A PRIMAL-DUAL FRAMEWORK FOR TRANSFORMERS AND NEURAL NETWORKS

Tan M. Nguyen*
Department of Mathematics
University of California, Los Angeles
tanmnguyen89@ucla.edu

Tam Nguyen*
Department of ECE
Rice University
nguyenminhtam9520@gmail.com

Nhat Ho
Department of Statistics & Data Sciences
University of Texas at Austin
minhnhat@utexas.edu

Andrea L. Bertozzi
Department of Mathematics
University of California, Los Angeles
bertozzi@math.ucla.edu

Richard G. Baraniuk
Department of ECE
Rice University
richb@rice.edu

Stanley J. Osher
Department of Mathematics
University of California, Los Angeles
sjo@math.ucla.edu

ABSTRACT

Self-attention is key to the remarkable success of transformers in sequence modeling tasks including many applications in natural language processing and computer vision. Like neural network layers, these attention mechanisms are often developed by heuristics and experience. To provide a principled framework for constructing attention layers in transformers, we show that the self-attention corresponds to the support vector expansion derived from a support vector regression problem, whose primal formulation has the form of a neural network layer. Using our framework, we derive popular attention layers used in practice and propose two new attentions: 1) the Batch Normalized Attention (Attention-BN) derived from the batch normalization layer and 2) the Attention with Scaled Head (Attention-SH) derived from using less training data to fit the SVR model. We empirically demonstrate the advantages of the Attention-BN and Attention-SH in reducing head redundancy, increasing the model's accuracy, and improving the model's efficiency in a variety of practical applications including image and time-series classification.

1 INTRODUCTION

Transformer models (Vaswani et al., 2017) have achieved impressive success with state-of-the-art performance in a myriad of sequence processing tasks, including those in computer vision (Dosovitskiy et al., 2021; Liu et al., 2021; Touvron et al., 2020; Ramesh et al., 2021; Radford et al., 2021; Arnab et al., 2021; Liu et al., 2022; Zhao et al., 2021; Guo et al., 2021), natural language processing (Devlin et al., 2018; Al-Rfou et al., 2019; Dai et al., 2019; Child et al., 2019; Raffel et al., 2020; Baevski & Auli, 2019; Brown et al., 2020; Dehghani et al., 2018), reinforcement learning (Chen et al., 2021; Janner et al., 2021), and other important applications (Rives et al., 2021; Jumper et al., 2021; Zhang et al., 2019; Gulati et al., 2020; Wang & Sun, 2022). Transformers can also effectively transfer knowledge from pre-trained models to new tasks with limited supervision (Radford et al., 2018; 2019; Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019). The driving force behind the success of transformers is the self-attention mechanism (Cho et al., 2014; Parikh et al., 2016; Lin et al., 2017), which computes a weighted average of feature representations of the tokens in the sequence with the weights proportional to similarity scores between pairs of representations. The weights calculated by the self-attention determine the relative importance between tokens and thus capture the contextual representations of the sequence (Bahdanau et al., 2014; Vaswani et al., 2017; Kim et al., 2017). It has

* Co-first authors. Please correspond to: tanmnguyen89@ucla.edu

been argued that the flexibility in capturing diverse syntactic and semantic relationships is critical for the success of transformers (Tenney et al., 2019; Vig & Belinkov, 2019; Clark et al., 2019; Voita et al., 2019a; Hewitt & Liang, 2019).

1.1 Background: Self-Attention

For a given input sequence $\mathbf{X} := [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times D_x}$ of N feature vectors, self-attention transforms $\mathbf{X}$ into the output sequence $\mathbf{H}$ in the following two steps:

Step 1. The input sequence $\mathbf{X}$ is projected into the query matrix $\mathbf{Q}$, the key matrix $\mathbf{K}$, and the value matrix $\mathbf{V}$ via three linear transformations

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q^\top; \mathbf{K} = \mathbf{X}\mathbf{W}_K^\top; \mathbf{V} = \mathbf{X}\mathbf{W}_V^\top,$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{D\times D_x}$, and $\mathbf{W}_V \in \mathbb{R}^{D_v \times D_x}$ are the weight matrices. We denote $\boldsymbol{Q} := [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N]^\top, \mathbf{K} := [\boldsymbol{k}_1, \cdots, \boldsymbol{k}_N]^\top$, and $\mathbf{V} := [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]^\top$, where the vectors $\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i$ for $i = 1, \cdots, N$ are the query, key, and value vectors, respectively.

Step 2. The output sequence $\mathbf{H} := [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_N]^\top$ is then computed as follows

$$\mathbf{H} = \mathrm{softmax}\Big(\mathbf{Q}\mathbf{K}^\top/\sqrt{D}\Big)\mathbf{V} := \mathbf{A}\mathbf{V}, \tag{1}$$

where the softmax function is applied to each row of the matrix $\mathbf{Q}\mathbf{K}^\top/\sqrt{D}$. The matrix $\mathbf{A} := \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}}\Big) \in \mathbb{R}^{N\times N}$ and its component a_{ij} for $i, j = 1, \cdots, N$ are called the attention matrix and attention scores, respectively. For each query vector $\boldsymbol{q}_i$ for $i = 1, \cdots, N$, an equivalent form of Eqn. (1) to compute the output vector $\boldsymbol{h}_i$ is given by

$$\boldsymbol{h}_i = \sum_{j=1}^{N} \mathrm{softmax}\Big(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{D}\Big)\boldsymbol{v}_j. \tag{2}$$

The self-attention computed by Eqn. (1) and (2) is called the scaled dot-product or softmax attention. In our paper, we call a transformer that uses this attention the softmax transformer. The structure that the attention matrix $\mathbf{A}$ learns from training determines the ability of the self-attention to capture contextual representation for each token. Additionally, a residual connection can be added to the output of the self-attention layer, $\boldsymbol{h}_i = \boldsymbol{x}_i + \sum_{j=1}^{N} \mathrm{softmax}\Big(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{D}\Big)\boldsymbol{v}_j$.

Multi-head Attention (MHA). In MHA, multiple heads are concatenated to compute the final output. This MHA mechanism allows transformers to capture more diverse attention patterns and increase the capacity of the model. Let H be the number of heads and $\mathbf{W}_O^{\text{multi}} = \big[\mathbf{W}_O^1, \dots, \mathbf{W}_O^H\big] \in \mathbb{R}^{D_v \times HD_v}$ be the projection matrix for the output where $\mathbf{W}_O^1, \dots, \mathbf{W}_O^H \in \mathbb{R}^{D_v\times D_v}$. The MHA is defined as

$$\mathrm{MultiHead}(\{\mathbf{H}\}_{s=1}^H) = \mathrm{Concat}(\mathbf{H}^1, \dots, \mathbf{H}^H)\mathbf{W}_O^{\text{multi}\top} = \sum_{s=1}^{H} \mathbf{H}^s\mathbf{W}_O^{s\top} = \sum_{s=1}^{H} \mathbf{A}^s\mathbf{V}^s\mathbf{W}_O^{s\top}. \tag{3}$$

Despite their remarkable success, most attention layers are developed based on intuitions and heuristic approaches. Beyond these intuitions, a coherent principled framework for synthesizing attention layers has remained elusive.

1.2 Contribution

We derive the self-attention as the support vector expansion of a given support vector regression (SVR) problem. The primal representation of the regression function has the form of a neural network layer. Thus, we establish a primal-dual connection between an attention layer in transformers and a neural network layer in deep neural networks. Our framework suggests a principled approach to developing an attention mechanism: *Starting from a neural network layer and a support vector regression problem, we derive the dual as a support vector expansion to attain the corresponding attention layer.* We then employ this principled approach to invent two novel classes of attentions: the Batch Normalized Attention (Attention-BN) derived from the batch normalization layer in deep neural networks and the Attention with Scaled Heads (Attention-SH) resulting from solving the support vector regression model with less amount of training data. Our contribution is three-fold.

1. We derive self-attention as a support vector expansion that solves a SVR problem, thus providing a principled primal-dual framework to study and develop self-attentions.

2. We re-derive popular attentions, such as the linear attention (Katharopoulos et al., 2020), the sparse attention (Child et al., 2019), and the multi-head attention (Vaswani et al., 2017), from our proposed framework.

3. We develop two new attention mechanism: the Batch Normalized Attention (Attention-BN) and the Attention with Scaled Heads (Attention-SH), as well as their linear versions, using our proposed framework.

We empirically demonstrate that 1) the Attention-BN significantly outperforms the baseline softmax and linear attention and 2) the Attention-SH performs better while being more efficient than the same baselines on a variety of practical tasks including image and time-series classification.

Organization. We structure this paper as follows: In Section 2, we present the primal-dual framework to derive self-attention from solving a SVR problem. Also in this section, we derive popular attentions and develop new ones based on our framework. We empirically validate and analyze the advantages of our new attentions in Sections 3 and 4. We discuss related works in Section 5. The paper ends with concluding remarks in Section 6. Technical proofs, additional derivations of other attention mechanisms, and more experimental details are provided in the Appendix.

2 Primal-Dual Interpretation of Self-Attention

We first provide a primal-dual interpretation of self-attention as a support vector regression problem in Section 2.1. Based on that primal-dual framework, we derive popular attention mechanisms as the support vector expansion in Section 2.2. Finally, we introduce two new attention mechanisms in Section 2.3, the Attention-BN and Attention-SH.

2.1 Attention as a Support Vector Regression Model

In this section, we derive self-attention from a support vector regression problem. Suppose we are given a training data $\{(\boldsymbol{k}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{k}_N, \boldsymbol{y}_N)\} \subset \mathcal{K} \times \mathcal{Y}$, where $\mathcal{K} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}^{D_v}$. Here, $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N$ are attention keys in self-attention, and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are the training targets. We consider the function f, taking the form

$$\boldsymbol{y} = f(\boldsymbol{x}) := \mathbf{W}\frac{\Phi(\boldsymbol{x})}{h(\boldsymbol{x})} + \boldsymbol{b}, \tag{4}$$

where $\boldsymbol{x} \in \mathcal{K} = \mathbb{R}^D$, $\Phi(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \ldots, \phi_{D_\phi}(\boldsymbol{x})] \in \mathbb{R}^{D_\phi}$, $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{D_v}]^\top \in \mathbb{R}^{D_v \times D_\phi}$, $\boldsymbol{b} \in \mathbb{R}^{D_v}$, and $h(\boldsymbol{x})$ is a vector-scalar function. We fit the function f to the training data $\{(\boldsymbol{k}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{k}_N, \boldsymbol{y}_N)\}$ with an $\mathbb{L}_2$ regularization on $\mathbf{W}$, i.e., a ridge regression, by solving the following convex optimization problem:

$$\begin{aligned}
&\underset{\substack{\mathbf{W} \\ \boldsymbol{\xi}_j, \tilde{\boldsymbol{\xi}}_j, j=1,\ldots,N}}{\text{minimize}} && \frac{1}{2}\|\mathbf{W}\|_{\mathrm{F}}^2 + C\sum_{j=1}^{N}\sum_{d=1}^{D_v}\left(\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\xi}}_j(d)\right) = \frac{1}{2}\sum_{d=1}^{D_v}\|\boldsymbol{w}_d\|^2 + C\sum_{j=1}^{N}\sum_{d=1}^{D_v}\left(\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\xi}}_j(d)\right) \\
&\text{subject to} && \begin{cases} \boldsymbol{y}_j(d) - \boldsymbol{w}_d^\top \Phi(\boldsymbol{k}_j)/h(\boldsymbol{k}_j) - \boldsymbol{b}(d) \leq \epsilon + \boldsymbol{\xi}_j(d) \\ \boldsymbol{w}_d^\top \Phi(\boldsymbol{k}_j)/h(\boldsymbol{k}_j) + \boldsymbol{b}(d) - \boldsymbol{y}_j(d) \leq \epsilon + \tilde{\boldsymbol{\xi}}_j(d) \\ \boldsymbol{\xi}_j(d), \tilde{\boldsymbol{\xi}}_j(d) \geq 0 \end{cases}, \; j = 1, \ldots, N, \; d = 1, \ldots, D_v.
\end{aligned} \tag{5}$$

The Eqn. 5 implies that there exists a function f that can approximates all pairs $(\boldsymbol{k}_j, \boldsymbol{y}_j)$ with ϵ precision. The additional slack variables $\boldsymbol{\xi}_j, \tilde{\boldsymbol{\xi}}_j$ relax this assumption and allows some of the training set data points to have the training error greater than ϵ just as in the soft-margin SVM (Cortes & Vapnik, 1995; Schölkopf et al., 2002). $C > 0$ is a constant determining the trade between the complexity penalizer $\sum_{d=1}^{D_v} \|\boldsymbol{w}_d\|^2$, i.e., the flatness of f, and the amount up to which deviations larger than ϵ are tolerated.

In order to derive the self-attention from the support vector regression defined by the optimization problem 5, the key idea to construct the Lagrangian from Eqn. 5 and find the representation of the

$\boldsymbol{w}_d$, $d = 1, \ldots, D_v$, in terms of the dual variables. We define the Lagrangian function as follows:

$$\begin{aligned}\mathcal{L} &:= \frac{1}{2}\sum_{d=1}^{D_v}\|\boldsymbol{w}_d\|^2 + C\sum_{j=1}^{N}\sum_{d=1}^{D_v}\left(\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\xi}}_j(d)\right) - \sum_{j=1}^{N}\sum_{d=1}^{D_v}\left(\boldsymbol{\eta}_j(d)\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\eta}}_j(d)\tilde{\boldsymbol{\xi}}_j(d)\right) \\ &- \sum_{j=1}^{N}\sum_{d=1}^{D_v}\boldsymbol{\alpha}_j(d)\left(\epsilon + \boldsymbol{\xi}_j(d) - \boldsymbol{y}_j(d) + \boldsymbol{w}_d^\top\frac{\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{k}_j)} + \boldsymbol{b}(d)\right) \\ &- \sum_{j=1}^{N}\sum_{d=1}^{D_v}\tilde{\boldsymbol{\alpha}}_j(d)\left(\epsilon + \tilde{\boldsymbol{\xi}}_j(d) + \boldsymbol{y}_j(d) - \boldsymbol{w}_d^\top\frac{\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{k}_j)} - \boldsymbol{b}(d)\right),\end{aligned} \tag{6}$$

where $\boldsymbol{\eta}_j$, $\tilde{\boldsymbol{\eta}}_j$, $\boldsymbol{\alpha}_j$ and $\tilde{\boldsymbol{\alpha}}_j$ are Lagrange multipliers. These dual variables have to satisfy positivity constraints, i.e., $\boldsymbol{\eta}_j(d), \tilde{\boldsymbol{\eta}}_j(d), \boldsymbol{\alpha}_j(d), \tilde{\boldsymbol{\alpha}}_j(d) \geq 0,\ \forall j = 1, \ldots, N,\ \forall d = 1, \ldots, D_v$. It follows from the saddle point condition that the partial derivatives of the Lagrangian function $\mathcal{L}$ with respect to the primal variables $\left(\boldsymbol{w}_d, \boldsymbol{b}(d), \{\boldsymbol{\xi}_j(d), \tilde{\boldsymbol{\xi}}_j(d)\}_{j=1}^N\right)$, $d = 1, \ldots, D_v$, have to vanish for optimality, namely, we have:

$$\partial_{\boldsymbol{b}(d)}\mathcal{L} = \sum_{j=1}^{N}(\tilde{\boldsymbol{\alpha}}_j(d) - \boldsymbol{\alpha}_j(d)) = 0 \Rightarrow \sum_{j=1}^{N}(\boldsymbol{\alpha}_j(d) - \tilde{\boldsymbol{\alpha}}_j(d)) = 0, \tag{7}$$

$$\partial_{\boldsymbol{w}_d}\mathcal{L} = \boldsymbol{w}_d - \sum_{j=1}^{N}(\boldsymbol{\alpha}_j(d) - \tilde{\boldsymbol{\alpha}}_j(d))\frac{\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{k}_j)} = 0 \Rightarrow \boldsymbol{w}_d = \sum_{j=1}^{N}(\boldsymbol{\alpha}_j(d) - \tilde{\boldsymbol{\alpha}}_j(d))\frac{\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{k}_j)}, \tag{8}$$

$$\partial_{\boldsymbol{\xi}_j(d)}\mathcal{L} = C - \boldsymbol{\alpha}_j(d) - \boldsymbol{\eta}_j(d) = 0,\ \ \partial_{\tilde{\boldsymbol{\xi}}_j(d)}\mathcal{L} = C - \tilde{\boldsymbol{\alpha}}_j(d) - \tilde{\boldsymbol{\eta}}_j(d) = 0. \tag{9}$$

Let $\boldsymbol{v}_j = [\frac{\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1)}{h(\boldsymbol{k}_j)}, \ldots, \frac{\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v)}{h(\boldsymbol{k}_j)}]^\top$, $j = 1, \ldots, N$, and substitute Eqn. 8 into Eqn. 4, we obtain the following support vector expansion of the linear basis function f:

$$\begin{aligned}f(\boldsymbol{x}) &= \left[\sum_{j=1}^{N}\frac{\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1)}{h(\boldsymbol{k}_j)}\frac{\Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{x})}, \ldots, \sum_{j=1}^{N}\frac{\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v)}{h(\boldsymbol{k}_j)}\frac{\Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{x})}\right]^\top + \boldsymbol{b}, \\ &= \sum_{j=1}^{N}\frac{\Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{k}_j)}{h(\boldsymbol{x})}\boldsymbol{v}_j + \boldsymbol{b}.\end{aligned} \tag{10}$$

Remark 1 *Notice that from Eqn. 9 and the conditions* $\boldsymbol{\eta}_j(d), \tilde{\boldsymbol{\eta}}_j(d), \boldsymbol{\alpha}_j(d), \tilde{\boldsymbol{\alpha}}_j(d) \geq 0$, *we can prove that* $\boldsymbol{\alpha}_j(d), \tilde{\boldsymbol{\alpha}}_j(d) \in [0, C]$. *Furthermore, we can show that* $\boldsymbol{\alpha}_j(d) * \tilde{\boldsymbol{\alpha}}_j(d) = 0$ *(Smola & Schölkopf, 2004; Schölkopf et al., 2002). As a results,* $\boldsymbol{v}_j(d) \in \left[-\frac{C}{h(\boldsymbol{k}_j)}, \frac{C}{h(\boldsymbol{k}_j)}\right]$, $d = 1, \ldots, D_v$.

Deriving Softmax Attention. Choosing the appropriate $h(\boldsymbol{x})$ and $\Phi(\boldsymbol{x})$ allows us to derive the popular softmax attention given in Eqn. 1 and 2. In particular, if we choose $h(\boldsymbol{x}) := \sum_j^N \Phi(\boldsymbol{x})^T\Phi(\boldsymbol{k}_j)$, Eqn. 10 becomes

$$f(\boldsymbol{x}) = \sum_{j=1}^{N}\frac{\Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{k}_j)}{\sum_j^N \Phi(\boldsymbol{x})^T\Phi(\boldsymbol{k}_j)}\boldsymbol{v}_j + \boldsymbol{b} = \frac{\sum_{j=1}^N \Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{k}_j)\boldsymbol{v}_j}{\sum_j^N \Phi(\boldsymbol{x})^T\Phi(\boldsymbol{k}_j)} + \boldsymbol{b}. \tag{11}$$

We then select $\Phi(\boldsymbol{x}) = \left(a_{l_0}^{(0)}, a_1^{(1)}, \ldots, a_{l_1}^{(1)}, \ldots, a_1^{(j)}, \ldots, a_{l_j}^{(j)}, \ldots\right)$ where $l_j = \binom{D+j-1}{j}$ and

$$a_l^{(j)} = \frac{(x_1/\sqrt[4]{D})^{n_1}\ldots(x_D/\sqrt[4]{D})^{n_D}}{\sqrt{n_1!\ldots n_D!}}\ \mid\ n_1 + \cdots + n_D = j,\ 1 \leq l \leq l_j. \tag{12}$$

Since

$$\exp\left(\boldsymbol{x}^T\boldsymbol{y}\right) = \sum_{j=0}^{\infty}\frac{(\boldsymbol{x}^T\boldsymbol{y})^j}{j!} = \sum_{j=0}^{\infty}\sum_{n_1+\cdots+n_D=j}\left(\frac{x_1^{n_1}\ldots x_D^{n_D}}{\sqrt{n_1!\ldots n_D!}}\right)\left(\frac{y_1^{n_1}\ldots y_D^{n_D}}{\sqrt{n_1!\ldots n_D!}}\right), \tag{13}$$

then Eqn. 11 becomes

$$f(\boldsymbol{x}) = \sum_{j=1}^{N}\frac{\exp\left(\boldsymbol{x}^\top\boldsymbol{k}_j/\sqrt{D}\right)}{\sum_{j'=1}^N \exp\left(\boldsymbol{x}^\top\boldsymbol{k}_{j'}/\sqrt{D}\right)}\boldsymbol{v}_j + \boldsymbol{b} = \sum_{j=1}^{N}\text{softmax}\left(\boldsymbol{x}^\top\boldsymbol{k}_j/\sqrt{D}\right)\boldsymbol{v}_j + \boldsymbol{b}. \tag{14}$$

Let $\boldsymbol{x} = \boldsymbol{q}_i$, $\boldsymbol{b} = 0$ and relax the boundness constraint of $\boldsymbol{v}_j$ in Remark 1. Eqn. 14 becomes Eqn. 2 of the softmax attention (Vaswani et al., 2017). We summarize our results in the following theorem.

Theorem 1 (Softmax Attention as a Support Vector Expansion) *Given the function f defined in Eqn. 4 with $h(\boldsymbol{x}) := \sum_j^N \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{k}_j)$ and the support vector regression problem defined in Eqn. 5, we set $\boldsymbol{b} = 0$, choose $\Phi(\boldsymbol{x})$ as in Eqn. 12, and relax the boundness constraint of the variables $\boldsymbol{v}_j = [\frac{\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1)}{h(\boldsymbol{k}_j)}, \dots, \frac{\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v)}{h(\boldsymbol{k}_j)}]^\top$, where $\boldsymbol{\alpha}_j$ and $\tilde{\boldsymbol{\alpha}}_j$ are dual variables of Eqn. 5, $j = 1, \dots, N$. Then, the support vector expansion of f derived from Eqn. 5 has the form of a softmax attention*

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \mathit{softmax}\left(\boldsymbol{x}^\top \boldsymbol{k}_j / \sqrt{D}\right) \boldsymbol{v}_j. \tag{15}$$

Remark 2 *Since $\boldsymbol{b}$ is set to 0, the centering constraint of $\boldsymbol{\alpha}_j$ and $\tilde{\boldsymbol{\alpha}}_j$ in Eqn. 7 can be ignored.*

Remark 3 *Theorem 1 and its derivation can be easily extended to capture the full form of the softmax attention with the residual connection, the query matrix projection $\mathbf{W}_Q$, the key matrix projection $\mathbf{W}_K$, and the value matrix projection $\mathbf{W}_V$. We include this result in Appendix D.*

Remark 4 *The primal representation of the function f as in Eqn. 4 has the form of a neural network layer where $\mathbf{W}$ is the weight, $\boldsymbol{b}$ is the bias term, $\Phi(\boldsymbol{x})$ is the input, and $h(\boldsymbol{x})$ is the normalization term. Thus, an attention layer and a neural network layer are primal-dual of each other.*

A principled approach to developing an attention mechanism. The observation in Remark 4 suggests a principled way to construct an attention layer: *Starting from a neural network layer and a support vector regression problem, we derive the dual as a support vector expansion to attain the corresponding attention layer.* Using this approach, we derive popular attention mechanisms in Section 2.2 and propose our new attention mechanisms in Section 2.3.

2.2 Deriving Popular Attention Mechanisms as the Support Vector Expansion

In this section, we derive popular attentions such as the linear attention (Katharopoulos et al., 2020), the sparse attention (Child et al., 2019), and the multi-head attention (Vaswani et al., 2017).

2.2.1 Linear Attention

The Eqn. 11, which is obtained when choosing $h(\boldsymbol{x}) := \sum_j^N \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{k}_j)$, already matches the formula of the linear attention. Here, we can let $\boldsymbol{b} = 0$ as above and select the function Φ that results in a positive similarity function, e.g. $\Phi(\boldsymbol{x}) = \text{elu}(\boldsymbol{x}) + 1$, as in (Katharopoulos et al., 2020).

2.2.2 Sparse Attention

The sparse attention (Child et al., 2019) can be derived by fitting the function f in Eqn. 4 using a different subset $\{(\boldsymbol{k}_{m_x(1)}, \boldsymbol{y}_{m_x(1)}), \dots, (\boldsymbol{k}_{m_x(M)}, \boldsymbol{y}_{m_x(M)})\}$ of training data $\{(\boldsymbol{k}_1, \boldsymbol{y}_1), \dots, (\boldsymbol{k}_N, \boldsymbol{y}_N)\}$ for each input data $\boldsymbol{x}$, where $\mathcal{M}_{\boldsymbol{x}} = \{m_{\boldsymbol{x}}(1), \dots, m_{\boldsymbol{x}}(M)\} \subset \{1, \dots, N\}$. The support vector expansion of f is then given by

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \mathbf{1}_{\mathcal{M}_{\boldsymbol{x}}}(j) \frac{\Phi(\boldsymbol{x})^\top \Phi(\boldsymbol{k}_j)}{h(\boldsymbol{x})} \boldsymbol{v}_j + \boldsymbol{b} \tag{16}$$

where $\mathbf{1}_{\mathcal{M}_{\boldsymbol{x}}}(j) = [j \in \mathcal{M}_{\boldsymbol{x}}] := \begin{cases} 1 \text{ if } j \in \mathcal{M}_{\boldsymbol{x}} \\ 0 \text{ otherwise} \end{cases}$. Note that the subsets $\mathcal{M}_{\boldsymbol{x}}$ are different for different $\boldsymbol{x}$. When letting $\boldsymbol{x} = \boldsymbol{q}_i$ where $\boldsymbol{q}_i$, $i = 1, \dots, N$, are the query vectors and choosing $\Phi, h, \boldsymbol{b}$ as in Section 2.1, we can obtain the sparse attention in (Child et al., 2019) where the binary matrix $\mathbf{M} = \left(\mathbf{1}_{\mathcal{M}_{\boldsymbol{q}_i}}(j)\right)_{i,j=1}^{N}$ becomes the sparse masking matrix.

2.2.3 Multi-head Attention (MHA)

The MHA can be derived by solving multiple support vector regression problems and then linearly combining their outputs. In particular, given H training datasets $\{(\boldsymbol{k}_1^1, \boldsymbol{y}_1^1), \dots, (\boldsymbol{k}_N^1, \boldsymbol{y}_N^1)\}, \dots, \{(\boldsymbol{k}_1^H, \boldsymbol{y}_1^H), \dots, (\boldsymbol{k}_N^H, \boldsymbol{y}_N^H)\} \subset \mathcal{K} \times \mathcal{Y}$, where $\mathcal{K} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}^{D_v}$. We define the function f applied on the input vector $\boldsymbol{x} = [\boldsymbol{x}^1, \dots, \boldsymbol{x}^H]$ as follows

$$\boldsymbol{y} = f(\boldsymbol{x}) := \sum_{s=1}^{H} \mathbf{W}_O^s \boldsymbol{y}^s = \sum_{s=1}^{H} \mathbf{W}_O^s f^s(\boldsymbol{x}^s) = \sum_{s=1}^{H} \mathbf{W}_O^s \left(\mathbf{W}^s \frac{\Phi^s(\boldsymbol{x}^s)}{h^s(\boldsymbol{x}^s)} + \boldsymbol{b}^s\right), \tag{17}$$

where each function $f^s(\boldsymbol{x}^s) = \mathbf{W}^s \frac{\Phi^s(\boldsymbol{x}^s)}{h^s(\boldsymbol{x}^s)} + \boldsymbol{b}^s$ is fitted to the training dataset $\{(\boldsymbol{k}_1^s, \boldsymbol{y}_1^s), \dots, (\boldsymbol{k}_N^s, \boldsymbol{y}_N^s)\}$. Following the same derivation and choosing $\{\Phi^s, h^s, \boldsymbol{b}^s\}_{s=1}^H$ as in Section 2.1, we can rewrite $f(\boldsymbol{x})$ in terms of the support vector expansions of the individual functions $f^s(\boldsymbol{x}^s)$, which are the individual softmax attentions

$$f(\boldsymbol{x}) = \sum_{s=1}^{H} \mathbf{W}_O^s \left(\sum_{j=1}^{N} \frac{\Phi^s(\boldsymbol{x}^s)^\top \Phi^s(\boldsymbol{k}_j^s)}{h^s(\boldsymbol{x}^s)} \boldsymbol{v}_j^s + \boldsymbol{b}^s \right) = \sum_{s=1}^{H} \mathbf{W}_O^s \left(\sum_{j=1}^{N} \text{softmax}\left({\boldsymbol{x}^s}^\top \boldsymbol{k}_j^s / \sqrt{D} \right) \boldsymbol{v}_j^s \right). \tag{18}$$

Comparing Eqn. 18 and Eqn. 3, we see that Eqn. 18 computes the MHA when choosing $\boldsymbol{x}^s = \boldsymbol{q}_i^s$ where $\boldsymbol{q}_i^s$, $i = 1, \dots, N$, are the query vectors at the s^{th} head.

2.3 Deriving New Attention Mechanisms: Batch Normalized Attention and Multiresolution Head Attention

In this section, we employ our primal-dual framework to develop new attention mechanisms. In particular, we derive: 1) the Batch Normalized Attention from employing the batch normalization (Ioffe & Szegedy, 2015); and 2) the Attention with Scaled Heads from using different amounts of training data. By 1) and 2), we demonstrate that *new attentions can be invented by modifying the primal neural network layer and the support vector regression problem in our framework, respectively*.

2.3.1 Batch Normalized Attention

We incorporate the batch normalization into the primal form of the function f in Eqn. 4. Given a training data $\{(\boldsymbol{k}_1, \boldsymbol{y}_1), \dots, (\boldsymbol{k}_N, \boldsymbol{y}_N)\} \subset \mathcal{K} \times \mathcal{Y}$, where $\mathcal{K} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}^{D_v}$ as in Section 2.1, the resultant f is defined as follows

$$f(\boldsymbol{x}) := \mathbf{W} \frac{\Phi((\boldsymbol{x} - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})}{h((\boldsymbol{x} - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})} + \boldsymbol{b}, \tag{19}$$

where

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{k}_j,\ \mathbf{s}^{-1} = \left[\frac{1}{\sqrt{\sigma_1^2 + \epsilon}}, \dots, \frac{1}{\sqrt{\sigma_D^2 + \epsilon}} \right]^\top,\ \sigma_d^2 = \frac{1}{N} \sum_{j=1}^{N} (\boldsymbol{k}_j(d) - \boldsymbol{\mu}(d))^2. \tag{20}$$

Here, $d = 1, \dots, D$, and the mean subtraction and division by the standard deviation is performed element-wise along the feature dimension of $\boldsymbol{x}$. Following the same derivation as in Section 2.1, we derive the following support vector expansion of f

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \frac{\Phi((\boldsymbol{x} - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})^\top \Phi((\boldsymbol{k}_j - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})}{h((\boldsymbol{x} - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})} \boldsymbol{v}_j + \boldsymbol{b}. \tag{21}$$

Here, $\boldsymbol{v}_j = \left[\frac{\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1)}{h((\boldsymbol{k}_j - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})}, \dots, \frac{\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v)}{h((\boldsymbol{k}_j - \boldsymbol{\mu}) \odot \mathbf{s}^{-1})} \right]^\top$, where $\boldsymbol{\alpha}_j$ and $\tilde{\boldsymbol{\alpha}}_j$ are the dual variables, $j = 1, \dots, N$. Same as in Section 2.1, in Eqn. 21, we choose Φ as in Eqn. 12, $h(\boldsymbol{x}) := \sum_j^N \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{k}_j)$, and $\boldsymbol{b} = 0$ to obtain the the Batch Normalized Attention, which is defined as follows.

Definition 1 (Batch Normalized Attention) *Given a set of the key and value vectors $\{\boldsymbol{k}_j, \boldsymbol{v}_j\}_{j=1}^N$, for each query vector $\boldsymbol{q}_i$, $i = 1, \dots, N$, the Batch Normalized Attention (Attention-BN) computes the corresponding output vector $\boldsymbol{h}_i$ of the query $\boldsymbol{q}_i$ by the following attention formula:*

$$\boldsymbol{h}_i = \sum_{j=1}^{N} \textit{softmax}\left(((\boldsymbol{q}_i - \boldsymbol{\mu}) \odot \boldsymbol{s}^{-1})^\top ((\boldsymbol{k}_j - \boldsymbol{\mu}) \odot \boldsymbol{s}^{-1}) / \sqrt{D} \right) \boldsymbol{v}_j, \tag{22}$$

where

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{k}_j,\ \boldsymbol{s}^{-1} = \left[\frac{1}{\sqrt{\sigma_1^2 + \epsilon}}, \dots, \frac{1}{\sqrt{\sigma_D^2 + \epsilon}} \right]^\top,\ \sigma_d^2 = \frac{1}{N} \sum_{j=1}^{N} (\boldsymbol{k}_j(d) - \boldsymbol{\mu}(d))^2. \tag{23}$$

The Effect of Normalization. Expanding the dot product in the Attention-BN (see Appendix C), Eqn. 22 becomes

$$\boldsymbol{h}_i = \sum_{j=1}^{N} \text{softmax}\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d) \boldsymbol{k}_j(d) - \frac{1}{N} \sum_{j'=1}^{N} \boldsymbol{k}_{j'}(d) \boldsymbol{k}_j(d)}{\sqrt{D}(\sigma_d^2 + \epsilon)} \right) \boldsymbol{v}_j. \tag{24}$$

Eqn. 24 implies that in the Attention-BN, the similarity between the query $\boldsymbol{q}_i$ and the key $\boldsymbol{k}_j$ is adjusted by the similarity between the key $\boldsymbol{k}_j$ and all the keys $\boldsymbol{k}_{j'}$, $j' = 1, \dots, N$. In particular, if the key $\boldsymbol{k}_j$ is too similar to other keys, the query $\boldsymbol{q}_i$ will attend to it less and vice versus.

2.3.2 Attention with Scaled Heads

The *Attention with Scaled Heads*, named Attention-SH, is derived based on the derivation of the MHA in Section 2.2.3. The key idea underlying the Attention-SH is to train multiple support vector regression problems using different amounts of training data. In particular, the Attention-SH follows Eqn. 17 in Section 2.2.3 and defines the same regression function f as the MHA. However, the Attention-SH fits the function f^s, $s = 1, \dots, H$, in Eqn. 17 with training sets $\{(\boldsymbol{k}_1^1, \boldsymbol{y}_1^1), \dots, (\boldsymbol{k}_{N_1}^1, \boldsymbol{y}_{N_1}^1)\}, \dots, \{(\boldsymbol{k}_1^H, \boldsymbol{y}_1^H), \dots, (\boldsymbol{k}_{N_H}^H, \boldsymbol{y}_{N_H}^H)\} \subset \mathcal{K} \times \mathcal{Y}$ of different sizes $N_1, \dots, N_H$, where $\mathcal{K} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}^{D_v}$. The resultant support vector expansion yields the formula of the Attention-SH as in the following definition.

Definition 2 (Attention with Scaled Heads) *Given H sets of the key and value vectors $\{\boldsymbol{k}_j^1, \boldsymbol{v}_j^1\}_{j=1}^{N_1}, \dots, \{\boldsymbol{k}_j^H, \boldsymbol{v}_j^H\}_{j=1}^{N_H}$, for each set of H query vectors $\boldsymbol{q}_i^1, \dots, \boldsymbol{q}_i^H$, $i = 1, \dots, N$, the Attention with Scaled Heads (Attention-SH) computes the corresponding output vector $\boldsymbol{h}_i$ of the queries $\boldsymbol{q}_i^1, \dots, \boldsymbol{q}_i^H$ by the following attention formula:*

$$\boldsymbol{h}_i = \sum_{s=1}^{H} \mathbf{W}_O^s \left(\sum_{j=1}^{N_s} \mathit{softmax} \left({\boldsymbol{q}_i^s}^\top \boldsymbol{k}_j^s / \sqrt{D} \right) \boldsymbol{v}_j^s \right). \tag{25}$$

Remark 5 *For a given input sequence $\mathbf{X} := [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D_x}$ of N feature vectors in self-attention, in order to generate the sets of $\{\boldsymbol{k}_j^s, \boldsymbol{v}_j^s\}_{j=1}^{N_s}$ at the scale s^{th}, we can downsample the input $\mathbf{X}$ before projecting into the key matrix $\mathbf{K}$ and the value matrix $\mathbf{V}$. There are multiples approaches to downsampling $\mathbf{X}$, such as using the average-pooling, max-pooling, 1-D convolution, or K-means clustering. In this paper, we employ the average-pooling to downsample $\mathbf{X}$.*

Linear Attention with Batch Normalization and Scaled Heads. The Attention-BN/SH can be extended to use with the linear attention. In particular, in the Linear Attention-BN/SH, we replace the softmax kernel in Eqn. 22 and Eqn. 25 by the linear kernel, respectively.

3 Experimental Results

In this section, we empirically demonstrate the advantages of our Attention-BN, Attention-SH, and their combination (Attention-BN+SH) over the baseline softmax attention on the UEA time-series classification benchmark (Bagnall et al., 2018), the Long Range Arena benchmark (Tay et al., 2021), and the image classification task on the Imagenet dataset (Deng et al., 2009; Russakovsky et al., 2015). We aim to show that: (i) Attention-BN significantly outperforms the softmax baseline across tasks; (ii) Attention-SH achieves better or comparable accuracy while saving computation and memory compared to the baseline; (iii) Attention-BN+SH, which combines both Attention-BN and Attention-SH, results in the best model performance in term of accuracy and efficiency; (iv) all our proposed models help reduce the redundancy in multi-head attention and benefit learning of the long-term dependency in long input sequences; (v) Attention-BN and Attention-SH can be applied on other attention mechanisms beyond the softmax attention. When combined with the linear attention (Katharopoulos et al., 2020), the resultant Linear Attention-BN and Linear Attention-SH yield similar advantages mentioned in (i), (ii), (iii) and (iv) over the baseline linear attention.

In our experiments, we compare the proposed models with the baseline softmax and linear attentions of the same configuration. For the Attention-BN and Attention-BN+SH, we observe that recentering queries and keys alone is sufficient for improving the model performance. In addition, weighting $\boldsymbol{\mu}$ with a constant β, as in Eqn. 26 in the Appendix, enables the Attention-BN/BN+SH to flexibly adjust the effect of normalization to the attention score and help increase the accuracy. Our results are averaged over 5 runs with different seeds. Details on datasets, models, and training are provided in Appendix A.

UEA Time Series Classification. Table 1 compares the accuracy of the Attention-BN and Attention-SH with the baseline softmax attention on 10 tasks in the UEA Time Series Classification benchmark (Bagnall et al., 2018). Both Attention-BN and Attention-SH significantly outperform the softmax baseline on most tasks and on average among all tasks. When combining two models, the resulting Attention-BN+SH yields the best accuracy with more than 1% overall improvement over the softmax baseline. Notably, the Attention-SH and Attention-BN+SH are much more efficient than the baseline since they need much fewer keys and values in computing the attention output. The efficiency advantage of the Attention-SH/BN+SH is discussed and analyzed in detail in Section 4.

Table 1: Test Accuracy (%) of the Attention-BN/SH/BN+SH vs. the baseline softmax attention on the UEA Time Series Classification Archive benchmark (Bagnall et al., 2018). Our proposed attentions significantly outperform the baseline. We also include the reported results from (Zerveas et al., 2021) and (Wu et al., 2022) (in parentheses) in addition to our reproduced results.

Dataset/Model	*Baseline Softmax*	Attention-BN	Attention-SH	Attention-BN+SH
EthanolConcentration	32.08 (33.70)	33.33	33.59	**34.35**
FaceDetection	68.70 (68.10)	68.62	**68.83**	68.67
HandWriting	32.08 (30.50)	33.17	33.29	**33.45**
HeartBeat	75.77 (77.60)	76.10	76.25	**72.26**
JapaneseVowels	99.46 (99.40)	**99.55**	99.46	**99.55**
PEMS-SF	82.66 (82.10)	**84.77**	83.04	83.81
SelfRegulationSCP1	91.46 (92.50)	91.58	91.70	**92.04**
SelfRegulationSCP2	54.72 (53.90)	56.11	55.93	**57.04**
SpokenArabicDigits	99.33 (99.30)	99.23	99.34	**99.42**
UWaveGestureLibrary	84.45 (85.60)	86.46	86.77	**87.60**
Average Accuracy	72.07 (72.27)	72.89	72.82	**73.22**

Table 2: Test Accuracy (%) of the Attention-BN/SH/BN+SH vs. the baseline softmax attention on the LRA benchmark (Tay et al., 2021). Our models significantly outperform the softmax baseline.

Dataset/Model	*Baseline Softmax*	Attention-BN	Attention-SH	Attention-BN+SH
ListOps	36.76	37.32	37.08	**37.33**
Text	64.90	65.07	**65.19**	65.03
Retrieval	79.68	81.05	80.74	**81.20**
Average Accuracy	60.45	61.14	61.00	**61.19**

Table 3: Top-1 and top-5 accuracy (%) of the Attention-BN/SH/SH+BN Deit vs. the baseline Deit with the softmax attention. The Attention-BN Deit outperforms the baseline in both top-1 and top-5 accuracy. The Attention-SH/BN+SH Deit achieve comparable accuracy with the baseline while being more efficient.

Metric/Model	*Baseline Softmax Deit*	Attention-BN Deit	Attention-SH Deit	Attention-BN+SH Deit
Top-1 Acc (%)	72.23	**72.79**	72.08	72.25
Top-5 Acc (%)	91.13	**91.43**	91.05	72.14

Long Range Arena (LRA) benchmark. In this experiment, we verify the advantage of our methods over the softmax baseline on tasks that involve very long sequences (e.g., the sequence length can be up to 4K) in the LRA benchmark (Tay et al., 2021). Those tasks require the model to capture long-range dependency in the input sequence. The summarized results in Table 2 indicate significant improvements of Attention-BN/SH/BN+SH over the baseline softmax attention. Same as in the UEA Time Series experiment, on this LRA benchmark, Attention-BN and Attention-SH both outperform the softmax attention on all three tasks. Moreover, Attention-BN+SH, that combines these two attention mechanisms, results in the most accurate models on average across tasks. Specifically, for the retrieval task, the most challenging task with the largest sequence length in the LRA benchmark, Attention-BN+SH achieve a remarkable improvement of more than 1.5% over the baseline.

Image Classification on Imagenet. We corroborate the advantage of our proposed attention over the baseline softmax attention when scaled up for the large-scale ImageNet image classification task. We summarize the results in Table 3. The Deit model (Touvron et al., 2021) equiped with the Attention-BN yields better performance than the softmax baseline. Meanwhile, Attention-SH/BN+SH Deit perform on par with the baseline while being more efficient. These results, together with other results above justify the benefits of our proposed methods across various tasks and data modalities, proving the effectiveness of our primal-dual approach to develop new attentions.

4 Empirical Analysis

Efficiency Analysis. The Attention-BN+SH not only improves the accuracy of the model remarkably but also help reduce the computational and memory cost significantly. Fig.1 presents the efficiency benefits of our Attention-BN+SH trained on the retrieval task when the model dimension D and sequence lengths N grow. The efficiency advantage of our model increase as N increase. In addition, the scaled-up models (with large D) remains significantly more efficiency than the baseline. When the model dimension is 64 and sequence length is 4096, which is the standard configuration of the task, the model's FLOPS, in both training and inference, reduce almost 25%, whereas the reductions for memory usage in training and testing are 31.9% and 47.3%, respectively. Notably, this efficient model also outperforms the baseline with more than 1.5% improvement in accuracy. These results prove the benefits of applying the Attention-BN+SH for long-sequence tasks and large-scale models.

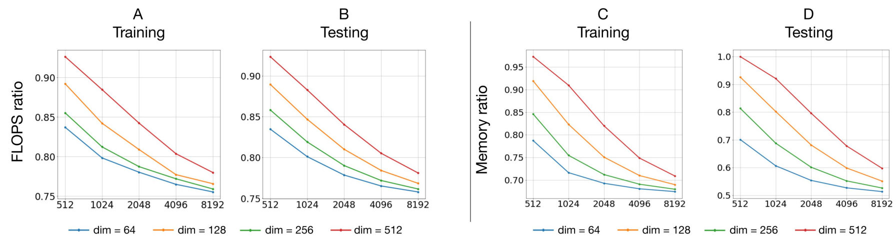

Figure 1: (Left) FLOPS ratios and (Right) memory usage ratios between the Attention-BN+SH and the softmax baseline trained on retrieval task for different model dimensions and sequence lengths. The reduction in computation and memory when using our models improves with sequence length. When scaling up the model, our methods remain significantly more beneficial than the baseline.

Table 4: Layer-average mean and standard deviation of $\mathcal{L}_2$ distances between heads of Attention-BN/SH/BN+SH versus dot-product attention transformer trained on the retrieval task. Our attentions attain greater $\mathcal{L}_2$ distances between heads than the baseline, suggesting that they capture more diverse attention patterns.

MetricModel	*Baseline Softmax*	Attention-BN	Attention-SH	Attention-BN+SH
Mean	2.01	2.45	**3.81**	3.19
Std	0.39	0.66	0.75	**1.01**

New Attentions Helps Reduce Head Redundancy. We compute the average $\mathcal{L}_2$ distances between heads to analyze the attention diversity. Given our trained models for the retrieval task, the layer-average mean and standard deviation of distances between heads are reported in Table 4. All our introduced attentions attain greater $\mathcal{L}_2$ distances compared to the baseline, reducing the risk of learning redundant heads. In particular, Attention-SH has the highest head difference, indicating the model's attention patterns are most spread out between heads.

Combining Attention-BN and Attention-SH with Other Attentions. The Attention-BN/SH/BN+SH can be extended to combine with other attention mechanisms. To demonstrate this potential of our methods, we study the Linear Attention-BN/SH/BN+SH, that combine the Attention-BN/SH/BN+SH with the linear attention (Katharopoulos et al., 2020) as explained at the end of Section 2.3. We summarize our results in Table 5 in Appendix B.1. On the UEA Time Series Classification task, the Linear Attention-BN/SH/BN+SH achieve better accuracy than the linear baseline while being more efficient.

5 Related Work

Interpretation of Attention Mechanism. Recent works have focused on understanding the attention mechanism in transformers from different perspectives. (Tsai et al., 2019) considers attention as a weighted moving average over the inputs via a smoothing kernel. (Nguyen et al., 2022) draws a connection between self-attention and nonparametric kernel regression. With this understanding, the work explores better regression estimators, e.g. the generalized Fourier nonparametric regression estimator, to improve transformers. In addition, (Cao, 2021) then shows that the linear transformer (Katharopoulos et al., 2020) corresponds to a Petrov-Galerkin projection (Reddy, 2004) and proves that the softmax normalization in the softmax attention is sufficient but not necessary. Other works that employ ordinary/partial differential equations to provide an interpretation for attention include (Lu et al., 2019; Sander et al., 2022). From a probabilistic perspective, (Tang & Matteson, 2021; Gabbur et al., 2021; Zhang & Feng, 2021) propose Gaussian mixture model frameworks to study the self-attention in transformers. Using graph-structured learning and message passing in graphical models is another attempt at understanding the attention mechanism Wang et al. (2018); Shaw et al. (2018); Kreuzer et al. (2021). Optimization perspectives of attention mechanisms are recently explored. (Sander et al., 2022) connects transformers with an optimization process across iterations by specifically constructing the core energy function. (Sahiner et al., 2022) derive finite-dimensional convex equivalence of attentions that can be solved for global optimality. Different from these approaches, our primal-dual framework focuses on deriving attention as the dual expansion of a primal neural network layer via solving a support vector regression problem. This framework allows us to not only explain many different types of attention mechanisms but also create new ones.

Efficient Transformers. To lower the quadratic computational and memory cost of transformers, efficient transformers have been studied (Roy et al., 2021). Among them are sparse transformers which incorporate sparse structures into the attention matrix (Parmar et al., 2018; Liu et al., 2018; Qiu et al., 2019; Child et al., 2019; Beltagy et al., 2020). Another class of efficient transformers are models that aim to have better coverage by integrating different access patterns (Child et al., 2019; Ho et al., 2019), which can also be learned from the data (Kitaev et al., 2020; Roy et al., 2021; Tay et al., 2020). Also, an emerging body of work is proposed to distill and prune the model, including (Sanh et al., 2019; Sun et al., 2019; Voita et al., 2019b; Sajjad et al., 2020). In other works, a side memory module is utilized in order to access multiple tokens simultaneously (Lee et al., 2019; Sukhbaatar et al., 2019; Asai & Choi, 2020; Beltagy et al., 2020). In another line of work, low-rank and kernelization methods have recently been proposed to improve the computational and memory efficiency of self-attention calculation (Tsai et al., 2019; Wang et al., 2020; Katharopoulos et al., 2020; Choromanski et al., 2021; Shen et al., 2021; Peng et al., 2021). Our Attention-SH/BN+SH is orthogonal to these methods.

6 Concluding Remarks

In this paper, we derive self-attention as a support vector expansion that solves a support vector regression (SVR) problem and provide a principled primal-dual framework to analyze and synthesize attention mechanisms. We show that many popular attention mechanisms can be derived under our framework. We then use our framework to invent two new attention mechanisms, the Batch Normalized Attention (Attention-BN) and the Attention with Scaled Heads (Attention-SH), that improve the accuracy and the efficiency of the baseline softmax attention. In our work, we approximate and learn the dual variables $\boldsymbol{\alpha}_j$ and $\tilde{\boldsymbol{\alpha}}_j$ using the value vector $\boldsymbol{v}_j$, $j = 1, \ldots, N$ in self-attention. It is natural to include more inductive biases and structures of those dual variables from solving the dual optimization problem of SVR into the value vectors $\boldsymbol{v}_j$. Furthermore, it is interesting to incorporate robust optimization methods into our framework to develop robust transformers. We leave these exciting research ideas as future work.

REFERENCES

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3159–3166, 2019.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6816–6826, 2021. doi: 10.1109/ICCV48922.2021.00676.

Akari Asai and Eunsol Choi. Challenges in information seeking qa: Unanswerable questions and paragraph retrieval. *arXiv preprint arXiv:2010.11915*, 2020.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxZX20qFQ.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in Neural Information Processing Systems*, 34, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://www.aclweb.org/anthology/W19-4828.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Prasad Gabbur, Manjot Bilkhu, and Javier Movellan. Probabilistic attention for interactive segmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://www.aclweb.org/anthology/D19-1275.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pp. 5156–5165. PMLR, 2020.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. *arXiv preprint arXiv:1702.00887*, 2017.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017. URL http://arxiv.org/abs/1703.03130.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1015.

Nikita Nangia and Samuel Bowman. ListOps: A diagnostic dataset for latent tree learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 92–99, New Orleans, Louisiana, USA, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-4013. URL https://www.aclweb.org/anthology/N18-4013.

Tam Minh Nguyen, Tan Minh Nguyen, Dung DD Le, Duy Khuong Nguyen, Viet-Anh Tran, Richard Baraniuk, Nhat Ho, and Stanley Osher. Improving transformers with probabilistic attention keys. In *International Conference on Machine Learning*, pp. 16595–16621. PMLR, 2022.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL https://www.aclweb.org/anthology/D16-1244.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4055–4064. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/parmar18a.html.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=QtTKTdVrFBB.

Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*, 2019.

Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The acl anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944, 2013.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI report*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

JN Reddy. *An introduction to the finite element method*, volume 1221. McGraw-Hill New York, 2004.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. URL https://www.aclweb.org/anthology/2021.tacl-1.4.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Arda Sahiner, Tolga Ergen, Batu Ozturkler, John Pauly, Morteza Mardani, and Mert Pilanci. Unraveling attention via convex duality: Analysis and interpretations of vision transformers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19050–19088. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/sahiner22a.html.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. Poor man's bert: Smaller and faster transformer models. *arXiv e-prints*, pp. arXiv–2004, 2020.

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pp. 3515–3530. PMLR, 2022.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3531–3539, 2021.

Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

Binh Tang and David S. Matteson. Probabilistic transformer for time series analysis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=HfpNVDg3ExA.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse Sinkhorn attention. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9438–9447. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/tay20a.html.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qVyeW-grC2k.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention, 2021.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 63–76, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4808. URL https://www.aclweb.org/anthology/W19-4808.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019b.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

Zifeng Wang and Jimeng Sun. TransTab: Learning Transferable Tabular Transformers Across Tables. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, 2022.

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. In *International Conference on Machine Learning*, 2022.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124, 2021.

Shaolei Zhang and Yang Feng. Modeling concentrated cross-attention for neural machine translation with Gaussian mixture model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1401–1411, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.121. URL https://aclanthology.org/2021.findings-emnlp.121.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision. In *NeurIPS*, 2021.

Supplement to "A Primal-Dual Framework for Transformers and Neural Networks"

A Additional details on the experiments

This section provides datasets, models, and training details for experiments in Section 3. As mentioned in Section 3, for Attention-BN/BN+SH models, recentering keys and values alone is sufficient for accuracy improvement, and we weight the mean $\boldsymbol{\mu}$ in Eqn 22 with a constant β. Hence Eqn 22 is simplified to:

$$\boldsymbol{h}_i = \sum_{j=1}^{N} \mathrm{softmax}\left((\boldsymbol{q}_i - \beta\boldsymbol{\mu})^\top(\boldsymbol{k}_j - \beta\boldsymbol{\mu})/\sqrt{D}\right)\boldsymbol{v}_j. \tag{26}$$

In our experiments, we consider the constant β in Attention-BN/BN+SH and the different downsampling scales in Attention-SH/SH+BN as hyper-parameters to finetune. All of our experiments are conducted on a server with 4 NVIDIA A100 GPUs.

A.1 UEA time series classification

Datasets and metrics The benchmark (Bagnall et al., 2018) consists of 30 datasets. Following (Wu et al., 2022), we choose 10 datasets, which vary in input sequence lengths, the number of classes, and dimensionality, to evaluate our models on temporal sequences. We report the test accuracy as evaluation for the benchmark.

Models and baselines The experiment setups and configurations for the softmax/linear baseline and our models are the same as in (Wu et al., 2022) [1] (for the PEMS-SF, SelfRegulationSCP2, UWaveGestureLibrary datasets) and (Zerveas et al., 2021) [2] (for the other tasks). In all models, the number of heads is 8, whereas the model dimension and number of transformer layers are varied. For Attention-SH/SH+BN, we downsample keys and values by the factor of 2, after every two successive heads.

A.2 Long range arena benchmark

Datasets and metrics We adopt the tasks: Listops (Nangia & Bowman, 2018), byte-level IMDb reviews text classification (Maas et al., 2011), and byte-level document retrieval (Radev et al., 2013) in the LRA benchmark for our experiments. They consist of long sequences of length $2K$, $4K$, and $4K$, respectively. The evaluation protocol and metric are the same as in (Tay et al., 2021).

Models and baselines All our models and softmax/linear baselines follow the same architecture and configuration as in (Zhu et al., 2021)[3]. Each model consists of two layers and 64 embedding dimensions. While one head at each layer remains intact, the keys and values of the other heads are halved in our Attention-SH/SH+BN experiments.

A.3 Image Classification on Imagenet

Datasets and metrics The ImageNet dataset (Deng et al., 2009; Russakovsky et al., 2015) consists of $1.28M$ training images and $50K$ validation images. The task is to classify 1000 categories. Top-1 and top-5 accuracies are reported.

Models and baselines Our baseline is DeiT-tiny model (Touvron et al., 2021) with 12 transformer layers, 4 attention heads per layer, and the model dimension of 192. For model setting and setting and configuration, we follow (Touvron et al., 2021)[4]. The downsampling scales in Attention-SH/BN+SH models are $[1, 1, 2, 4]$ for 4 heads at each layer, respectively.

[1] Implementation available at https://github.com/thuml/Flowformer.
[2] Implementation available at https://github.com/gzerveas/mvts_transformer.
[3] Implementation available at https://github.com/NVIDIA/transformer-ls.
[4] Implementation available at https://github.com/facebookresearch/deit.

Table 5: Test Accuracy (%) of the Linear Attention-BN/SH/BN+SH vs. the baseline linear attention (Katharopoulos et al., 2020) on the UEA Time Series Classification Archive benchmark (Bagnall et al., 2018). Our proposed attentions outperform the baseline.

Dataset/Model	*Baseline Linear*	Linear Attention-BN	Linear Attention-SH	Linear Attention-BN+SH
EthanolConcentration	33.84	**34.98**	34.76	34.35
FaceDetection	69.17	69.22	**69.38**	69.12
HandWriting	32.87	32.86	32.82	**32.98**
HeartBeat	75.61	75.78	74.96	**75.94**
JapaneseVowels	99.37	**99.60**	99.28	99.33
PEMS-SF	83.43	85.74	**86.51**	84.97
SelfRegulationSCP1	90.90	91.81	90.76	**91.92**
SelfRegulationSCP2	55.18	**56.11**	54.44	55.74
SpokenArabicDigits	99.07	99.01	99.03	98.91
UWaveGestureLibrary	85.63	**86.04**	84.89	85.78
Average Accuracy	72.51	**73.12**	72.68	72.90

B Additional Experimental Results

B.1 UEA Time Series Classification using the Linear Attention-BN/SH/BN+SH

Table 5 summarizes the comparison between the Linear Attention-BN/SH/BN+SH and the baseline linear attention on the UEA Time Series Classification task. The Linear Attention-BN/SH/BN+SH achieve better accuracy than the linear baseline while being more efficient.

C Batch Normalized Attention: Derivation of Eqn. 24

$$
\begin{aligned}
\boldsymbol{h}_i &= \sum_{j=1}^{N} \mathrm{softmax}\left(\sum_{d=1}^{D} \frac{(\boldsymbol{q}_i(d)-\boldsymbol{\mu}(d))(\boldsymbol{k}_j(d)-\boldsymbol{\mu}(d))}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)\boldsymbol{v}_j \\
&= \sum_{j=1}^{N} \mathrm{softmax}\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_j(d)-\boldsymbol{q}_i(d)\boldsymbol{\mu}(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_j(d)+\boldsymbol{\mu}(d)\boldsymbol{\mu}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)\boldsymbol{v}_j \\
&= \sum_{j=1}^{N} \frac{\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_j(d)-\boldsymbol{q}_i(d)\boldsymbol{\mu}(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_j(d)+\boldsymbol{\mu}(d)\boldsymbol{\mu}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)}{\sum_{j'=1}^{N}\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_{j'}(d)-\boldsymbol{q}_i(d)\boldsymbol{\mu}(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_{j'}(d)+\boldsymbol{\mu}(d)\boldsymbol{\mu}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)}\boldsymbol{v}_j \\
&= \sum_{j=1}^{N} \frac{\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_j(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_j(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{\mu}(d)\boldsymbol{\mu}(d)-\boldsymbol{q}_i(d)\boldsymbol{\mu}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)}{\sum_{j'=1}^{N}\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_{j'}(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_{j'}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{\mu}(d)\boldsymbol{\mu}(d)-\boldsymbol{q}_i(d)\boldsymbol{\mu}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)}\boldsymbol{v}_j \\
&= \sum_{j=1}^{N} \frac{\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_j(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_j(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)}{\sum_{j'=1}^{N}\exp\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_{j'}(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_{j'}(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)}\boldsymbol{v}_j \\
&= \sum_{j=1}^{N} \mathrm{softmax}\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_j(d)-\boldsymbol{\mu}(d)\boldsymbol{k}_j(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)\boldsymbol{v}_j \\
&= \sum_{j=1}^{N} \mathrm{softmax}\left(\sum_{d=1}^{D} \frac{\boldsymbol{q}_i(d)\boldsymbol{k}_j(d)-\frac{1}{N}\sum_{j'=1}^{N}\boldsymbol{k}_{j'}(d)\boldsymbol{k}_j(d)}{\sqrt{D}(\sigma_d^2+\epsilon)}\right)\boldsymbol{v}_j. \qquad (27)
\end{aligned}
$$

D Attention with the Residual Connection and Matrix Projections

In this supplement, we first discuss attention with the residual connection and matrix projections in Appendix D.

Suppose we are given a training data $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^{D_x}$ and $\mathcal{Y} = \mathbb{R}^{D_v}$. Here, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are attention keys in self-attention, and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are the training targets. In order to derive the attention with the residual connection and query, key, and value matrix projections, we define the function f as follows

$$\boldsymbol{y} = f(\boldsymbol{x}) := \mathbf{W}\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x})}{h(\boldsymbol{x})} + \boldsymbol{x} + \boldsymbol{b}, \tag{28}$$

where $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^{D_x}$, $\mathbf{W}^{\text{proj}} = [\boldsymbol{w}_1^{\text{proj}}, \dots, \boldsymbol{w}_D^{\text{proj}}]^\top \in \mathbb{R}^{D \times D_x}$, $\Phi(\cdot) = [\phi_1(\cdot), \dots, \phi_{D_\phi}(\cdot)] : \mathbb{R}^D \to \mathbb{R}^{D_\phi}$, $\mathbf{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_{D_v}]^\top \in \mathbb{R}^{D_v \times D_\phi}$, $\boldsymbol{b} \in \mathbb{R}^{D_v}$, and $h(\boldsymbol{x})$ is a vector-scalar function. We fit the function f to the training data $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \dots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$ with an $\mathbb{L}_2$ regularization on $\mathbf{W}$ and $\mathbf{W}^{proj}$ by solving the following convex optimization problem:

$$\begin{aligned}
&\underset{\substack{\mathbf{W}, \mathbf{W}^{\text{proj}} \\ \boldsymbol{\xi}_j, \tilde{\boldsymbol{\xi}}_j, j=1,\dots,N}}{\text{minimize}} \quad \frac{1}{2}\sum_{d=1}^{D_v} \|\boldsymbol{w}_d\|^2 + \frac{1}{2}\sum_{d=1}^{D_v} \|\boldsymbol{w}_d^{\text{proj}}\|^2 + C\sum_{j=1}^{N}\sum_{d=1}^{D_v}\Big(\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\xi}}_j(d)\Big) \\
&\text{subject to} \quad \begin{cases} \boldsymbol{y}_j(d) - \boldsymbol{w}_d^\top \Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)/h(\boldsymbol{x}_j) - \boldsymbol{x}_j - \boldsymbol{b}(d) \le \epsilon + \boldsymbol{\xi}_j(d) \\ \boldsymbol{w}_d^\top \Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)/h(\boldsymbol{x}_j) + \boldsymbol{x}_j + \boldsymbol{b}(d) - \boldsymbol{y}_j(d) \le \epsilon + \tilde{\boldsymbol{\xi}}_j(d) \\ \boldsymbol{\xi}_j(d), \tilde{\boldsymbol{\xi}}_j(d) \ge 0 \end{cases}, \; j = 1, \dots, N, \; d = 1, \dots, D_v.
\end{aligned} \tag{29}$$

The Lagrangian of the optimization problem 29 is given by

$$\begin{aligned}
\mathcal{L}_1 := &\frac{1}{2}\sum_{d=1}^{D_v} \|\boldsymbol{w}_d\|^2 + \frac{1}{2}\sum_{d=1}^{D_v} \|\boldsymbol{w}_d^{\text{proj}}\|^2 + C\sum_{j=1}^{N}\sum_{d=1}^{D_v}\Big(\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\xi}}_j(d)\Big) - \sum_{j=1}^{N}\sum_{d=1}^{D_v}\Big(\boldsymbol{\eta}_j(d)\boldsymbol{\xi}_j(d) + \tilde{\boldsymbol{\eta}}_j(d)\tilde{\boldsymbol{\xi}}_j(d)\Big) \\
&- \sum_{j=1}^{N}\sum_{d=1}^{D_v} \boldsymbol{\alpha}_j(d)\left(\epsilon + \boldsymbol{\xi}_j(d) - \boldsymbol{y}_j(d) + \boldsymbol{w}_d^\top \frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x}_j)} + \boldsymbol{x}_j + \boldsymbol{b}(d)\right) \\
&- \sum_{j=1}^{N}\sum_{d=1}^{D_v} \tilde{\boldsymbol{\alpha}}_j(d)\left(\epsilon + \tilde{\boldsymbol{\xi}}_j(d) + \boldsymbol{y}_j(d) - \boldsymbol{w}_d^\top \frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x}_j)} - \boldsymbol{x}_j - \boldsymbol{b}(d)\right),
\end{aligned} \tag{30}$$

Similar to the derivation in Section 2.1, the partial derivatives of $\mathcal{L}_1$ with respect to the primal variable $\boldsymbol{w}_d$, $d = 1, \dots, D_v$, have to vanish for optimality, which leads to

$$\partial_{\boldsymbol{w}_d}\mathcal{L}_1 = \boldsymbol{w}_d - \sum_{j=1}^{N}(\boldsymbol{\alpha}_j(d) - \tilde{\boldsymbol{\alpha}}_j(d))\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x}_j)} = 0 \Rightarrow \boldsymbol{w}_d = \sum_{j=1}^{N}(\boldsymbol{\alpha}_j(d) - \tilde{\boldsymbol{\alpha}}_j(d))\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x}_j)}. \tag{31}$$

Note that here we only find the form of the optimal solution for $\mathbf{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_{D_v}]^\top$. The optimal value of $\mathbf{W}^{\text{proj}}$ can then be found by optimization algorithm such as the (stochastic) gradient descent when training the transformer.

Let $\boldsymbol{v}_j = [\frac{\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1)}{h(\boldsymbol{x}_j)}, \dots, \frac{\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v)}{h(\boldsymbol{x}_j)}]^\top$, $j = 1, \dots, N$, we obtain the following support vector expansion of the function f:

$$\begin{aligned}
f(\boldsymbol{x}) &= \left[\sum_{j=1}^{N}(\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1))\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x}_j)}, \dots, \sum_{j=1}^{N}(\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v))\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x}_j)}\right]^\top \frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x})}{h(\boldsymbol{x})} + \boldsymbol{x} + \boldsymbol{b}, \\
&= \left[\sum_{j=1}^{N}\frac{\boldsymbol{\alpha}_j(1) - \tilde{\boldsymbol{\alpha}}_j(1)}{h(\boldsymbol{x}_j)}\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x})^\top\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x})}, \dots, \sum_{j=1}^{N}\frac{\boldsymbol{\alpha}_j(D_v) - \tilde{\boldsymbol{\alpha}}_j(D_v)}{h(\boldsymbol{x}_j)}\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x})^\top\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x})}\right]^\top + \boldsymbol{x} + \boldsymbol{b}, \\
&= \underbrace{\sum_{j=1}^{N}\frac{\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x})^\top\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)}{h(\boldsymbol{x})}\boldsymbol{v}_j + \boldsymbol{x}}_{\text{Residual connection}} + \boldsymbol{b}.
\end{aligned} \tag{32}$$

Here, the support vector expansion of f already includes a residual connection. The softmax attention can then be derived by selecting $h(\boldsymbol{x}) := \sum_j^N \Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x})^T\Phi(\mathbf{W}^{\text{proj}}\boldsymbol{x}_j)$ and choosing Φ as in Eqn. 12 in Section 2.1. Note that in Eqn. 32, $\{\boldsymbol{x}_j\}_{j=1}^N$ and $\boldsymbol{x}$ are the training samples and test sample, respectively. In order to derive the query, key and value matrix projections in attention, we can then relax Eqn. 32 by letting $\mathbf{W}^{\text{proj}}\boldsymbol{x}_j = \mathbf{W}_K\boldsymbol{x}_j$, $\mathbf{W}^{\text{proj}}\boldsymbol{x} = \mathbf{W}_Q\boldsymbol{x}$, $\boldsymbol{v}_j = \mathbf{W}_V\boldsymbol{x}_j$ and choosing the test sample $\boldsymbol{x}$ among the training samples $\{\boldsymbol{x}_j\}_{j=1}^N$.

Remark 6 *Here, for self-attention, we choose the test sample $\boldsymbol{x}$ among the training samples $\{\boldsymbol{x}_j\}_{j=1}^N$ to compute the attention score of a token to other tokens in the same sequence. For cross-attention where a token in a sequence attends to tokens in another sequence, this constraint can be removed.*