Optimized Coding and Parameter Selection for Efficient FPGA Design of Attention Mechanisms

Ehsan Kabir, Austin R. J. Downey, Jason D. Bakos, David Andrews, Miaoqing Huang

UNIVERSITY OF ARKANSAS

UNIVERSITY OF South Carolina

Introduction

- Deep neural networks (DNN) have shown significant advancements in natural language processing, machine translation, computer vision.
- An important feature named attention mechanism within any DNN enables a high level of computational parallelism for both the training and inference phases.
- Suitable for acceleration on hardware like FPGAs, due to FPGA's high degree of parallelism, low latency, and energy efficiency.
- We optimized high level synthesis (HLS) code to increase parallel DSP consumption.
- We introduced an efficient tiling technique and optimized the value of parameters within an attention layer to improve latency without exhausting computational and memory resources.

Accelerating the Computation within the Attention Layer of a Neural Network

- A novel architecture to enhance parallel processing within the attention layer.
- An efficient coding in HLS to ensure high utilization of DSPs.
- An efficient tiling of weight matrices to accommodate large models in on-chip memory.
- A theoretical model validating both predicted and experimental latency.

Tiling Technique

Load inputs into the Input BRAM

Load Values of Each Tile into Weight BRAM

$$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \begin{array}{cccccc} \text{Tile 1} & \text{Tile 2} & \text{Tile 3} & \text{Tile 4} & \text{Tile 5} & \text{Tile 6} \\ \hline W_{00} & W_{01} & W_{02} & W_{03} & W_{04} & W_{05} \\ W_{10} & W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{20} & W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \end{array}$$

1st Iteration: Computation with 1st Tile

Computation output of 1st Tile

$$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \overset{\text{Tile 1}}{\begin{bmatrix} W_{00} \\ W_{10} \\ W_{20} \end{bmatrix}} \longrightarrow \begin{bmatrix} X_{00}W_{00} + X_{01}W_{10} + X_{02}W_{20} \\ X_{10}W_{00} + X_{11}W_{10} + X_{12}W_{20} \end{bmatrix}$$

2nd Iteration: Computation with 2nd Tile

Computation output of 2nd Tile

$$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \overset{\text{Tile 2}}{\begin{bmatrix} W_{01} \\ W_{11} \\ W_{21} \end{bmatrix}} \longrightarrow \begin{bmatrix} X_{00}W_{01} + X_{01}W_{11} + X_{02}W_{21} \\ X_{10}W_{01} + X_{11}W_{11} + X_{12}W_{21} \end{bmatrix}$$

nth Iteration: Computation with nth Tile

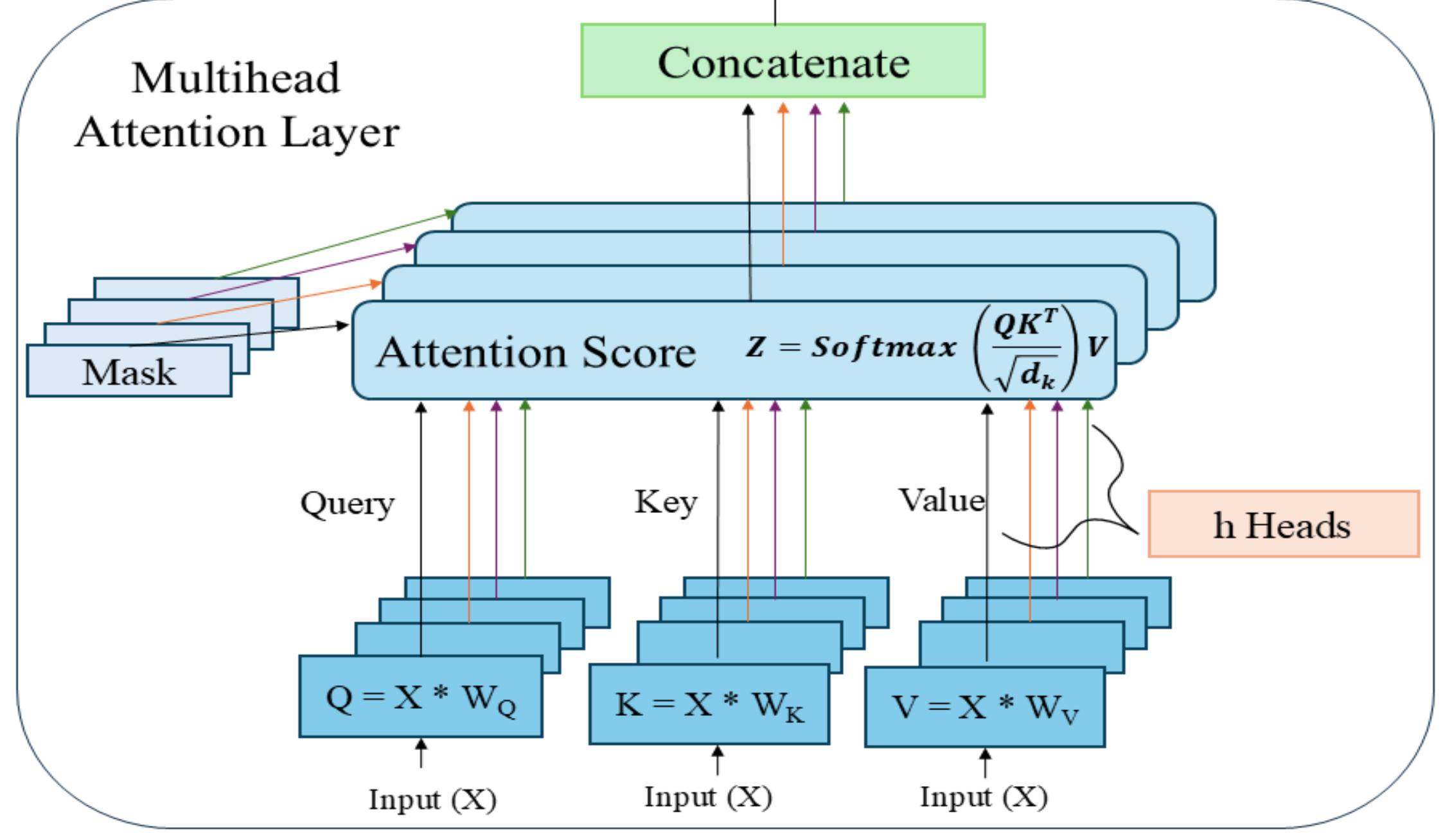
Computation output of nth Tile

$$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \overset{\text{Tile n}}{\begin{bmatrix} W_{01} \\ W_{11} \\ W_{21} \end{bmatrix}} \longrightarrow \begin{bmatrix} X_{00}W_{0n} + X_{01}W_{1n} + X_{02}W_{2n} \\ X_{10}W_{0n} + X_{11}W_{1n} + X_{12}W_{2n} \end{bmatrix}$$

Final Matrix = Output for 1st tile + Output for 2nd tile + ...+ Output for nth tile

Background

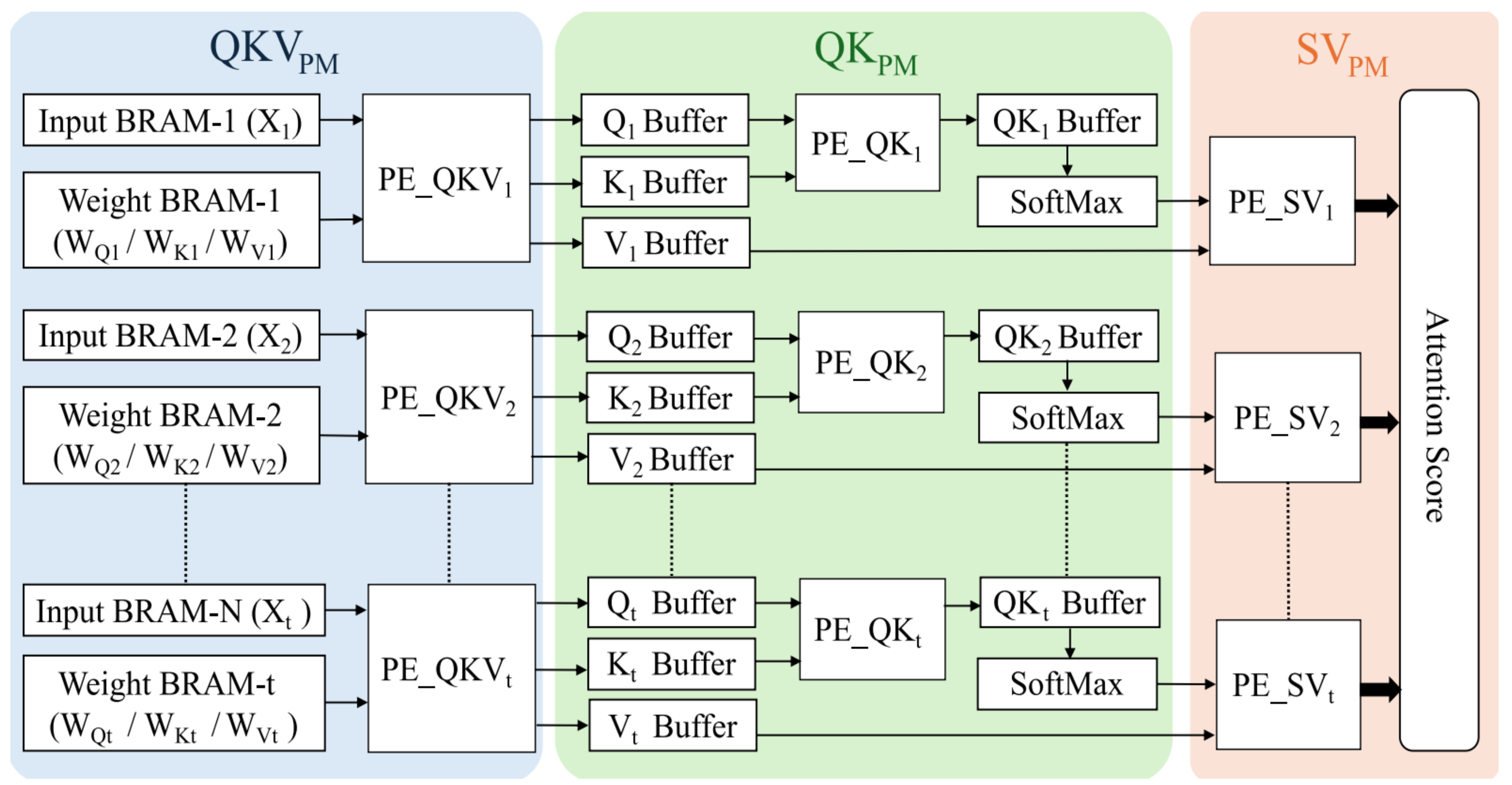
The input sequence X is linearly mapped into Query (Q), Key (K), Value (V) matrices using weights and biases. The parameter dk = dmodel/h is the 2nd dimension of Q and K. dmodel is a hyperparameter called embedding dimension and h is number of heads.



Accelerator Architecture

- Designed with C in Vitis HLS.
- Three main processing modules: QKV_{PM} , QK_{PM} and SV_{PM}. QKV_{PM} : Generates Q, K, V matrices. QK_{PM} : Matrix-matrix multiplication operations between the Q and K matrices.
- SV_{PM} : Matrix-matrix multiplication operations with V and the output from QK_{PM}.

Accelerator Architecture for Attention Mechanism



HLS Design Technique

Algorithm 4 Q, K, V Calculation

```
1:  for $(i = 1; i <= Sequence\ Length; i = i + 1)$ do
2:      #pragma HLS pipeline off
3:      $S_q \leftarrow 0$
4:      $S_k \leftarrow 0$
5:      $S_v \leftarrow 0$
6:      for $(k = 1; k <= \frac{d_{model}}{h}; k++)$ do
7:          #pragma HLS pipeline II = 1
8:          for $(j = 1; j <= Tiles; j++)$ do
9:              $S_q \leftarrow S_q + x[i][j] \times w_q[k][j];$
10:             $S_q \leftarrow S_q + x[i][j] \times w_k[k][j];$
11:             $S_q \leftarrow S_q + x[i][j] \times w_v[k][j];$
12:         end for
13:         $Q[i][k] \leftarrow Q[i][k] + S_q;$
14:         $K[i][k] \leftarrow K[i][k] + S_k;$
15:         $V[i][k] \leftarrow V[i][k] + S_v;$
16:     end for
17: end for
```

Algorithm 2 Load Weights

```
1:  for $(i = 1; i <= \frac{Embedding\ Dimension}{Number\ of\ Heads}; i = i + 1)$ do
2:      #pragma HLS pipeline off
3:      for $(j = 1; j <= Tiles; j = j + 1)$ do
4:          #pragma HLS pipeline II = 1
5:          $w_q[i][j] \leftarrow weights\_Q[index];$
6:          $w_k[i][j] \leftarrow weights\_K[index];$
7:          $w_v[i][j] \leftarrow weights\_V[index];$
8:          $index \leftarrow index + 1;$
9:      end for
10: end for
```

Algorithm 3 Load Biases

```
1:  for $(i = 1; i <= \frac{Embedding\ Dimension}{Number\ of\ Heads}; i = i + 1)$ do
2:      #pragma HLS pipeline II = 1
3:      $b_q[i] \leftarrow bias\_Q[index];$
4:      $b_k[i] \leftarrow bias\_K[index];$
5:      $b_v[i] \leftarrow bias\_V[index];$
6:      $index \leftarrow index + 1;$
7:  end for
```

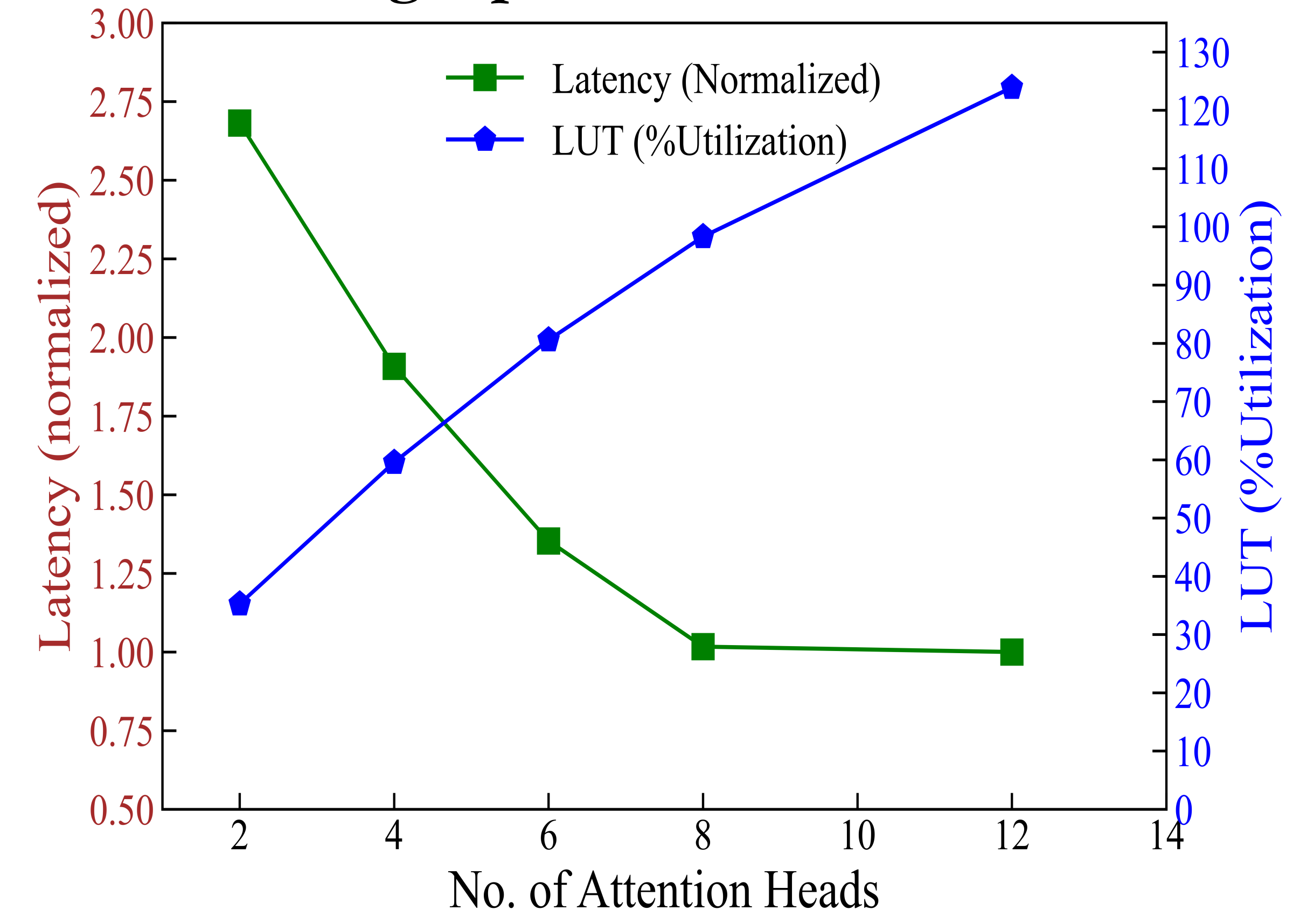
Analytical Model

$$LI = [(d_{model} - 1) \times 1 + PD\_L] \times SL$$
$$LB = (\frac{d_{model}}{h} - 1) \times 1 + PD\_L$$
$$LIA = [(TS - 1) \times 1 + PD\_L] \times SL$$
$$LWA = [(\frac{d_{model}}{h} - 1) \times 1 + PD\_L] \times SL$$
$$SA = [(\frac{d_{model}}{h} - 1) \times 1 + PD\_MHA] \times SL$$
$$BA = [(\frac{d_{model}}{h} - 1) \times 1 + PD\_BA] \times SL$$
$$S = [(SL - 1) \times 1 + PD\_S] \times SL$$
$$SV = [(\frac{d_{model}}{h} - 1) \times 1 + PD\_SV] \times SL$$

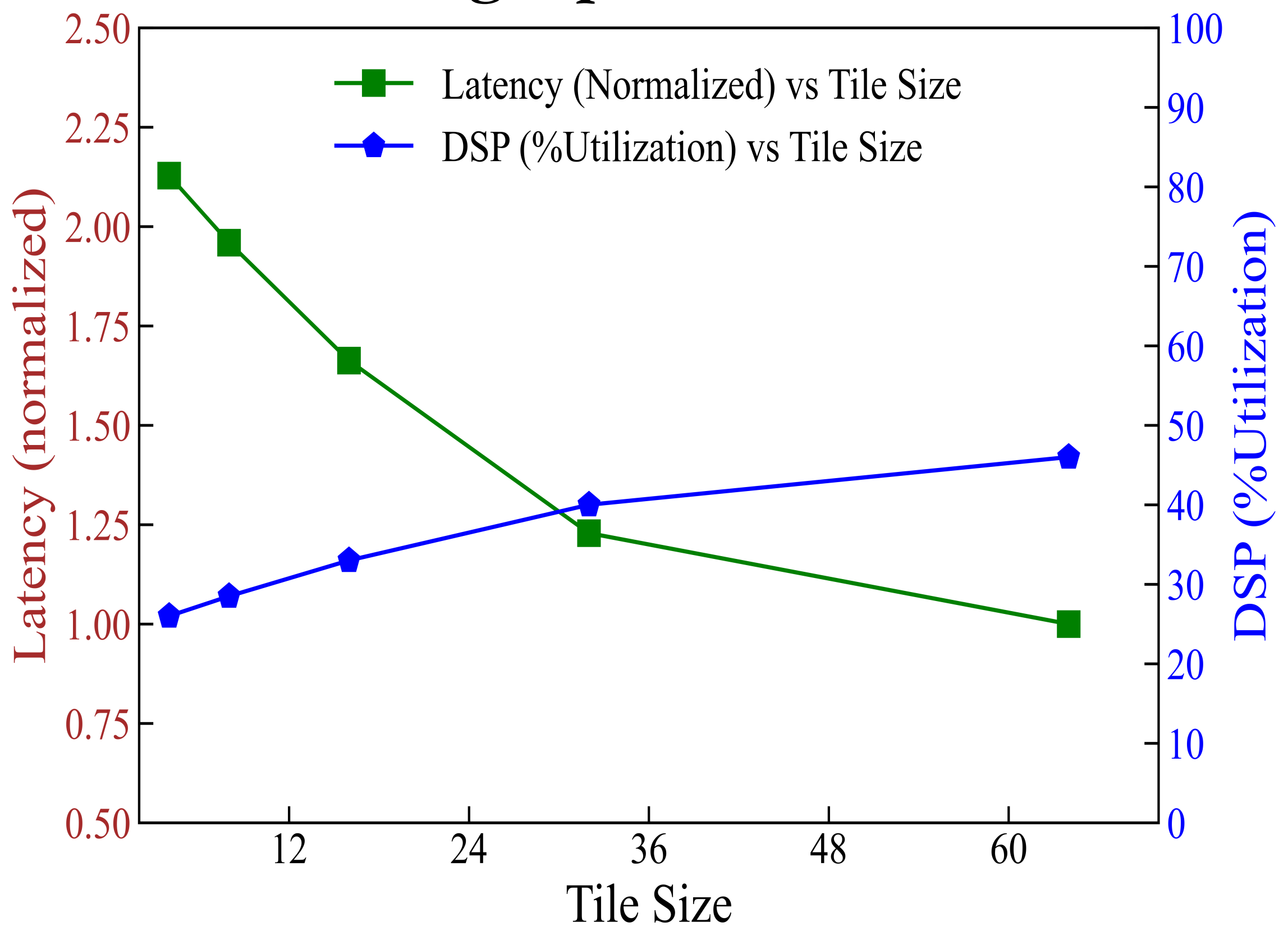
$$No.\ of\ DSPs = 9 \times h + 3 \times h \times TS + 2 \times SL \times h + 2 \times d_{model}$$

$$LAT_{total}(cc) = LI + LB + LIA + LWA + SA + BA + S + SV$$
$$LAT_{total}(ms) = \frac{LAT_{total}(cc) \times 10^3}{Frequency\ (Hz)}$$

Choosing Optimum Head Number



Choosing Optimum Tile Size



Validation of Experimental and Analytical Results

Method	Sequence Length	Embedding Dimension	Number of Heads	Tile Size	DSPs	Frequency (MHz)	Latency (ms) Attention (SA)
Experimental	64	768	8	64	4157		0.94
Analytical					4168		0.98
Experimental	32	768	8	64	3898	400	0.534
Analytical					3656		0.579
Experimental	64	512	8	64	3887		0.597
Analytical					3656		0.592

References

[1] E. Kabir et al., "Accelerating lstm-based high-rate dynamic system models," in FPL Conference, 2023.
[2] B. Li et al., "Ftrans," in International Symposium on Low Power Electronics and Design, 2020.
[3] E. Kabir et al., "Protea," in SC24-W: Workshops, 2024.