amount of learnable parameters:

$$3M_3^2 \tag{4-7}$$

The final choice for the dimensionalities is shown in table 4-4 and the corresponding total amount of learnable per self-attention mechanism is shown in table 4-5.

Dimensionalities	Chosen Size
M_1	20
M_2	28
M_3	28
$\tilde{R}_1$	3
$\tilde{R}_2$	7

Table 4-4: List of variables and their chosen sizes

As with test 1, the datasets might have different embedding dimensions, as such the dataset inputs (see table 4-1) are first passed through the input projection layer.

Self-attention variant	Total number of learnable parameters
Primal KMLSVD	1988
Primal KSVD	1967
canonical $\mathcal{C}$ tensor	2352

Table 4-5: Total amount of learnable parameters per self-attention variant used in test 3

Initially, the plan was to also record the Video Random Access Memory (VRAM) and Watt usage of Primal KMLSVD attention, Primal attention and Canonical self-attention on the ten different datasets and add the result to a new table. The VRAM usage and Watt usage would be recorded by the onboard software of the TU Delft GPU server. However, the VRAM and watt usage were identical across attention variants and only seemed to differ per dataset. What exactly the cause was for this behavior is unknown and due to a lack of time, no further attempts at measuring it were made. Unfortunately, this means that the only metric of "efficiency" is time.
During testing, it was found that the dimensionality M of the self-attention mechanism strongly affects the accuracy. Especially noteworthy is that, for some datasets, setting this dimensionality higher lead to better results and for other dataset, the inverse was the case. To properly account for this strange behavior, an additional version of test 3 was performed where, instead of a learnable parameter budget, an equal dimensionality within the transformer is enforced (see equation 4-8).

$$\begin{aligned} M_1 = M_2 = M_3 = M_4 = 20 \\ \tilde{R}_1 = \tilde{R}_2 = \frac{M_1}{h} = \frac{20}{4} = 5. \end{aligned} \tag{4-8}$$

With this additional test, both the expressive power of the self-attention mechanisms (as ensured by keeping the total amount of learnable parameters equal over different self-attention mechanisms) and the "power" of the self-attention mechanisms is verified. By setting the dimensionality to be the same over different attention-mechanisms, the effects of operating in a higher or lower dimensional space is eliminated. Taken together, these two variations of

test 3 (equalizing the number of parameters and equalizing the dimensionality) enable a more complete and fair comparison between self-attention mechanisms.

4-2 Results

The results of the tests, described in section 4-1, are discussed in this section. Subsections 4-2-1, 4-2-2 and 4-2-3 discuss the results of test 1 (4-1-1), test 2 (4-1-2) and test 3 (4-1-3) respectively. In subsection 4-2-4 additional results of other small scale tests are discussed.

4-2-1 Test 1 Results

The results of test 1 are shown in table 4-6. In this table, the Frobenius norm feature map(3-8) and Multi-linear Singular Value Decomposition (MLSVD) enforced $\mathcal{S}$ structure is used, as this combination lead to the highest average and maximum score. For the results of other feature maps and $\mathcal{S}$ enforcement strategies, see appendix A-1

MLSVD enforced $\mathcal{S}$ structure using Frobenius norm feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	38.3	41.2	245	36.6	39.8	99	39.4	42.2	502	38.5	40.7	190
Face Detection	67.5	69.2	800	60.2	61.9	899	67.5	69.6	$1.33{\cdot}10^3$	67.6	68.7	667
Handwriting*	28.4	31.5	79.7	9.09	10.6	48.9	28.0	31.2	115	27.6	31.5	63.2
Heartbeat	74.6	76.2	43.4	75.2	78.1	81.9	74.7	76.8	108	74.8	77.4	26.2
Japanese Vowels	96.6	98.7	28.6	96.3	98.7	45.7	98.0	99.0	112	96.6	98.0	26.1
PEMS-SF-	80.6	84.9	24.7	76.6	82.7	27.8	91.0	93.5	196	81.2	85.6	32.4
Self Regulation SCP1	91.3	92.5	67.2	88.3	92.2	48.8	91.5	92.8	174	91.4	93.2	89.6
Self Regulation SCP2	54.8	57.6	88.4	55.6	62.5	50.1	55.0	60.4	222	55.3	58.3	79.6
Spoken Arabic Digits	98.1	98.7	452	94.6	95.1	740	98.4	98.8	$1.05{\cdot}10^3$	98.3	98.6	445
U Wave Gesture Library	86.3	87.5	14.2	77.3	81.6	22.0	86.2	89.1	66.2	86.5	89.1	13.5
Average	71.4	73.6	183	70.0	70.3	206	**73.0**	**75.3**	388	71.8	74.1	**163**
	Full layer											
Ethanol Concentration	34.9	36.5	135	36.2	38.9	174	35.1	37.0	228	34.4	37.0	77.0
Face Detection*	68.4	69.5	$1.95{\cdot}10^3$	60.0	61.2	$2.02{\cdot}10^3$	68.3	70.5	$3.30{\cdot}10^3$	68.2	69.8	$1.43{\cdot}10^3$
Handwriting	29.5	31.0	240	8.37	9.85	236	28.7	30.9	329	28.9	30.9	177
Heartbeat	75.1	76.8	102	75.0	76.8	142	75.2	77.4	226	75.2	78.1	53.9
Japanese Vowels	96.5	99.0	78.4	95.9	98.7	120	97.8	98.7	280	96.7	98.0	67.9
PEMS-SF	80.9	87.1	52.1	75.0	79.9	52.3	90.0	95.0	477	79.6	84.2	56.7
Self Regulation SCP1	89.1	90.1	63.3	89.4	92.8	88.6	89.8	91.1	163	88.8	90.1	81.1
Self Regulation SCP2	54.2	57.6	76.7	54.9	63.2	81.4	54.8	57.6	274	54.0	56.3	62.7
Spoken Arabic Digits	98.0	98.4	$1.05{\cdot}10^3$	95.1	95.7	$1.85{\cdot}10^3$	98.1	98.8	$2.82{\cdot}10^3$	98.0	98.3	$1.16{\cdot}10^3$
U Wave Gesture Library	85.8	87.5	35.0	74.8	78.9	57.1	86.0	87.9	168	86.0	87.5	31.2
Average	71.0	73.2	374	66.5	70.0	482	72.4	74.5	827	71.0	73.0	320

Table 4-6: Comparison of mean accuracies, max accuracies, and average learning times (in seconds) across datasets for two transformer variants using the Frobenius norm feature map: (i) replacing only the last self-attention layer with primal KMLSVD, and (ii) using only primal KMLSVD throughout.

As shown in Table 4-6, the i)LLT with the ii)$\mathcal{C}_{\text{learn}}$ configuration, employing the iii)MLSVD $\mathcal{S}$ enforcement strategy and the iv) Frobenius norm feature map (3-8), achieves the highest mean accuracy across all 10 timeseries datasets. This performance, however, comes at the cost of significantly longer training times compared to other primal KMLSVD formulations. These longer training times are inherent to the $\mathcal{C}_{\text{learn}}$ Primal KMLSVD attention formulation,

as the same behavior is observed consistently across all transformer types, feature maps and $\mathcal{S}$ enforcement strategies. Consequently, this formulation is considered the "winner" and used in tests 2 and 3.
It should be noted, however, that the Primal KMLSVD attention tests using the cost $\mathcal{S}$ enforcement strategy could not be fully completed in time. The partial results that were obtained are depicted in Appendix A-1 (specifically, tables A-1, A-3 and A-5). These partial results suggest that the i)LLT with the ii) standard Primal KMLSVD attention configuration or $C_{\text{learn,diag}}$ configuration, employing the iii)cost $\mathcal{S}$ enforcement strategy and the iv) Frobenius norm feature map performs the best accuracy wise with the added benefit of significantly reduced training time compared to its MLSVD $\mathcal{S}$ enforced counterpart. However, because not all the experiments for the cost enforced $\mathcal{S}$ structure were completed in time, these results are considered inconclusive Therefore, the i) LLT with the ii) $\mathcal{C}_{\text{learn}}$ configuration, employing the iii) MLSVD $\mathcal{S}$ enforcement strategy and the iv) Frobenius norm feature map (3-8) remains the "winner" of Test 1. This conclusion should be taken with a grain of salt until full cost enforced tests can be conducted. Nevertheless, just to be sure, the cost enforced $\mathcal{S}$ setup has still been included in Test 3 4-2-3.
While a reasonable argument could be made in favor of the $\mathcal{C}_{\text{learn,diag}}$ formulation (particularly when used with the cosine similarity or Frobenius norm feature maps and the MLSVD $\mathcal{S}$ enforcement strategy) this variant ultimately was not selected as the "winning" configuration. This setup was considered because its accuracy scores are only marginally lower than those of the $\mathcal{C}_{\text{learn}}$ formulation, while offering substantially shorter training times. Nevertheless, as discussed in subsection 4-1-1, accuracy is assigned the highest importance among the evaluation metrics, whereas training time is the least important. Furthermore, the increased learning times of $\mathcal{C}_{\text{learn}}$ configurations, although notable, is not large enough to consider it too limiting. It should also be noted that, due to the author's lack of programming knowledge, this large training time could be significantly decreased when coded by a more competent programmer.

Additional Test 1 Findings

Besides the $\mathcal{C}_{\text{learn}}$ configuration, employing the MLSVD $\mathcal{S}$ enforcement strategy and the Frobenius norm feature map (3-8) being the "winner", some additional observations and conclusions can be drawn from this test.
First, the data dependent setup $W_{n|X}$ (as defined in section 3-2) consistently has the lowest accuracy scores across any feature map, transformer design or $\mathcal{S}$ enforcement strategy. In fact, its performance is consistently lower than that of the standard, unmodified Primal KMLSVD attention formulation. This indicates that the data-dependent modification, at least in its current implementation, does not provide any benefit and can be considered ineffective.
Second, the LLT consistently has higher accuracy scores and lower training times. Whether this is due to normal self-attention being faster and more accurate than Primal KMLSVD attention is looked at in Test 3 4-2-3.
Third, the effect of the feature maps on the accuracy scores is shown in table 4-7. Note that, when looking at the average accuracies over the feature maps, the feature maps do not have a noticeable effect on the accuracy scores nor training times with the exception of the SM+ feature map. The SM+ feature map, by far, has the highest training times. Due to this, the

Frobenius norm feature map and Cosine feature map are considered "valid" feature maps and the SM+ feature map is considered to be invalid due to it requiring significantly more time to attain equal results.
Finally, looking at the η_1, η_2 hyperparameter values that resulted in the highest accuracy scores, some strange behavior is seen. Note that often the values $\eta_1 = 0$ and/or $\eta_2 = 0$ result in the highest accuracy scores (see appendix A-1-4). To verify whether this observation was a result of the random nature of deep learning or some structural effects of the η_1, η_2 parameters, a bar plot for each dataset was made. These plots show the mean accuracy, the range where 68% of accuracy values lie in and the maximum accuracy as a result of each combination of η_1 and η_2 value (see figure 4-3 for an example). The plots showed that for the datasets EthanolConcentration, FaceDetection, PemsSF and JapaneseVowels the $\eta_1 = 0$ hyperparameter value gave significantly better results when applied on the FLT. However, when using the LLT, the hyperparameter values didn't seem to affect the accuracy scores that much. The variation that was detected in the LLT model is likely a result of the random nature of deep learning, as opposed to the effects of the hyperparameter values. The finding that η_1, η_2 had little to no effect on the LLT performance and $\eta_1 = 0$ had significant effect on the FLT performance is quite surprising. This would mean that the primal KMLSVD structure and/or the $\mathcal{S}$ structures are not necessary or even important in order to get good accuracy scores. This, in turn, could mean that a KMLSVD framework (whether primal or dual) is not necessary in order to define 3D self-attention. An attempt at setting up a 3D self attention framework not using the KMLSVD formulation is described in section 4-2-4.

Variant	Frobenius Feature Map						Cosine Feature Map						SM+ Feature Map					
	Last Layer			Full Layer			Last Layer			Full Layer			Last Layer			Full Layer		
	Mean	Max	Time	Mean	Max	Time	Mean	Max	Time	Mean	Max	Time	Mean	Max	Time	Mean	Max	Time
Normal	71.4	73.6	183	71.0	73.2	374	72.7	74.6	246	72.3	74.9	510	72.7	74.9	329	72.2	74.0	752
$W_{n\|X}$	70.0	70.3	206	66.5	70.0	482	67.6	70.7	177	66.8	69.8	315	67.8	69.8	320	66.7	69.8	914
$\mathcal{C}_{\text{learn}}$	73.0	75.3	388	72.4	74.5	827	72.9	75.1	366	71.6	74.0	781	72.8	74.9	400	72.3	74.0	859
$\mathcal{C}_{\text{learn,diag}}$	71.8	74.1	163	71.0	73.0	320	72.6	74.8	215	72.4	74.3	433	72.7	75.0	327	72.4	74.2	625
Average	**71.6**	73.3	**235**	70.2	72.7	501	71.5	**73.8**	251	70.8	73.2	510	71.5	73.7	344	70.9	73.0	788

Table 4-7: Average mean accuracy, max accuracy, and training time across three feature maps for each Primal KMLSVD attention formulation, comparing Last Layer and Full Layer settings. The final row reports overall averages across all formulations.

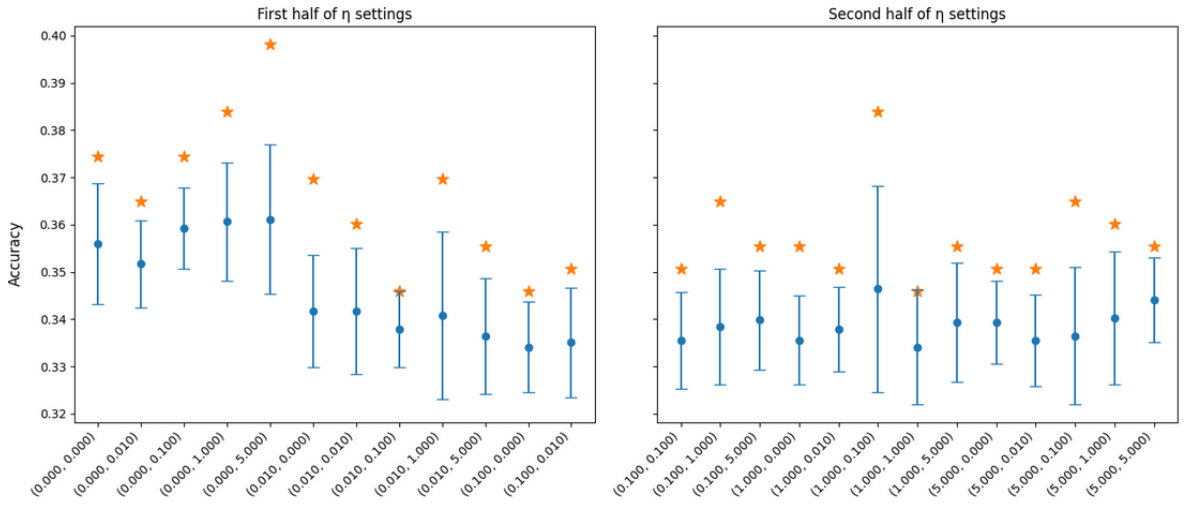


Figure 4-3: Mean accuracy (blue dot), variance (blue bar) and maximum accuracy (star) attained when using different η_1 and η_2 values on the Ethanol Concentration dataset. On the y-axis is the accuracy, and on the x-axis are the different η_1 and η_2 combinations used during testing.

4-2-2 Test 2 Results

Test 2 aims to find the best dual KMLSVD setup (last transformer layer vs all transformer layers set to dual KMLSVD attention). As such, no use of the validation split has been made. All accuracies mentioned in the results are based on the test split only.
As mentioned in section 4-1-2, the training time for Dual KMLSVD attention was too large to generate results for each of the 10 datasets in time, making the Dual KMLSVD attention formulation infeasible for self-attention. Nevertheless, the results are shown in appendix A-2. The training times were so long, in fact, that the datasets had to be sub-sampled to reduce the training time. This, in turn, affected the accuracy scores.

4-2-3 Test 3 Results

The results for test 3 are shown in table 4-8. Primal KMLSVD attention has the lowest accuracy scores and highest training times across the two tests. Due to these results, primal KMLSVD attention cannot be used as an improved version of self-attention.

Comparison of Attention Mechanisms using Frobenius Norm Feature Map

	LPB								
	$\mathcal{C}_{\text{learn}}$ Primal KMLSVD attention			Primal Attention			Canonical self-attention		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	32.3	40.4	$1.06 \cdot 10^3$	33.5	40.4	868	35.4	42.3	843
Face Detection	74.5	76.6	$7.09 \cdot 10^3$	72.6	74.3	$4.70 \cdot 10^3$	74.1	76.0	$3.65 \cdot 10^3$
Handwriting	54.3	57.7	$1.02 \cdot 10^3$	54.9	57.7	614	54.7	59.4	424
Heartbeat	72.7	82.9	$1.57 \cdot 10^3$	72.2	80.5	$1.26 \cdot 10^3$	73.7	80.5	$1.07 \cdot 10^3$
Japanese Vowels	95.7	97.3	910	95.0	97.3	532	95.5	97.3	388
PEMS-SF	91.5	94.2	$1.08 \cdot 10^3$	91.5	94.1	845	92.1	94.1	555
Self Regulation SCP1	92.2	96.6	$1.26 \cdot 10^3$	92.1	94.8	960	93.0	98.3	850
Self Regulation SCP2	48.3	55.6	462	49.7	52.8	366	50.8	52.8	345
Spoken Arabic Digits	99.2	99.5	$3.94 \cdot 10^3$	99.2	99.8	$1.91 \cdot 10^3$	99.3	99.5	$1.29 \cdot 10^3$
U Wave Gesture Library	94.4	98.4	582	96.1	98.4	313	94.8	98.4	237
Average	75.5	79.9	$1.90 \cdot 10^3$	75.7	79.0	$1.24 \cdot 10^3$	76.3	79.9	965
	ED								
	$\mathcal{C}_{\text{learn}}$ Primal KMLSVD attention			Primal Attention			Canonical self-attention		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	28.7	36.5	816	29.8	34.6	658	31.0	34.6	654
Face Detection	66.6	70.2	$3.64{\cdot}10^3$	66.4	70.0	$2.67{\cdot}10^3$	67.2	68.9	$2.16{\cdot}10^3$
Handwriting	26.9	28.8	451	25.2	28.8	236	25.6	28.2	185
Heartbeat	82.2	85.4	552	82.4	87.8	434	81.0	85.4	377
Japanese Vowels	95.1	97.3	291	95.3	97.3	161	95.7	97.3	119
PEMS-SF	86.5	91.2	599	85.3	91.2	457	83.2	88.2	423
Self Regulation SCP1	91.6	93.1	725	91.2	94.8	576	91.6	93.1	524
Self Regulation SCP2	45.6	52.8	579	43.3	52.8	443	49.4	58.3	414
Spoken Arabic Digits	98.1	98.9	$2.71{\cdot}10^3$	97.9	98.4	$1.28{\cdot}10^3$	98.3	98.6	891
U Wave Gesture Library	85.3	87.5	228	86.1	87.5	120	86.6	89.1	93
Average	70.7	74.2	$1.06{\cdot}10^3$	70.3	74.3	704	71.0	74.2	584

Table 4-8: Comparison of mean validation accuracy, max validation accuracy, and training time across datasets for three attention mechanisms (canonical self-attention, primal attention, and primal KMLSVD attention) under two settings: learnable parameter budget and equal dimensionality, using the Frobenius norm feature map.

First, even though Primal KMLSVD attention has lower accuracy scores and higher training times than the other two self-attention types, it still attains comparable accuracy scores to primal attention and canonical self-attention and thus could serve as an alternative to self-attention. It just does not make sense to do so due to its lower accuracy and longer training times.

Second, since it has been demonstrated that Primal KMLSVD attention can function as a self-attention mechanisms, it could find more suitable application in applications where its 3D structure is more advantageous. For example, applying Primal KMLSVD attention on a 3D dataset (e.g., color images) or datasets where different types of data are used (e.g., camera, radar, and lidar measurements in autonomous driving). However, to make such applications possible, it could be necessary to design a new transformer architecture tailored to 3D or multimodal inputs and to re-define or restructure how Primal KMLSVD self-attention is integrated within the model. This could in turn involve adapting the Primal KMLSVD attention definition, mentioned in sections 3-1-2 and 3-2, to more accurately fit these types of data.

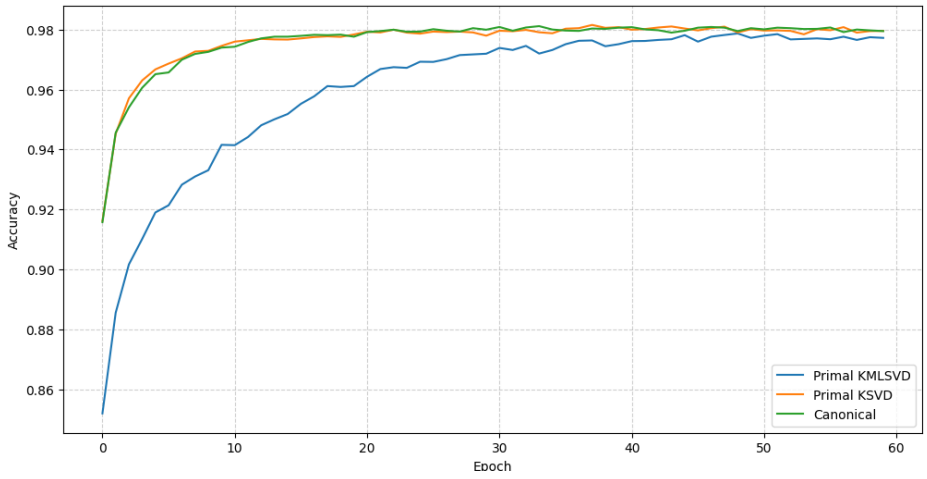
Third, when examining the results in Table 4-8, the difference in accuracy scores between the Learnable Parameter Budget (LPB) and Equal Dimensionality (ED) versions of Test 3

stands out as unexpected. The lower accuracy scores of Primal attention and Canonical self-attention for the ED case is to be expected, since both operate in a lower dimensional space and significantly fewer learnable parameters compared to their LPB counterparts. However, the Primal KMLSVD attention formulation shows strange behavior. Recall that the only difference between the Primal KMLSVD attention formulation used in LPB and ED is the size of the $\hat{R}$ dimension (see table 4-4 and equation 4-8). This behavior is counterintuitive: one would typically expect that reducing the number of learnable parameters, while keeping the embedding dimension of data fixed at $M = 20$, would lead to lower training times and lower accuracy scores. However, the very opposite is the case, reducing $\hat{R}$ leads to an increase in both training time and accuracy. The particularly strong effect on accuracy is surprising and suggests that the behavior may not be due to random fluctuations, especially considering that the accuracy scores of LPB are consistently higher than that of ED. Multiple iterations of this test were performed with the same effect. Whether this phenomenon arises from hardware or software artifacts (e.g., GPU availability or coding errors) or whether it reflects an inherent property of the Primal KMLSVD formulation itself remains unclear and warrants further research.
Fourth, Primal KMLSVD attention learns "slower" than primal attention and canonical self-attention. Two examples of this slower learning behavior are depicted in figure 4-4. In this figure, the accuracy curve on the training data of the Spoken Arabic Digits and Self Regulation SCP2 datasets over 60 epochs is shown. In both these graphs, the accuracy curve of Primal KMLSVD attention (blue), Canonical self-attention (green) and Primal attention (yellow) is shown. Note that for both datasets, the Primal KMLSVD attention curve converges and rises slower than the curves belonging to Primal attention and Canonical self-attention. This slower learning behavior could explain the fact that Primal KMLSVD attention attains lower accuracy scores than its Primal attention and Canonical self-attention counterparts.

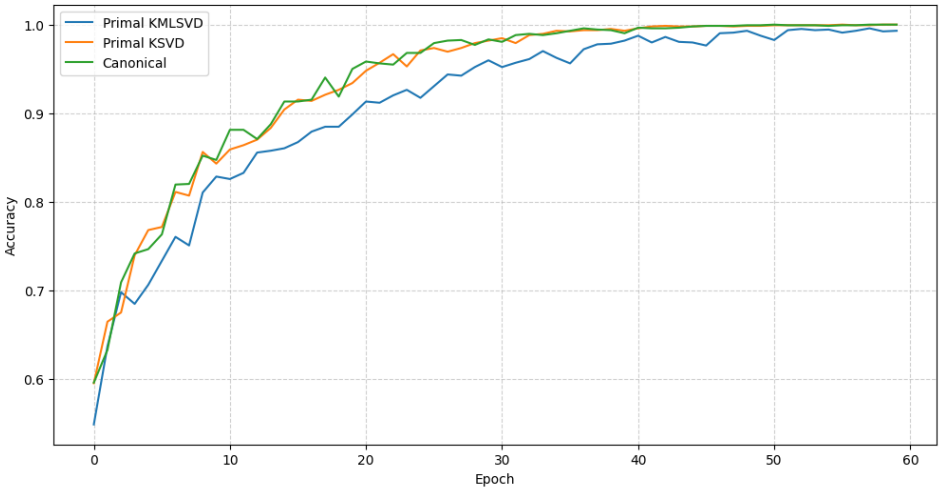
Additional cost enforced test

To ensure that the best-performing Primal KMLSVD attention formulation was used for evaluation, test 3 (both LPB and ED) was repeated using the i)$C_{\text{learn,diag}}$ Primal KMLSVD attention with ii)cost enforced $\mathcal{S}$ strategy and iii)LLT design using the iv)Frobenius norm feature map(3-8). This was done because, due to time constraints during test 1, the comparison between cost enforced and MLSVD enforced $\mathcal{S}$ Primal KMLSVD attention (see Section 3-2) could not be fully completed for all datasets.
The corresponding results are presented in Appendix A-3 and table A-14. The cost enforced $\mathcal{S}$ variant attains higher accuracy scores and lower training times than its MLSVD enforced counterpart. In several datasets, the cost–enforced Primal KMLSVD attention even matches or surpasses canonical self-attention and primal attention. These results further reinforce the initial observation of cost enforced $\mathcal{S}$ Primal KMLSVD attention being the better option. The problem of the longer training times, however, is still present for the cost enforced $\mathcal{S}$ formulation. That is why, even though cost enforced Primal KMLSVD attenion outperforms canonical self-attention in the LPB setting and performs roughly the same in the ED setting, it still cannot be considered an improvement or viable alternative to canonical self-attention due to its substantially higher training time.
It should also be noted that, although identical training settings were used, slight accuracy

differences appear between the Primal attention and canonical self-attention results reported in table A-14 and those in table 4-8. However, the accuracy differences are small enough that it can reasonably be attributed to the inherent randomness of deep learning. Finally, the same strange behavior of Primal KMLSVD attention performing worse with more learnable parameters under the same dimensionality is seen in both MLSVD enforced primal KMLSVD attention (table 4-8) and cost enforced Primal KMLSVD attention (table A-14), further suggesting that this phenomenon is an inherent property of the Primal KMLSVD formulation.



(a) Training accuracy curve for the Spoken Arabic Digits timeseries dataset over epochs for three self-attention formulations: i) Primal KMLSVD attention, ii) Primal attention and iii) Canonical self-attention



(b) Training accuracy curve for the Self Regulation SCP2 dataset over epochs for three self-attention formulations: i) Primal KMLSVD attention, ii) Primal attention and iii) Canonical self-attention

Figure 4-4: Example of two training accuracy curves over epochs for the three self-attention mechanisms used in Test 3.

4-2-4 Additional Tests

Some additional smaller scale tests are described in this section. These smaller scale tests were all applied on the Pems-SF and/or Japanese Vowels datasets (see table 4-1) due to these datasets being relatively fast to train on.

Rank and Energy of the 3D Attention Score tensor

Chen et al. found that the rank of the attention matrix decreases with depth in the attention blocks and that the first few singular values contribute more to the mean cumulative explained variance in later layers [4]. To examine whether similar behavior appears in the 3D KMLSVD attention setting, a small-scale test was conducted using Dual KMLSVD attention on the PEMS-SF dataset. The Dual KMLSVD framework was chosen because it computes the attention tensor explicitly, unlike the Primal variant.
Seeing as the attention kernel is now 3-dimensional, two aspects were analyzed:

1. The rank of the mode-1, mode-2 and mode-3 unfoldings of the attention tensor

2. The mean cumulative explained variance derived from the MLSVD

Across multiple runs, the ranks of the three unfoldings remained largely unchanged across different attention blocks. However, each unfolding did exhibit a low-rank structure that was generally lower than the dataset's embedding dimension, $R_n < M$ where Rn is the rank of the mode-n unfolding and M is the embedding dimension of the data.
To study the mean cumulative explained variance, the MLSVD of the attention kernels was computed. Specifically, the core tensor $\mathcal{S}$ (2-18) is used to derive the mode specific singular values. For each mode (i.e., n=1,2,3) the matrix $S_{(n)}S_{(n)}^T$ is computed and the diagonal entries of these matrices are treated as the singular values corresponding to each mode. These singular values are in turn used to compute the mean cumulative variance to see how the information is distributed. Seeing as multiple attention kernels are computed, the mean cumulative variance curve is computed for each attention kernel and then averaged to showcase the general trend. Figure 4-5 illustrates the cumulative mean cumulative explained variance for each mode unfolding of the attention tensor (left: mode-1, middle: mode-2, right: mode-3). The x-axis represents the singular value index, while the y-axis indicates the cumulative proportion of the total variance (or energy) captured up to that singular value.
Overall, the results suggest that a general trend of shaper mean cumulative explained variance in deeper attention blocks, as indicated by the steeper rise of the curves, holds particularly in the last layer. However, that each deeper attention block has a steeper mean cumulative explained variance is not the case as in some cases, earlier layers have a steeper rise than later ones. In the left most graph, for instance, layer 2 seems to have a less steep curve than layer 0 and layer 1.

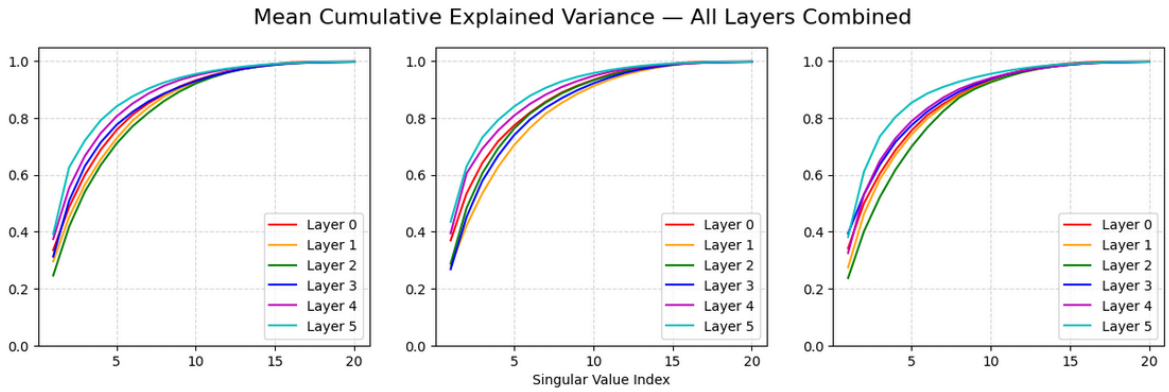


Figure 4-5: Mean cumulative explained variance averaged over the different attention kernel tensors

Non-KMLSVD Framework for 3D attention

To verify if a 3D self-attention mechanism is possible without using the KMLSVD framework (whether primal or dual), an alternate formulation was explored that is heavily based on the dual KMLSVD attention framework 3-1-1. This formulation replaces MLSVD related components (i.e. $U_1, U_2, U_3, \mathcal{S}$) with learnable parameters matrices and tensors.
The starting point is the computation of the 3D non-linear attention kernel(3-3) $\mathcal{K} \in \mathbb{R}^{N \times N \times N}$ non-linearized with the Cubemax function(3-5). In standard dual KMLSVD attention, the outputs are the E_1, E_2 and E_3 matrices which are generated with the help of the factor matrices U_1, U_2, U_3 and core tensor $\mathcal{S}$. These factor matrices and tensors are in turn generated by applying the MLSVD on $\mathcal{K}$. In the proposed modification, these factor matrices and the core tensor are replaced with learnable parameters that are initialized with a normal distribution. For simplicity in this experiment, no multi-headed self-attention was used and the ranks of all mode-n-unfoldings were set to be equal to the embedding dimension:

$$R_1 = R_2 = R_3 = M = 20 \tag{4-9}$$

Three possible configurations were considered:

1. Replace $U_1 \in \mathbb{R}^{N \times M}, U_2 \in \mathbb{R}^{N \times M}, U_3 \in \mathbb{R}^{N \times M}$ and $\mathcal{S} \in \mathbb{R}^{M \times M \times M}$ with learnable parameter $W_1^a \in \mathbb{R}^{N \times M}, W_2^a \in \mathbb{R}^{N \times M}, W_3^a \in \mathbb{R}^{N \times M}$ and $\mathcal{T}^a \in \mathbb{R}^{M \times M \times M}$ where N is the sequence length of the input data and M the embedding dimension of the input data.

2. Replace $(U_3 \otimes U_2) \in \mathbb{R}^{N^2 \times M^2}, (U_3 \otimes U_1) \in \mathbb{R}^{N^2 \times M^2}, (U_2 \otimes U_1) \in \mathbb{R}^{N^2 \times M^2}$ and $\mathcal{S} \in \mathbb{R}^{M \times M \times M}$ with learnable parameters $W_1^b \in \mathbb{R}^{N^2 \times M^2}, W_2^b \in \mathbb{R}^{N^2 \times M^2}, W_3^b \in \mathbb{R}^{N^2 \times M^2}$ and $\mathcal{T}^b \in \mathbb{R}^{M \times M \times M}$.

3. Replace $(U_3 \otimes U_2)\, S_{(1)} \in \mathbb{R}^{N^2 \times M}, (U_3 \otimes U_1)\, S_{(2)} \in \mathbb{R}^{N^2 \times M}$ and $(U_2 \otimes U_1)\, S_{(3)} \in \mathbb{R}^{N^2 \times M}$ with learnable parameters $W_1^c \in \mathbb{R}^{N^2 \times M}, W_2^c \in \mathbb{R}^{N^2 \times M}$ and $W_3^c \in \mathbb{R}^{N^2 \times M}$.

With all of the above three options, the E matrices are still computed with equation 2-23 using one of the above three substitutions. With this in mind, the "new" E matrix formulas

are defined as follows(4-10)

$$\begin{aligned}
&\text{option 1} && \text{option 2} && \text{option 3}\\
&E_1 = K_{(1)}\left(W_3^a \otimes W_2^a\right) T_{(1)}^a, && E_1 = K_{(1)} W_1^b T_{(1)}^b, && E_1 = K_{(1)} W_1^c,\\
&E_2 = K_{(2)}\left(W_3^a \otimes W_2^a\right) T_{(2)}^a, && E_2 = K_{(2)} W_2^b T_{(2)}^b, && E_2 = K_{(2)} W_2^c,\\
&E_3 = K_{(3)}\left(W_3^a \otimes W_2^a\right) T_{(3)}^a, && E_3 = K_{(3)} W_3^b T_{(3)}^b, && E_3 = K_{(3)} W_3^c.
\end{aligned} \tag{4-10}$$

In the non-KMLSVD 3D attention framework, the attention score tensor(3-3) and attention kernel tensor(3-6) are computed explicitly. These methods still follow the same initial steps: both compute the queries, keys and fibers(3-1), as well as the attention kernel. However, instead of then performing an MLSVD on the attention kernel tensor, the matrices E_1, E_2 and E_3 are obtained directly using equation 4-10. This approach therefore bypasses the need for any MLSVD related matrices or tensors.

These three non-KMLSVD 3D attention formulations(4-10) were all tested on the PemsSF and Japanese Vowels datasets and compared to standard self-attention using the same two transformer designs used in test 14-1-1.

The results of this test are shown in table 4-9. The non-KMLSVD attention framework has higher accuracy scores than some Primal KMLSVD attention formulations on the Japanese Vowels dataset. However, on the PemsSF dataset it performs worse than all Primal KMLSVD attention formulations accuracy wise (see table 4-8 and Appendix A-1). These results suggest that a non-KMLSVD 3D attention mechanism may indeed be feasible. Nevertheless, this conclusion cannot yet be drawn with confidence, as additional testing on the full set of ten datasets is required to ensure robustness, generality and a more fair comparison with the 3D Primal-Dual KMLSVD attention framework.

	Full layer								
	option 1			option 2			option 3		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Japanese Vowels	97.0	98.0	55.2	96.8	99.3	55.5	96.9	98.7	54.2
PEMS-SF	83.0	88.5	$1.06 \cdot 10^3$	83.0	90.0	$1.08{\cdot}10^3$	84.5	91.4	$1.06{\cdot}10^3$
	Last layer								
Japanese Vowels	96.5	98.3	43.6	96.9	98.3	43.9	96.5	97.6	45.2
PEMS-SF	83.7	89.2	286	83.0	89.2	292	83.8	89.9	283

Table 4-9: Comparison of mean and maximum validation accuracies and training time (in seconds) across datasets for different non-KMLSVD 3D self-attention definitions using two transformer variants: (i) replacing only the last self-attention layer with primal KMLSVD, and (ii) using primal KMLSVD for all layers.

Small Scale Test on a Textual Database

To complement the time series experiments, the IMDB dataset was included as an additional benchmark. Since natural language understanding relies heavily on capturing contextual dependencies, this dataset provides an additional (possibly stronger) test of whether Primal-Dual KMLSVD attention can truly operate as an alternative to standard self-attention mechanisms.

Note that a textual dataset like IMDB consists entirely of text and as such does not have a set

sequence length or a dimensional embedding. In this thesis, the embedding of text to numeric data is done in two steps. The first step is tokenization of the dataset, where a piece of text is "translated" to a numerical vector. This was done by using a pre-existing tokenization called BERT[14]. Then this vector is embedded into a higher dimension using a simple linear layer. Due to the sheer size of this dataset, a simplified transformer design, based on figure 4-2 was made in order to speed up the learning process. In this experiment, the Frobenius norm feature map, LLT design and MLSVD enforced $\mathcal{S}$ structure were used for Primal KMLSVD attention.

Model	**Accuracy (%)**
Canonical self-attention	71.4
Primal KMLSVD attention	69.8

Table 4-10: Attained accuracy of a single training iteration on the IMDB dataset for Canonical self-attention and Primal KMLSVD attention.

Due to time constraints, only a singular training run with fixed hyperparameter values was conducted. The recording of the training times for some reason failed and, due to the time constraint, could not be attempted again. Due to this, only the accuracy scores are shown in table 4-10. These accuracy scores are likely heavily influenced by random variation and should not be viewed as conclusive performance indicators. Nevertheless, the fact that Primal KMLSVD attention achieves reasonable accuracy on this textual dataset is strong evidence that primal KMLSVD attention can be used as a valid self-attention mechanism.

Chapter 5

Conclusion, further research and discussion

In this thesis, the primal-dual Kernel Multi-linear Singular Value Decomposition (KMLSVD) framework, as defined by Wesel and Batselier [3], is slightly modified and used to define 3D self-attention. This connection between the Primal-Dual (Kernel) Multi-linear Singular Value Decomposition (MLSVD) formulation allows for a novel definition of three-dimensional self-attention. Casting self-attention into a three-dimensional space could possibly allow for a more accurate or faster self-attention mechanism. Multiple Primal KMLSVD attention formulations were tested and compared to Primal attention[4] and canonical self-attention[24], with the best performing formulation being the $C_{\text{learn,vec}}$ formulation, with the Frobenius norm feature map(3-8), using the cost-enforced $\mathcal{S}$ strategy and the Last Layer Transformer (LLT) design. However, as not all cost-enforced $\mathcal{S}$ tests could be completed within the available time, the conclusion that cost-enforced is better than MLSVD enforced is somewhat limited in scope.

The main research question of this thesis is "Can 3D Primal-Dual KMLSVD attention actually function as a self-attention mechanism?". The results of all three tests combined (see section 4-2) demonstrate that both the Primal and Dual formulation can function as a self-attention mechanism. Although Dual KMLSVD attention is theoretically possible and attains reasonably accuracy scores, it is not practically feasible due to the excessive training time. The fact that Primal KMLSVD attention attains comparable and sometimes better accuracy results to canonical self-attention further reinforces the conclusion that 3D Primal KMLSVD attention works as a self-attention mechanism.

The first research sub-question is *"Can Primal-Dual KMLSVD attention be used as an improvement or alternative to self-attention?"*. The results of test 3 (see section 4-2-3) suggest that it cannot be used as an improvement or alternative. This is due to the fact that the best performing Primal KMLSVD attention formulation attains comparable or slightly better accuracy results compared to canonical self-attention at the cost of significantly increased training times. Due to these longer training times with little accuracy increase, the 3D Primal-Dual KMLSVD attention framework was not considered to be an improvement over canonical

self-attention.
The second research sub-question is *"Is the Primal or Dual KMLSVD attention formulation more favorable in terms of computational efficiency and predictive performance?"*. The results of test 1 and mainly test 2 (sections 4-2-1 and 4-2-2) clearly show that the Dual KMLSVD attention formulation is significantly slower than the Primal KMLSVD formulation. The Dual KMLSVD formulation is so slow that it is even considered as infeasible making the Primal formulation more favorable.
The third and final research sub-question is *"Can alternative primal KMLSVD attention formulations and feature maps improve training efficiency and predictive accuracy compared to the baseline approach?"*. The results of test 1 suggest a positive answer. Mainly, the formulation of making the $\mathcal{C}$ tensor learnable (either as a full tensor or a hyper-diagonal tensor) lead to increases in accuracy scores and sometimes even decreases in training time. The feature maps considered in this thesis did not seem to affect the attained accuracy scores overly much, but this could be a result of the limited scope of feature maps considered.
To completely write off the idea of using Primal-Dual KMLSVD attention may be too hasty. As of writing this thesis, only tests on inherently two-dimensional datasets have been attempted. This might explain why the results of Primal KMLSVD attention are not better than canonical self-attention because in hindsight, the Primal Dual KMLSVD framework is three-dimensional and capable of handling three-dimensional data. Applying a three-dimensional framework on inherently two-dimensional datasets may be redundant, as the additional dimension does not introduce new or functional information. Further research into applying Primal-Dual KMLSVD attention on three-dimensional, or higher order datasets, could lead to a more positive result. As of writing this thesis, the current Primal-Dual KMLSVD attention framework is geared towards two-dimensional datasets. A re-formulation or re-structuring of the feature maps specifically to accommodate multidimensional datasets could lead to significant improvements. Furthermore, the definition of a 3D self-attention mechanism that does not use the KMLSVD framework is also quite significant and warrants further research. A more in-depth investigation into different variations of Primal KMLSVD attention, the choice of feature maps within the mechanism, and the effect of the number of heads could provide valuable insights. Such research may lead to an improved Primal KMLSVD attention formulation, or at the very least, a clearer understanding of what does and does not work. Finally, the fact that primal-dual KMLSVD is not limited to 3D and can be expanded to higher dimensions could also be a source of new Primal-Dual KMLSVD attention definitions in higher dimensions (e.g., 4D, 5D etc.).

Appendix A

Appendix

In section A-1 the additional results of test 1 are shown. In section A-2, the results for test 2 are shown. Finally, in section A-3 the results of the additional test with cost enforced $\mathcal{S}$ Primal KMLSVD attention using the Frobenius nor feature map are shown.

A-1 Additional Test 1 results

In test 1 multiple feature maps, Primal KMLSVD attention variations, transformer designs and $\mathcal{S}$ enforcement strategies are tested. Due to the sheer number of results, only the "winning combination" was shown in section 4-2-1. In this section, all the other combinations that did not make the cut are shown along with the hyper parameter values η_1 and η_2 that lead to the results shown in section 4-2-1 and in this section. Note that, due to time constraints, the cost enforced $\mathcal{S}$ Primal KMLSVD attention tables are not completely filled in.

A-1-1 Frobenius norm feature map results

Cost enforced $\mathcal{S}$ structure using Frobenius norm feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	38.5	43.6	514	37.1	41.7	95.0	38.2	42.7	539	38.9	42.7	585
Face Detection	-	-	-	-	-	-	-	-	-	-	-	-
Handwriting	28.5	32.7	443	9.82	11.2	195	27.7	32.2	164	28.4	32.2	134
Heartbeat	74.8	77.4	666	75.7	78.1	359	74.9	77.4	315	74.8	77.4	328
Japanese Vowels	98.3	99.3	219	97.5	99.0	235	98.2	99.3	179	98.4	99.3	237
PEMS-SF	91.0	94.2	417	84.0	92.1	495	91.4	96.4	627	91.1	95.7	391
Self Regulation SCP1	91.8	93.5	851	90.5	93.2	661	91.7	93.9	410	91.7	93.9	223
Self Regulation SCP2	55.3	61.1	575	56.7	64.6	599	55.1	60.4	488	55.8	60.4	427
Spoken Arabic Digits	-	-	-	-	-	-	-	-	-	-	-	-
U Wave Gesture Library	86.9	88.7	127	79.7	84.4	122	86.4	89.1	94	86.8	89.1	122
Average	70.6	73.8	477	66.4	70.5	345	70.5	73.9	419	70.7	73.8	306
	Full layer											
Ethanol Concentration	36.0	38.9	330	37.4	40.8	176	36.0	38.4	441	36.2	38.9	591
Face Detection	-	-	-	-	-	-	-	-	-	-	-	-
Handwriting	29.7	32.1	$1.13{\cdot}10^3$	10.3	11.9	476	29.4	31.2	372	30.1	31.8	298
Heartbeat	75.2	77.4	$1.05{\cdot}10^3$	75.0	77.4	560	75.1	76.8	472	75.2	77.4	506
Japanese Vowels	98.3	99.7	561	96.7	98.3	591	97.8	98.7	415	98.2	99.0	549
PEMS-SF	91.2	95.7	694	81.8	86.3	786	90.9	95.0	967	91.3	95.0	589
Self Regulation SCP1	89.5	91.5	$1.26{\cdot}10^3$	90.5	92.8	998	90.2	91.5	535	89.3	91.1	284
Self Regulation SCP2	55.2	58.3	756	56.1	64.6	927	55.4	59.7	699	54.3	59.0	539
Spoken Arabic Digits	-	-	-	-	-	-	-	-	-	-	-	-
U Wave Gesture Library	86.4	88.3	327	80.9	84.4	311	86.5	88.7	213	86.5	88.3	273
Average	70.2	72.7	764	66.1	69.6	603	70.2	72.5	514	70.1	72.6	454

Table A-1: Comparison of mean accuracies, maximum accuracies, and average learning times (in seconds) across datasets for two transformer variants: (i) Last Layer Transformer (LLT), and (ii) Full Layer Transformer (FLT)

MLSVD enforced $\mathcal{S}$ structure using Frobenius norm feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	38.3	41.2	245	36.6	39.8	99	39.4	42.2	502	38.5	40.7	190
Face Detection	67.5	69.2	800	60.2	61.9	899	67.5	69.6	$1.33{\cdot}10^3$	67.6	68.7	667
Handwriting*	28.4	31.5	79.7	9.09	10.6	48.9	28.0	31.2	115	27.6	31.5	63.2
Heartbeat	74.6	76.2	43.4	75.2	78.1	81.9	74.7	76.8	108	74.8	77.4	26.2
Japanese Vowels	96.6	98.7	28.6	96.3	98.7	45.7	98.0	99.0	112	96.6	98.0	26.1
PEMS-SF-	80.6	84.9	24.7	76.6	82.7	27.8	91.0	93.5	196	81.2	85.6	32.4
Self Regulation SCP1	91.3	92.5	67.2	88.3	92.2	48.8	91.5	92.8	174	91.4	93.2	89.6
Self Regulation SCP2	54.8	57.6	88.4	55.6	62.5	50.1	55.0	60.4	222	55.3	58.3	79.6
Spoken Arabic Digits	98.1	98.7	452	94.6	95.1	740	98.4	98.8	$1.05{\cdot}10^3$	98.3	98.6	445
U Wave Gesture Library	86.3	87.5	14.2	77.3	81.6	22.0	86.2	89.1	66.2	86.5	89.1	13.5
Average	71.4	73.6	183	70.0	70.3	206	**73.0**	**75.3**	388	71.8	74.1	**163**
	Full layer											
Ethanol Concentration	34.9	36.5	135	36.2	38.9	174	35.1	37.0	228	34.4	37.0	77.0
Face Detection*	68.4	69.5	$1.95{\cdot}10^3$	60.0	61.2	$2.02{\cdot}10^3$	68.3	70.5	$3.30{\cdot}10^3$	68.2	69.8	$1.43{\cdot}10^3$
Handwriting	29.5	31.0	240	8.37	9.85	236	28.7	30.9	329	28.9	30.9	177
Heartbeat	75.1	76.8	102	75.0	76.8	142	75.2	77.4	226	75.2	78.1	53.9
Japanese Vowels	96.5	99.0	78.4	95.9	98.7	120	97.8	98.7	280	96.7	98.0	67.9
PEMS-SF	80.9	87.1	52.1	75.0	79.9	52.3	90.0	95.0	477	79.6	84.2	56.7
Self Regulation SCP1	89.1	90.1	63.3	89.4	92.8	88.6	89.8	91.1	163	88.8	90.1	81.1
Self Regulation SCP2	54.2	57.6	76.7	54.9	63.2	81.4	54.8	57.6	274	54.0	56.3	62.7
Spoken Arabic Digits	98.0	98.4	$1.05{\cdot}10^3$	95.1	95.7	$1.85{\cdot}10^3$	98.1	98.8	$2.82{\cdot}10^3$	98.0	98.3	$1.16{\cdot}10^3$
U Wave Gesture Library	85.8	87.5	35.0	74.8	78.9	57.1	86.0	87.9	168	86.0	87.5	31.2
Average	71.0	73.2	374	66.5	70.0	482	72.4	74.5	827	71.0	73.0	320

Table A-2: Comparison of mean accuracies, max accuracies, and average learning times (in seconds) across datasets for two transformer variants using the Frobenius norm feature map: (i) LLT, and (ii) FLT

A-1-2 Cosine similarity feature map results

The following two tables **ADD REF** show test 1 results using the Cosine similarity feature map.

Cost enforced $\mathcal{S}$ structure using cosine similarity feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	38.9	44.6	514	38.2	41.7	96.6	38.7	42.7	541	38.5	40.8	191
Face Detection	-	-	-	-	-	-	-	-	-	-	-	-
Handwriting	27.9	30.3	450	9.76	10.9	215	28.3	31.5	187	28.2	31.6	150
Heartbeat	74.9	76.8	657	75.0	78.1	356	75.2	77.4	268	74.9	76.8	322
Japanese Vowels	98.3	99.0	221	97.5	99.0	244	98.0	99.0	245	98.2	99.3	237
PEMS-SF	90.7	95.0	428	85.1	92.1	493	91.7	94.2	410	91.0	95.0	336
Self Regulation SCP1	91.8	93.5	853	90.3	92.2	696	91.9	93.5	413	91.7	93.9	209
Self Regulation SCP2	54.8	61.8	570	56.4	66.7	601	55.4	61.1	488	55.6	59.7	443
Spoken Arabic Digits	-	-	-	-	-	-	-	-	-	-	-	-
U Wave Gesture Library	86.3	88.3	127	80.3	82.8	305	86.3	88.3	101	86.7	88.3	125
Average	70.5	73.7	478	66.6	70.4	376	70.7	73.5	332	70.6	73.2	252
	Full layer											
Ethanol Concentration	36.1	39.8	332	39.3	42.7	179	36.6	40.3	448	34.4	37.0	77.9
Face Detection	-	-	-	-	-	-	-	-	-	-	-	-
Handwriting	29.9	31.9	$1.16{\cdot}10^3$	9.72	11.6	517	29.5	31.5	446	29.9	31.5	337
Heartbeat	75.0	77.4	$1.06{\cdot}10^3$	75.3	76.8	560	75.0	77.4	404	75.1	76.8	502
Japanese Vowels	98.1	99.3	567	96.2	98.0	615	97.8	99.0	580	98.3	99.7	551
PEMS-SF	90.3	93.5	719	83.6	90.0	806	91.0	96.4	640	90.4	93.5	526
Self Regulation SCP1	90.6	91.5	$1.27{\cdot}10^3$	89.6	91.5	$1.05{\cdot}10^3$	90.2	91.5	537	90.1	91.5	269
Self Regulation SCP2	54.8	59.0	873	56.5	63.2	926	54.5	60.4	685	54.4	60.4	546
Spoken Arabic Digits	-	-	-	-	-	-	-	-	-	-	-	-
U Wave Gesture Library	86.4	87.9	329	78.9	82.8	305	86.3	88.7	228	86.3	87.9	279
Average	70.2	72.5	789	66.1	69.6	620	70.1	73.2	496	69.9	72.3	386

Table A-3: Comparison of mean accuracies, maximum accuracies, and average learning times (in seconds) across datasets for two transformer variants using the cosine similarity feature map: (i) LLT, and (ii) FLT

MLSVD enforced $\mathcal{S}$ structure using cosine similarity feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	39.0	41.2	266	37.9	40.8	73.4	38.9	42.2	507	38.5	40.8	258
Face Detection	67.6	69.5	949	60.2	61.4	673	67.6	70.1	$1.15{\cdot}10^3$	67.7	69.5	716
Handwriting	27.6	31.2	36.8	8.71	10.6	35.1	28.2	31.5	101	27.3	31.5	42.8
Heartbeat	74.6	76.8	77.7	74.8	78.1	57.6	74.9	76.8	97.5	74.9	76.8	43.8
Japanese Vowels	98.0	99.0	48.6	96.4	98.0	44.5	97.3	98.3	89.2	97.7	98.7	42.2
PEMS-SF	88.6	92.1	90.8	83.0	87.8	84.2	90.7	92.8	194	88.8	92.1	66.6
Self Regulation SCP1	91.4	92.8	114	88.4	91.1	41.1	91.1	94.2	180	91.5	92.8	114
Self Regulation SCP2	55.8	59.0	140	56.5	61.1	53.6	55.1	57.6	219	55.4	59.7	118
Spoken Arabic Digits	98.2	98.6	713	94.6	95.6	687	98.2	98.7	$1.05{\cdot}10^3$	98.2	98.8	725
U Wave Gesture Library	85.7	85.7	24.3	75.9	82.4	20.1	86.6	88.7	70.0	86.0	87.5	21.3
Average	72.7	74.6	246	67.6	70.7	**177**	**72.9**	**75.1**	366	72.6	74.8	215
	Full layer											
Ethanol Concentration	35.4	41.2	266	38.0	39.8	131	35.4	37.4	243	35.5	37.0	122
Face Detection	68.7	70.1	$2.24{\cdot}10^3$	59.9	61.9	$1.60{\cdot}10^3$	67.9	69.3	$2.88{\cdot}10^3$	68.5	70.0	$1.73{\cdot}10^3$
Handwriting	28.6	30.3	97.4	8.53	9.35	89.7	28.4	30.6	286	28.9	30.4	115
Heartbeat	74.9	77.4	135	74.2	76.2	108	74.8	77.4	202	74.8	76.2	88.6
Japanese Vowels	98.1	99.0	132	94.9	97.0	117	97.7	98.7	253	97.8	99.0	111
PEMS-SF	89.4	95.7	193	79.6	84.2	168	89.6	92.1	485	89.7	92.1	136
Self Regulation SCP1	89.9	90.8	107	88.3	91.8	83.9	89.7	90.8	166	89.9	90.8	97.9
Self Regulation SCP2	54.7	59.0	125	55.4	61.1	112	53.8	58.3	271	55.4	59.7	118
Spoken Arabic Digits	98.0	98.5	$1.74{\cdot}10^3$	94.6	95.6	687	98.0	98.3	$2.85{\cdot}10^3$	98.0	98.6	$1.76{\cdot}10^3$
U Wave Gesture Library	85.7	87.1	61.8	75.0	81.3	52.5	85.5	86.7	178	85.8	89.1	52.8
Average	72.3	74.9	510	66.8	69.8	315	71.6	74.0	781	72.4	74.3	433

Table A-4: Comparison of mean accuracies, maximum accuracies, and average learning times (in seconds) across datasets for two transformer variants using the cosine similarity feature map: (i) LLT, and (ii) FLT

A-1-3 SM+ feature map results

Cost enforced $\mathcal{S}$ structure using SM+ feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	39.1	42.2	526	38.2	42.2	104	38.6	43.1	574	38.4	42.2	602
Face Detection	-	-	-	-	-	-	-	-	-	-	-	-
Handwriting	28.3	32.4	467	9.87	11.5	221	27.7	30.6	176	28.8	31.6	204
Heartbeat	75.0	78.7	675	75.6	78.7	363	74.9	76.8	334	75.1	76.8	198
Japanese Vowels	98.3	99.3	232	97.4	98.7	260	97.9	99.0	265	98.3	99.7	239
PEMS-SF	91.2	95.7	432	84.2	90.7	504	90.9	93.5	416	91.5	97.8	340
Self Regulation SCP1	91.9	93.9	880	90.3	93.2	695	92.1	93.9	401	91.8	93.2	202
Self Regulation SCP2	55.4	59.7	557	56.5	66.0	454	55.0	60.4	444	55.6	60.4	265
Spoken Arabic Digits	-	-	-	-	-	-	-	-	-	-	-	-
U Wave Gesture Library	86.5	88.3	138	79.2	83.6	126	86.8	88.3	110	86.9	88.7	146
Average	70.7	73.8	488	66.4	70.6	341	70.5	73.2	340	70.8	73.8	275
	Full layer											
Ethanol Concentration	35.6	37.4	331	35.1	37.9	206	36.0	37.9	503	35.6	38.4	678
Face Detection	-	-	-	-	-	-	-	-	-	-	-	-
Handwriting	29.7	31.8	$1.24{\cdot}10^3$	10.3	12.1	560	29.0	30.9	412	30.3	32.2	503
Heartbeat	75.2	78.1	$1.14{\cdot}10^3$	75.5	77.4	580	75.1	78.7	535	74.8	76.8	323
Japanese Vowels	98.2	99.3	610	96.9	99.0	672	97.7	99.7	646	98.3	99.3	577
PEMS-SF	91.4	95.0	745	82.9	88.5	836	90.9	95.7	665	91.5	96.4	543
Self Regulation SCP1	90.6	92.2	$1.36{\cdot}10^3$	87.5	92.2	$1.09{\cdot}10^3$	90.6	91.8	541	90.3	91.5	277
Self Regulation SCP2	54.5	59.0	738	56.4	63.9	708	55.4	57.6	567	54.7	58.3	277
Spoken Arabic Digits	-	-	-	-	-	-	-	-	-	-	-	-
U Wave Gesture Library	86.3	87.9	366	79.6	82.0	331	86.2	87.9	257	86.3	88.3	344
Average	70.2	72.6	663	65.5	69.1	623	70.1	72.5	516	70.2	72.7	440

Table A-5: Comparison of mean accuracies, maximum accuracies, and average learning times across datasets for two transformer variants using the SM+ feature map: (i) LLT, and (ii) FLT

MLSVD enforced $\mathcal{S}$ structure using SM+ feature map

	Last layer											
	Normal			$W_{n\|X}$			$\mathcal{C}_{\text{learn}}$			$\mathcal{C}_{\text{learn,diag}}$		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	38.1	41.2	365	36.7	40.3	63.7	38.6	41.2	561	38.8	42.7	373
Face Detection	67.8	68.8	$1.24{\cdot}10^3$	59.8	61.4	$1.36{\cdot}10^3$	68.1	69.1	$1.32{\cdot}10^3$	67.5	68.5	$1.16{\cdot}10^3$
Handwriting	26.9	30.0	103	8.79	1.15	72.5	27.4	30.4	116	27.4	30.4	92.5
Heartbeat*	74.7	76.8	83.8	74.8	77.4	66.3	74.5	76.2	97.6	74.7	76.2	73.8
Japanese Vowels	98.2	99.0	101	96.9	98.3	98.6	97.9	99.3	97.0	98.1	99.3	72.1
PEMS-SF	89.6	92.8	53.8	84.0	87.8	51.4	90.0	92.8	216	89.1	92.8	58.1
Self Regulation SCP1	91.6	92.8	147	88.5	91.8	50.6	91.8	93.5	208	91.3	92.8	111
Self Regulation SCP2	55.8	61.4	167	56.9	64.6	53.8	55.1	59.7	222	55.4	59.7	164
Spoken Arabic Digits	98.3	98.8	990	94.5	95.1	$1.44{\cdot}10^3$	98.1	98.5	$1.08{\cdot}10^3$	98.3	98.8	$1.12{\cdot}10^3$
U Wave Gesture Library	86.2	87.5	43.6	76.8	80.5	47.3	86.3	87.9	79.8	86.3	89.1	43.6
Average	72.7	74.9	329	67.8	69.8	**320**	**72.8**	74.9	400	72.7	**75.0**	327
	Full layer											
Ethanol Concentration	35.1	36.5	177	33.9	38.4	161	35.1	36.0	247	34.9	37.0	162
Face Detection	68.1	69.5	$3.35{\cdot}10^3$	60.5	62.2	$3.66{\cdot}10^3$	68.8	69.6	$3.38{\cdot}10^3$	68.1	69.5	$3.00{\cdot}10^3$
Handwriting	29.3	30.4	322	8.87	10.4	210	29.1	31.8	335	28.9	31.2	263
Heartbeat	74.7	76.8	205	74.6	76.2	190	74.8	77.4	206	74.8	76.2	159
Japanese Vowels	97.8	98.7	305	95.5	97.6	298	97.9	98.7	277	98.1	99.0	202
PEMS-SF	88.4	92.8	118	80.2	85.6	108	89.7	93.5	547	90.0	94.2	129
Self Regulation SCP1	89.7	90.4	155	86.8	90.8	140	90.7	91.5	372	89.6	90.8	95.0
Self Regulation SCP2	55.1	59.0	156	56.5	61.8	141	54.7	56.9	283	55.2	57.6	119
Spoken Arabic Digits	98.0	98.4	$2.61{\cdot}10^3$	95.4	96.0	$4.10{\cdot}10^3$	97.8	98.2	$2.92{\cdot}10^3$	98.1	98.5	$2.01{\cdot}10^3$
U Wave Gesture Library	86.2	87.5	117	74.6	78.9	134	85.5	87.5	199	85.9	87.5	106
Average	72.2	74.0	752	66.7	69.8	914	72.3	74.0	859	72.4	74.2	625

Table A-6: Comparison of mean accuracies, maximum accuracies, and average learning times (in seconds) across datasets for two transformer variants using the SM+ feature map: (i) LLT, and (ii) FLT

A-1-4 Best η_1 and η_2 KMLSVD Cost Hyper Parameters Per Dataset

Frobenius Norm Feature Map Hyper Parameters

$\mathcal{S}$ structure using Frobenius norm feature map

	Last Layer															
	Standard				$W_{n\|X}$				$\mathcal{C}_{\text{learn}}$				$\mathcal{C}_{\text{learn,diag}}$			
Dataset	Best	Mean	Best	Max	Best	Mean	Best	Max	Best	Mean	Best	Max	Best	Mean	Best	Max
	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2
Ethanol Concentration	[0	5]	[5	5]	[0	0.1]	[0.01	0.1]	[5	1]	[1	0.01]	[0.1	0.1]	[1	5]
Face Detection	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
Handwriting	[0	1]	[0.01	0.01]	[0	1]	[0	1]	[0	1]	[0.1	0.01]	[0.1	0.1]	[1	5]
Heartbeat	[1	0.01]	[0	0.01]	[0	0]	[0.01	5]	[-	-]	[-	-]	[0	0.01]	[0.1	5]
Japanese Vowels	[1	5]	[0	0.1]	[0	1]	[0	0.1]	[0.01	1]	[1	0.01]	[0.01	1]	[0.01	1]
PEMS-SF	[0	0.1]	[0.01	5]	[0	0.01]	[0.01	1]	[0.01	0.1]	[5	0]	[5	0.01]	[0	0.01]
Self Regulation SCP1	[1	0.1]	[0.1	1]	[0	0]	[0.1	0.1]	[0	1]	[0.01	0.1]	[0.01	0]	[1	0.01]
Self Regulation SCP2	[0	0.1]	[0.1	1]	[1	0.1]	[1	0.1]	[0.1	0.1]	[5	0.01]	[0	0]	[0	0]
Spoken Arabic Digits	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
U Wave Gesture Library	[0	0]	[0.1	5]	[0	0.1]	[0	5]	[0.1	0.01]	[0.01	0.01]	[0	5]	[0.1	0.1]
	Full Layer															
Ethanol Concentration	[0.01	1]	[0.01	5]	[0	0.01]	[0	5]	[0.1	0.01]	[0.01	0]	[1	1]	[0	5]
Face Detection	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
Handwriting	[0.01	5]	[0.1	0.1]	[0	1]	[0	1]	[0.01	1]	[0.01	1]	[0	0.1]	[0.1	5]
Heartbeat	[0	5]	[5	0.01]	[0	0.01]	[0	0]	[-	-]	[-	-]	[5	1]	[5	1]
Japanese Vowels	[0	0.01]	[0.1	0]	[0	0.1]	[0	0.1]	[0	0.01]	[0	0.01]	[0	5]	[0	1]
PEMS-SF	[1	5]	[5	0.1]	[0	0]	[0	5]	[0.01	0]	[0.1	1]	[1	0.1]	[5	1]
Self Regulation SCP1	[1	5]	[5	1]	[0	0.01]	[0	0]	[5	0.1]	[5	0.1]	[5	1]	[5	1]
Self Regulation SCP2	[1	1]	[0.01	0]	[0	0.01]	[1	0.01]	[0.01	1]	[0.01	1]	[0.01	0.1]	[0.01	0]
Spoken Arabic Digits	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
U Wave Gesture Library	[0.1	5]	[0.01	5]	[0	0.01]	[0	5]	[0	0.01]	[5	1]	[1	5]	[0	0]

Table A-7: η_1 and η_2 values leading to the highest (i) mean accuracy and (ii) max accuracy, reported for two transformer designs: (i) LLT, and (ii) FLT. Using cost-enforced $\mathcal{S}$ structure and Frobenius norm feature map.

	Last Layer							
	Standard		$W_{n\|X}$		$\mathcal{C}_{\text{learn}}$		$\mathcal{C}_{\text{learn,diag}}$	
Dataset	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max
Ethanol Concentration	0.01	1	0.01	0.1	0.1	0.1	1	0.01
Face Detection	0.01	1	0.1	0.01	0	0.01	5	5
Handwriting	0.01	5	0	0.01	0.01	0	5	5
Heartbeat	1	0.1	0.01	1	0.1	0.01	1	5
Japanese Vowels	5	0.1	0	0.01	0.01	0.01	0.01	1
PEMS-SF	5	5	0	0.01	1	0	0.01	0.01
Self Regulation SCP1	0.01	0.01	1	0.1	0	1	1	1
Self Regulation SCP2	1	0.01	1	5	0.01	5	1	1
Spoken Arabic Digits	0.1	5	1	0	1	0	1	1
U Wave Gesture Library	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0..01
	Full Layer							
Dataset	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max
Ethanol Concentration	0.1	0.1	0	0	0	0.01	1	0.1
Face Detection	0.1	1	0.1	0	0.1	5	5	5
Handwriting	0.01	0.1	0	0	0.01	0	5	1
Heartbeat	1	1	0	0	0	1	1	1
Japanese Vowels	1	0.1	0	0	0	0	1	0.01
PEMS-SF	1	1	0.01	0.01	1	0	5	5
Self Regulation SCP1	0.1	0.1	0	0	5	1	1	5
Self Regulation SCP2	0.1	0.01	5	1	0.1	5	5	0.1
Spoken Arabic Digits	5	1	0	0.01	1	5	0.1	5
U Wave Gesture Library	0.01	5	0	0	0.01	0.01	5	1

Table A-8: Best mean and max results for η_1 configurations across datasets for two transformer variants: (i) LLT, and (ii) FLT. Using MLSVD-enforced $\mathcal{S}$ structure and Frobenius norm feature map.

Cosine Feature Map Hyper Parameters

$\mathcal{S}$ structure using Cosine Similarity feature map

	Last Layer															
	Standard				$W_{n\|X}$				$\mathcal{C}_{\text{learn}}$				$\mathcal{C}_{\text{learn,diag}}$			
Dataset	Best		Mean		Best		Max		Best		Mean		Best		Max	
	η_1		η_2		η_1		η_2		η_1		η_2		η_1		η_2	
Ethanol Concentration	[0.01	1]	[0.01	1]	[0	0.1]	[0	5]	[0.01	0]	[1	0]	[5	5]	[1	0]
Face Detection	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
Handwriting	[0.01	0.01]	[0	0]	[0	0]	[0	1]	[0	1]	[0	1]	[0.01	0]	[0	0.01]
Heartbeat	[0.01	0]	[0	1]	[0	5]	[0	0.01]	[0.01	0]	[0.1	0]	[0.1	0]	[0	0.01]
Japanese Vowels	[0	1]	[0	0]	[0	0.01]	[0	1]	[0	0]	[0	0]	[0.01	0.01]	[0	5]
PEMS-SF	[0	0.01]	[0.01	0]	[0	1]	[0.01	1]	[0	0.01]	[0	0.01]	[0.01	0.01]	[0.1	5]
Self Regulation SCP1	[0.1	0]	[0.01	0]	[0	0]	[0	0.1]	[5	1]	[0	0]	[5	0.01]	[0	5]
Self Regulation SCP2	[0.01	1]	[0.01	1]	[0	0.01]	[0	0.1]	[5	0.1]	[1	0]	[5	5]	[5	5]
Spoken Arabic Digits	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
U Wave Gesture Library	[0	0.01]	[0.01	0]	[0	0]	[0	0]	[0	5]	[0	1]	[0	0.01]	[0	0.01]
	Full Layer															
Ethanol Concentration	[0	5]	[0	5]	[0	5]	[0	5]	[0	1]	[0	5]	[0	5]	[0	5]
Face Detection	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
Handwriting	[0	1]	[0	0.01]	[0	0.1]	[1	1]	[0	1]	[0	0.01]	[0	0]	[0.01	0]
Heartbeat	[1	0.1]	[0	0]	[0	0.01]	[0	0.1]	[0	0]	[0.1	1]	[0.1	0.1]	[0	]
Japanese Vowels	[0	0]	[0	5]	[0	5]	[0	5]	[0	5]	[0	0.01]	[0	0.01]	[0	0.01]
PEMS-SF	[0	5]	[0	0.01]	[0	0]	[0	0]	[0	0.01]	[0	0.01]	[0	5]	[0	0.01]
Self Regulation SCP1	[1	0.01]	[0.01	5]	[0	0.01]	[0	1]	[0.01	0]	[0.01	0]	[1	0.1]	[0.1	0.01]
Self Regulation SCP2	[0.1	0.1]	[1	5]	[0.01	0]	[0	0]	[1	1]	[0.01	5]	[0	0.1]	[5	0.1]
Spoken Arabic Digits	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
U Wave Gesture Library	[0	1]	[0	1]	[0	5]	[0	5]	[0	0.1]	[0	0.01]	[0	0.1]	[0	0.1]

Table A-9: η_1 and η_2 values leading to the highest (i) mean accuracy and (ii) max accuracy, reported for two transformer designs: (i) LLT, and (ii) FLT. Using cost-enforced $\mathcal{S}$ structure and Cosine norm feature map.

	Last Layer							
	Standard		$W_{n\|X}$		$\mathcal{C}_{\text{learn}}$		$\mathcal{C}_{\text{learn,diag}}$	
Dataset	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max
Ethanol Concentration	1	0	0	0	1	0	0.01	1
Face Detection	5	0.1	5	1	0.1	0	0.1	0.1
Handwriting	0.01	0.01	0	0	0	0	1	1
Heartbeat	0.1	0	0	0	5	0	0.1	0.01
Japanese Vowels	0.01	0.01	0	0	0	0	0.1	0
PEMS-SF	0	0	0	0	0	0	0	0.1
Self Regulation SCP1	0	5	0	0.01	0.01	5	5	5
Self Regulation SCP2	1	0	0	5	1	0.1	5	0
Spoken Arabic Digits	0	0	0.1	0.01	0	0.01	0.1	0.1
U Wave Gesture Library	0	0.01	0	0	0	0.01	0.01	0.01
	Full Layer							
Dataset	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max
Ethanol Concentration	0	0	0	0	0	0.1	0	0.01
Face Detection	0.1	0.1	0.01	0.01	0	0	1	1
Handwriting	0	0	0	0	0	0	0	0.1
Heartbeat	0	0.1	0	0	5	0	5	1
Japanese Vowels	0	0	0	0	0	0	0.01	0
PEMS-SF	0	0	0	0	0	0	0	0
Self Regulation SCP1	1	0.1	0	0	5	0.01	5	1
Self Regulation SCP2	0	0	0	0.1	0.1	0.1	1	0
Spoken Arabic Digits	0.01	5	0	0	0	0.1	0.01	0.01
U Wave Gesture Library	5	1	0	0.1	0	0	0	0

Table A-10: Best mean and max results for η configurations across datasets for two transformer variants: (i) LLT, and (ii) FLT. UUsing MLSVD-enforced $\mathcal{S}$ structure and Cosine norm feature map.

SM+ Feature Map Hyper Parameters

$\mathcal{S}$ structure using Frobenius norm feature map

	Last Layer															
	Standard				$W_{n\|X}$				$\mathcal{C}_{\text{learn}}$				$\mathcal{C}_{\text{learn,diag}}$			
Dataset	Best		Mean		Best		Max		Best		Mean		Best		Max	
	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2	η_1	η_2
Ethanol Concentration	[0.1	1]	[0.1	0.1]	[0	5]	[0.1	0.1]	[0.01	0]	[0	1]	[5	0]	[0.01	5]
Face Detection	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
Handwriting	[0.1	0]	[0	0]	[0	0.01]	[0	0.01]	[0	0]	[0	5]	[1	0.1]	[0.01	1]
Heartbeat	[0	0.1]	[0	0.1]	[0	0.1]	[0	0.1]	[0.01	1]	[0	1]	[0.1	0.01]	[0	0]
Japanese Vowels	[0	0.1]	[0	0.1]	[0	0]	[0	0]	[0	0]	[0.01	0.01]	[5	5]	[0	0.1]
PEMS-SF	[0.1	0.1]	[0.1	0.1]	[0	5]	[0	0.01]	[0	5]	[0	1]	[1	0]	[1	0.1]
Self Regulation SCP1	[5	1]	[0.01	0.1]	[0	1]	[0	0]	[0	1]	[0.01	0.01]	[0.1	0.01]	[0	0.01]
Self Regulation SCP2	[0.01	0]	[0.1	5]	[0.01	0]	[0.1	5]	[1	1]	[1	0]	[0.01	5]	[0.01	0]
Spoken Arabic Digits	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
U Wave Gesture Library	[0.01	5]	[0.01	0.1]	[0	1]	[0.01	0.1]	[0	0.1]	[0	0.1]	[0	5]	[0.01	0]
	Full Layer															
Ethanol Concentration	[0	0.1]	[0	0.1]	[0	0.1]	[0	0.1]	[0	0]	[0	1]	[0	5]	[0	0]
Face Detection	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
Handwriting	[0	0]	[0	1]	[0	5]	[0	0.01]	[0	5]	[1	0.01]	[0	5]	[0	5]
Heartbeat	[0.01	0]	[0	0]	[0	1]	[0	1]	[5	1]	[0	1]	[0.01	0]	[5	0.01]
Japanese Vowels	[0	0]	[0	1]	[0	1]	[0	0]	[0	0.1]	[0	0.1]	[0	5]	[0	0]
PEMS-SF	[0.01	0.01]	[0	0]	[0	5]	[0	5]	[0	0]	[0	5]	[0.1	1]	[0.1	1]
Self Regulation SCP1	[0.1	0.1]	[0.1	1]	[0.01	0.1]	[0.1	0.01]	[0.1	1]	[5	0]	[1	1]	[0.01	0]
Self Regulation SCP2	[5	0.01]	[0	5]	[0	0.01]	[1	5]	[0.1	0.01]	[0.1	0.1]	[0.1	5]	[0.01	1]
Spoken Arabic Digits	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]	[-	-]
U Wave Gesture Library	[0	5]	[0	0.1]	[0	0.01]	[0	0]	[0	0.1]	[0	1]	[0.01	0.01]	[0.1	0.1]

Table A-11: η_1 and η_2 values leading to the highest (i) mean accuracy and (ii) max accuracy, reported for two transformer designs: (i) LLT, and (ii) FLT. Using cost-enforced $\mathcal{S}$ structure and SM+ norm feature map.

	Last Layer							
	Standard		$W_{n\|X}$		$\mathcal{C}_{\text{learn}}$		$\mathcal{C}_{\text{learn,diag}}$	
Dataset	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max
Ethanol Concentration	0.01	1	0.01	0	0	5	5	5
Face Detection	1	5	1	1	0.1	0.01	1	0
Handwriting	0	0	0.01	0.01	0	0	5	1
Heartbeat	1	0.01	0.01	0	1	0.01	5	0.01
Japanese Vowels	0.01	0	0	0	0	0	0.1	0
PEMS-SF	0	0	0.01	0.01	0	0	0.01	0.01
Self Regulation SCP1	0.1	0.1	0	0	0.01	0	0.1	1
Self Regulation SCP2	0.01	0.01	0.01	5	0	0	1	0.01
Spoken Arabic Digits	0	5	5	0.01	0	0	1	1
U Wave Gesture Library	0	0.1	0	0	0	0	0.01	0.01
	Full Layer							
Dataset	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max	Best Mean	Best Max
Ethanol Concentration	0	0	0	0.1	0	0	0.1	0.01
Face Detection	0.01	5	0.01	0	0.1	0.1	0	5
Handwriting	0.1	0.01	0	0	0	0	1	1
Heartbeat	0	0	0	0	1	1	0.01	0
Japanese Vowels	0	0	0	0	0	0	0.01	0
PEMS-SF	0	0.1	0	0	0	0	0.1	0.01
Self Regulation SCP1	0.1	0.1	1	0.1	0.1	0.01	5	5
Self Regulation SCP2	0.1	5	0	0	0.01	0.01	1	1
Spoken Arabic Digits	0	0.01	0	0	0	0	1	1
U Wave Gesture Library	0	0	0	0	0	0	0	5

Table A-12: Best mean and max results for η configurations across datasets for two transformer variants: (i) LLT, and (ii) FLT. Using MLSVD-enforced $\mathcal{S}$ structure and SM+ norm feature map.

A-2 Additional Test 2 Results

Below is the table corresponding to the results of test 2, Dual KMLSVD attention applied on ten timeseries datasets. Due to time constraints and the significantly long training times associated with Dual KMLSVD attention, not all ten datasets could be fully finished in time.

MLSVD enforced $\mathcal{S}$ structure using Frobenius norm feature map (Single Variant)

	Attention Variant					
	Last layer			Full layer		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	34.9	41.7	$1.35{\cdot}10^4$	30.9	38.9	$4.26{\cdot}10^4$
Handwriting	24.8	26.9	$1.78{\cdot}10^4$	27.9	30.0	$6.89{\cdot}10^4$
Heartbeat	73.4	76.2	$9.06{\cdot}10^3$	73.9	75.6	$3.71{\cdot}10^4$
Japanese Vowels	97.6	98.6	$5.18{\cdot}10^3$	97.6	98.6	$2.10{\cdot}10^4$
PEMS-SF	85.9	92.8	$1.21{\cdot}10^4$	86.4	90.6	$4.83{\cdot}10^4$
Self Regulation SCP1	90.3	93.6	$1.39{\cdot}10^4$	86.6	88.9	$4.57{\cdot}10^4$
Self Regulation SCP2	50.8	60.4	$8.66{\cdot}10^3$	52.8	61.1	$2.88{\cdot}10^4$
U Wave Gesture Library	85.9	87.5	$1.07{\cdot}10^4$	85.7	87.1	$3.60{\cdot}10^4$
Average	68.0	72.2	$1.14{\cdot}10^4$	67.7	71.4	$4.11{\cdot}10^4$

Table A-13: Comparison of mean accuracies, max accuracies, and average learning times (in seconds) across finished datasets using Dual KMLSVD attention, for both LLT and FLT.

A-3 Additional Test 3 Results

The table below shows the additional Test 3 done using the i) $\mathcal{C}_{\text{learn,diag}}$ Primal KMLSVD attention with ii) cost enforced $\mathcal{S}$ strategy using the iii) Frobenius norm feature map 3-8.

Comparison of Attention Mechanisms using Frobenius Norm Feature Map

	Learnable Parameter Budget								
	Primal KMLSVD attention			Primal Attention			Canonical self attention		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	31.2	44.2	$1.02{\cdot}10^3$	33.5	42.3	871	34.8	44.2	859
Face Detection	74.0	75.9	$4.70{\cdot}10^3$	72.8	74.9	$3.40{\cdot}10^3$	74.9	76.9	$2.70{\cdot}10^3$
Handwriting	54.4	58.2	889	55.7	59.4	539	56.2	60.6	391
Heartbeat	76.3	82.9	$1.04{\cdot}10^3$	74.2	90.2	827	70.5	80.5	742
Japanese Vowels	95.5	97.3	629	95.7	98.7	370	95.1	96.0	269
PEMS-SF	90.9	94.1	741	92.1	94.1	594	90.0	94.1	522
Self Regulation SCP1	92.9	96.6	819	91.6	94.8	667	91.4	93.1	583
Self Regulation SCP2	51.9	58.3	116	51.7	61.1	109	51.9	55.6	113
Spoken Arabic Digits	99.1	99.6	$2.31{\cdot}10^3$	99.2	99.5	$1.27{\cdot}10^3$	99.2	99.3	842
U Wave Gesture Library	94.8	98.4	334	95.3	98.4	195	95.2	100	133
Average	76.1	80.1	$1.26{\cdot}10^3$	**76.2**	**81.3**	884	75.9	80.0	**715**
	Equal Dimensionality								
	Primal KMLSVD attention			Primal Attention			Canonical self attention		
Dataset	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)	Mean (%)	Max (%)	Time (s)
Ethanol Concentration	27.7	38.5	750	29.4	34.6	647	30.0	34.6	633
Face Detection	66.7	69.0	$3.78{\cdot}10^3$	66.3	68.3	$3.05{\cdot}10^3$	66.3	68.6	$2.46{\cdot}10^3$
Handwriting	24.7	29.4	209	25.0	28.2	147	26.5	28.8	115
Heartbeat	81.7	85.4	445	81.5	85.4	367	82.0	87.8	335
Japanese Vowels	94.5	97.3	336	94.7	97.3	210	95.0	98.7	156
PEMS-SF	85.0	94.1	537	86.2	91.2	438	84.4	88.2	389
Self Regulation SCP1	91.7	94.8	636	89.8	91.4	525	90.2	93.1	487
Self Regulation SCP2	48.3	58.3	483	46.9	58.3	404	47.8	55.6	380
Spoken Arabic Digits	98.1	98.6	$2.40{\cdot}10^3$	98.1	98.9	$1.43{\cdot}10^3$	98.3	98.9	884
U Wave Gesture Library	85.3	89.1	187	85.2	87.5	113	85.8	89.1	87.4
Average	70.4	**75.5**	976	70.3	74.1	733	**70.6**	74.3	**593**

Table A-14: Comparison of mean validation accuracy, max validation accuracy, and training time across datasets for three attention mechanisms (canonical self-attention, primal attention, and primal KMLSVD attention) under two settings: learnable parameter budget and equal dimensionality, using the Frobenius norm feature map.

Bibliography

[1] AIAAIC. "chatgpt training emits 502 metric tons of carbon". https://www.aiaaic.org/aiaaic-repository/ai-algorithmic-and-automation-incidents/chatgpt-training-emits-502-metric-tons-of-carbon, 2022. Accessed: 2025-11-10.

[2] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31:606–660, 2017.

[3] K. Batselier and F.Wesel. A kernelizable primal-dual formulation of the multilinear singular value decomposition.

[4] Y. Chen, Q.Tao, F.Tonin, and J.A.K. Suykens. Primal-attention: Self-attention through asymmetric kernel svd in primal representation, 2023.

[5] Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier ai models, 2025.

[6] C.Zetai and L.Clifford. Tensor time series imputation through tensor factor modelling, 2024.

[7] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[8] D.Gong and H.Zhang. Self-attention limits working memory capacity of transformer-based models, 2024.

[9] G. H. Golub. and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1989.

[10] M. He, F. He, L. Shi, X. Huang, and J. A. K. Suykens. Learning with asymmetric kernels: Least squares and feature interpretation, 2022.

[11] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *International 1989 Joint Conference on Neural Networks*, pages 593–605 vol.1, 1989.

[12] H.Saratchandran, J.Zheng, Y.Ji, W.Zhang, and S.Lucey. Rethinking attention: Polynomial alternatives to softmax in transformers, 2025.

[13] J.Cordonnier, A.Loukas, and M.Jaggi. Multi-head attention: Collaborate instead of concatenate, 2021.

[14] J.Devlin, M.Chang, K.Lee, and K.Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[15] J.J.Tithi, H.Wu, A.Abuhatzera, and F.Petrini. Scaling intelligence: Designing data centers for next-gen language models, 2025.

[16] K.Choromanski, V.Likhosherstov, D.Dohan, X.Song, A.Gane, T.Sarlos, P.Hawkins, J.Davis, A.Mohiuddin, L.Kaiser, D.Belanger, L.Colwell, and A.Weller. Rethinking attention with performers, 2022.

[17] M. Khojaste-Sarakhsi, S. M. T. Fatemi Ghomi S. S. Haghighi, and E. Marchiori. Deep learning for alzheimer's disease diagnosis: A survey. *Artificial Intelligence in Medicine*, 130:102332, 2022.

[18] F.Duman K.Maheskaya, P.Wijewardena, and C.Hegde. On the computational complexity of self-attention, 2022.

[19] L. Lathauwer and B. De Moor. A multi-linear singular value decomposition. *Society for Industrial and Applied Mathematics*, 21:1253–1278, Mar. 2000.

[20] N.Jegham, M.Abdelatti, L.Elmoubarki, and A.Hendawi. How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference, 2025.

[21] R.Azad, L.Niggemeier, M.Huttemann, A.Kazerouni, E.K.Aghdam, Y.Velichko, U.Bagci, and D.Merhof. Beyond self-attention: Deformable large kernel attention for medical image segmentation, 2023.

[22] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

[23] J.A.K. Suykens. Svd revisited: A new variational principle, compatible feature maps and nonlinear extensions. *Applied and Computational Harmonic Analysis*, 40(3):600–609, 2016.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[25] W.Vanderbauwhede. Estimating the increase in emissions caused by ai-augmented search, 2025.

[26] Y.Liu, T.Han, S.Ma, J.Zhang, Y.Yang, J.Tian, H.He, A.Li, M.He, Z.Liu, Z.Wu, L.Zhao, D.Zhu, X.Li, N.Qiang, D.Shen, T.Liu, and B.Ge. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017, September 2023.

[27] M. Yu and M. Fujita. Parallel scheduling self-attention mechanism: Generalization and optimization, 2020.

[28] H. Zhao, J. Jia, and V. Koltun. Exploring self-attention for image recognition, 2020.

Glossary

List of Acronyms

DCSC	Delft Center for Systems and Control
ED	Equal Dimensionality
FLT	Full Layer Transformer
GPU	Graphics Processing Unit (video card)
KMLSVD	Kernel Multi-linear Singular Value Decomposition
KSVD	Kernel Singular Value Decomposition
MLSVD	Multi-linear Singular Value Decomposition
LLM	Large Language Models
LLT	Last Layer Transformer
LPB	Learnable Parameter Budget
SVD	Singular Value Decomposition
VRAM	Video Random Access Memory