LL-ViT: Edge Deployable Vision Transformers with Look Up Table Neurons

Shashank Nag[1*], Alan T.L. Bacellar[1], Zachary Susskind[1], Anshul Jha[2],
Logan Liberty[1], Aishwarya Sivakumar[1], Eugene B. John[2], Krishnan Kailas[3],
Priscila M.V. Lima[4], Neeraja J. Yadwadkar[1], Felipe M.G. França[5], Lizy K. John[1*]
[1]The University of Texas at Austin, USA
[2]The University of Texas at San Antonio, USA [3]Independent Researcher[†]
[4]Federal University of Rio de Janeiro, Brazil [5]Instituto de Telecomunicações, Portugal

***Abstract*—Vision Transformers have been tremendously successful in computer vision tasks. However, their large computational, memory, and energy demands are a challenge for edge inference on FPGAs – a field that has seen a recent surge in demand. We recognize the benefits of recent works on logic and Look Up Table (LUT) based networks, such as LogicNets, NeuraLUT, DWN, among others, in offering models that simultaneously reduce both the memory and compute footprints. However, these models natively do not perform well on common vision tasks, such as CIFAR-10/100. In this work, we propose LL-ViT, a novel edge optimized vision transformer design that integrates layers of LUT neurons within the transformer architecture. Based on our characterization that reveals that a majority of model weights and computations are from the channel mixer (MLP layer), we design an alternate LUT-based channel mixer, and simultaneously develop an FPGA-based accelerator for LL-ViT. Contrary to some attempts to replace each multiplication with a table lookup, our architecture utilizes a neural learning approach which natively learns the LUT functions. This approach allows for reduced model sizes, and a computational and energy-efficient inference solution for vision transformer models. Evaluating on edge-suitable workloads, we achieve accuracies of 95.5% on CIFAR-10, 78.8% on CIFAR-100, and 60.9% on Tiny-ImageNet datasets, comparable to the baseline transformer. LL-ViT eliminates over 60% of the model weights and 50% of the multiplications in the model, and achieves 1.9× energy efficiency and 1.3× lower latency over an integer quantized ViT accelerator, while also offering superior throughput against prior works at a 10.9W power budget.**

I. Introduction

Vision Transformers (ViTs) have recently garnered significant attention due to their versatility and strong performance across a wide range of vision tasks, including image classification, object detection, and semantic segmentation [1], [2], [3], [4]. With the rise of generative models such as DALL-E [5], the importance of ViTs has only grown, positioning them as a key component in modern computer vision systems.

There is a growing demand for deploying computer vision models on edge devices, driven by applications in robotics, autonomous systems, and other latency-sensitive edge domains. While convolutional neural networks (CNNs) have traditionally dominated this space, a recent market trend highlights increasing interest in deploying ViTs at the edge [6]. Many ViTs exhibit regular inter-layer structure, which offers opportunities for efficient dataflow design and hardware acceleration. Furthermore, as ViT architectures continue to evolve rapidly, FPGAs present a promising platform for their deployment owing to their flexibility and customizability.

Despite their potential, deploying ViTs on FPGAs for edge inference remains challenging. These models are large and computationally intensive, with memory and energy requirements that far exceed the capabilities of commercial edge platforms [7]. Even compact variants such as DeiT-T [2] require $\sim$ 20MB of weight storage, equivalent to $\sim$ 9100 BRAM18s, surpassing the on-chip memory available on many FPGAs.

To address these limitations, prior works have explored various optimization strategies, including on-demand loading of weights from off-chip memory and quantization-based compression [7], [8], [9]. However, both approaches have trade-offs: frequent off-chip memory access significantly increases energy consumption, while aggressive quantization may degrade model accuracy. More recently, hybrid solutions using quantized models combined with pipelined execution on platforms like the AMD/Xilinx Versal AI Engine have demonstrated high throughput [10], [11], but often at the cost of high power consumption (of the order of 40W), which is unsuitable for battery-operated, power-constrained edge scenarios.

These challenges motivate the exploration of alternative paradigms that simultaneously reduce both model size and compute requirements. Recent work in look-up-table (LUT) and logic-based neural networks, including PolyLUT [12], NeuraLUT [13], Differentiable Weightless Neural Networks (DWNs) [14], LogicNets [15], amigoLUT [16], and treeLUT [17], has shown promise in leveraging compact representations with efficient lookup-based inference. DWNs, in particular, introduced differentiable mechanisms for learning LUTs and their connectivity, enabling the direct training of LUTs in a NN, yielding compact models with massive potential for hardware efficiency. However, such models have thus far been limited to small datasets and simpler tasks, and their performance on standard computer vision benchmarks like

[†]Work done while at IBM T.J. Watson Research Center, Yorktown Heights, USA. [*]Correspondence at shashanknag@utexas.edu, ljohn@ece.utexas.edu

CIFAR-10 remains relatively low [14].

In this paper, we investigate how LUT-based neural layers can be integrated into the architecture of ViTs to make them more suitable for edge deployment, specifically targeting edge-suitable workloads such as CIFAR-10, CIFAR-100 [18] and TinyImageNet [19]. Our goal is to co-design models and hardware architectures that enable real-time inference while reducing both memory footprint and energy consumption.

In vision transformers, multi-head self-attention (MHA) layers are generally used for token mixing, and Multilayer Perceptron (MLP) layers are used for channel mixing [20]. We observe that in typical ViTs, a major fraction of compute and model weights is attributed to the MLP layers, as shown in Fig. 1. While various model optimization techniques with transformers have been explored in the past, the core MLP block has continued to dominate the overall model, and it has been shown that these MLPs are infact the key behind the knowledge learnt in these models [21].

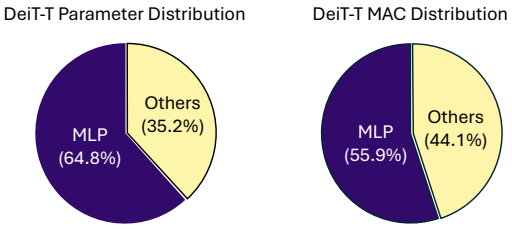

Fig. 1: The MLP layers contribute over 60% of the overall model weights and 55% of the overall multiply-accumulate (MAC) operations in the DeiT-T vision transformer [2].

This observation motivates us to explore a LUT-based neural network design to implement channel mixing in vision transformers, in lieu of the MLP block. Look-up-Table (LUT) or RAM node based neurons, which involve no multiplications, and only look-up operations, offer a much more energy-efficient inference solution than traditional multiply-add neurons [22], [15], [14]. At implementation, the LUT-neurons can be effectively packed into the LUT slices on FPGAs. These neuron implementations are self-contained, and can operate without additional compute or memory resources required for storing weights or to generate outputs – saving BRAMs, DSPs and LUT slices for other compute, thus yielding high energy savings.

We also note that learning LUT-neurons from scratch would lead to smaller and efficient structures than performing a post-training mapping of conventional neurons, or individual multiplications to LUTs [23], [14]. While multiplication free networks that convert multiplications into table lookups have been proposed, each multiplication would end up being converted to one table lookup [24].

In this work, we propose a novel learnable LUT-based channel-mixing block and integrate into a transformer encoder layer(Fig. 2). With this, we propose **LL-ViT** : Learned-LUT based Vision Transformers – an algorithm hardware co-design technique for FPGA-based edge-deployable vision transformers. We offer a class of models optimized for FPGA acceleration,

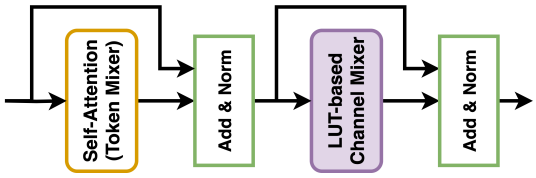

Fig. 2: Proposed Learned-LUT based Vision Transformer (LL-ViT) - overview of a single encoder layer in the model

that incorporate multiplication-free channel mixers in vision transformers (Fig. 2), offering an edge-efficient inference solution that delivers the model performance advantages of conventional vision transformers, while offering hardware performance benefits of LUT neurons. Specifically, we make the following contributions in this paper:

- We propose a novel learnable LUT-based channel-mixer block, and demonstrate its seamless integration into a transformer model.
- We integrate the proposed channel-mixer with a self-attention based token mixer, offering an edge-optimized vision transformer model well-positioned for FPGAs, with reduced computational demand and memory consumption.
- We co-design a low-power accelerator for LL-ViT that avoids off-chip weight movement, and evaluate its performance on FPGA against prior works.
- While logic and LUT-based models have traditionally been applied to small-scale datasets, for the first time, we extend it to vision transformers and demonstrate their performance on relatively complex computer vision tasks. LL-ViT achieves comparable accuracies of 95.5% on CIFAR-10, 78.8% on CIFAR-100, and 60.9% on TinyImageNet, at a smaller model size and computational cost compared to baseline ViTs.

We note that low energy, low latency, and reduced memory requirements with a LUT based implementation of the channel mixer make LL-ViTs excellent candidates for edge inference deployments. LL-ViT offers $1.9\times$ improved energy efficiency and $1.3\times$ lower latency against a quantized baseline FPGA accelerator. With over 60% reduction of model weights, LL-ViT fits completely on a Virtex platform offering a high throughput of 1083 FPS, at a reasonable power consumption of 10.9W.

II. Background and Related Work

A. Vision Transformers and their FPGA acceleration

Vision Transformers (ViTs) are models for computer vision applications with a stack of encoders (Fig. 3), that were inspired by the success of transformer models for language tasks [1], [2], [25]. Over the years, there has been active research in making ViTs more efficient, both in terms of model performance, as well as inference efficiency [26]. However, the core backbone of the encoders in these models is largely similar – a multi-head self-attention block that captures flow of information across tokens (token mixer), and a Multi-Layer Perceptron (MLP) block to cater to the interaction across different feature channels within each token (channel mixer). Within these blocks, the

operations primarily involve matrix multiplications, and non-linear operations including GELU, LayerNorm, among others.

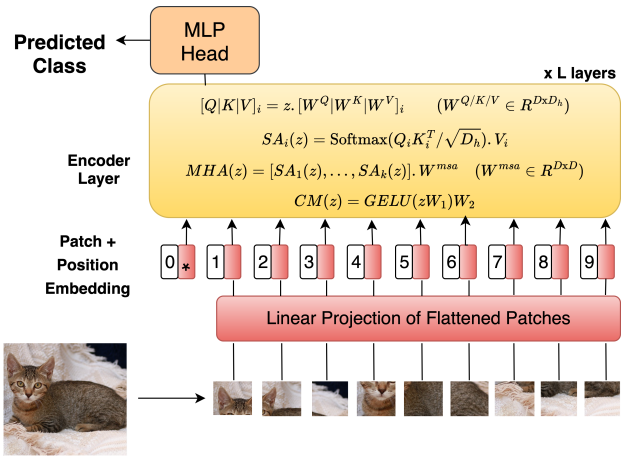

Fig. 3: A typical vision transformer model consisting of a stack of encoder blocks. Figure adapted from [1].

In order to address the issues with increasing model sizes and the computational complexity of ViTs, and the consequent deployment implications, various model optimization techniques have been studied in the past. Particularly, quantization of these models has been investigated, which helps reduce the memory requirement and compute precision. I-ViT [27] and I-BERT [28] quantize weights and activations to 8-bits, and BinaryViT [29] has explored binary-precision quantized designs that are suitable for efficient inference implementations. However, the computational complexity of these models remains unchanged. Compact Vision/ Convolution Transformers [30] proposes a set of compact ViTs with fewer encoder layers, that are well-suited for the small-data and resource-constrained environment regime. SwiftFormer proposes an additive attention mechanism geared for mobile applications, as an alternative to the self-attention mechanism [31].

In recent years, numerous works have explored energy-efficient transformer architectures and their hardware accelerator designs targeting FPGAs. ViTA [7] introduced an edge-optimized accelerator, and ME-ViT [8] developed an accelerator that minimizes memory traffic; but both fetch in weights from off-chip memory atleast once every inference request. ViA [32] proposes an FPGA architecture with an effective partitioning strategy, but involves write backs to memory between the MLP and MHA processing blocks. HeaT-ViT [33] offers an algorithm-hardware co-design solution using token selectors and pruning strategies to reduce the computation. More recent works include SSR [10] and HG-PIPE [11] primarily targeting Versal platforms. While SSR [10] uses a combination of temporal and spatial acceleration to offer higher throughput or latency, HG-PIPE adopts an aggressive quantization (3-bit / 4-bit) and LUT-based compute techniques to ensure that the model weights can reside on-chip. However, both of these works suffer from high power consumption, making them sub-optimal for the edge.

B. Multiplication-free Neural Networks

Of late, there has been a surge of interest in approximate matrix multiplication (AMM) to enhance energy efficiency of model inference. A transformer architecture was proposed that approximates floating-point multiplications with piece-wise affine functions, achieving comparable performance while reducing computational demands primarily for energy savings [34]. LUT-GEMM [35] introduces an AMM method using Look-up-Tables (LUTs) to reduce both energy consumption and latency by modifying GPU kernels. LUT-NN [24] utilizes a centroid-based multiplication approximation technique to replace multiplications with lookup operations. However, these methods primarily rely on LUTs as an inference-time optimization, substituting multiplication operations with lookup operations to achieve post-training speedup. We now turn our attention to an unconventional class of neural networks that employs LUTs as the core computational units or "neurons" of the model, learning them directly during training. This integration of LUTs at the neural architecture level goes beyond inference optimizations, aiming to fully utilize LUT's learning capacity in the training process for improved efficiency.

Learned LUT/ LUT-neuron based Neural Networks: In recent times, LUT-neuron based neural networks have gained significant traction. These neurons (Fig. 4) are efficient to implement, and eliminate the need for power and resource hungry multiply-accumulate (MAC) operations in conventional neurons. LogicNets [15] trained models using quantized linear

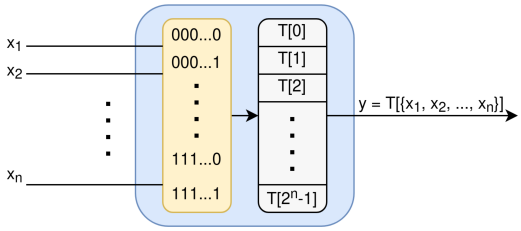

Fig. 4: LUT or RAM-node Neuron : The input sequence is concatenated and used to "look up" the output in the LUT – with no MAC operations involved.

neurons that are sparsely connected across layers – each of these would then be translated to a LUT at implementation. PolyLUT [12] extends this idea, by encapsulating neurons which can learn polynomial functions and then converted to LUTs. NeuraLUT [13] further extends this to support new dense subnetwork architectures encapsulated within each logical-LUT. CompressedLUT [36] and ReducedLUT [37] offer solutions to efficiently compress these logical-LUTs for their hardware implementation using physical-LUTs. TreeLUT [17] trains Gradient Boosted Decision Trees (GBDTs), and converts them to efficient FPGA-based implementations for efficient inference. NullaNet [38] is another work which trains networks with binary activations and floating point weights, enabling layers to be interpreted as boolean functions that can then be reduced for implementation. DiffLogicNets [39] designs differentiable logic-based neural networks, which can be implemented as

TABLE I: Breakdown of computations and model weights in a typical Vision Transformer. n_{layers} represents the number of layers, N represents the number of tokens, and D is the latent dimension of the model. In DeiT-T, $n_{layers} = 12$, $D = 192$, for an input image of size 224×224, with a patch size of 16×16, and one token for the classification head, $N = 197$.

Layer	# Parameters	# MAC ops	DeiT-T : # Parameters	DeiT-T : # MAC ops
Q, K, V Projection	$3 \times n_{layers} \times D \times D$	$3 \times n_{layers} \times N \times D \times D$	1,327,104	261,439,488
$Q.K^T$	-	$n_{layers} \times N \times D \times N$	-	89,415,936
SoftMax.V	-	$n_{layers} \times N \times N \times D$	-	89,415,936
Multi-head concat	$n_{layers} \times D \times D$	$n_{layers} \times N \times D \times D$	442,368	87,146,496
Feed-forward 1 (MLP)	$4 \times n_{layers} \times D \times D$	$4 \times n_{layers} \times N \times D \times D$	1,769,472	348,585,984
Feed-forward 2 (MLP)	$4 \times n_{layers} \times D \times D$	$4 \times n_{layers} \times N \times D \times D$	1,769,472	348,585,984

logic gates (which are effectively LUT2s).

Weightless Neural Networks (WNNs) is another such class of neural networks inspired by the dendritic trees of biological neurons [40], [41]. These networks aims train LUTs directly, based on the idea that an n-input LUT is a highly expressive structure which can represent any one of 2^{2^n} possible non-linear functions. Consequently, training a LUT-neuron directly could be more efficient than using conventional DNN neurons during training, as a LUT-neuron has a higher learning capacity, with a VC dimension of 2^n [42].Recent works such as BTHOWeN [22] and ULEEN [43] have demonstrated single layer models with fewer neurons compared to traditional DNNs, at comparable accuracies. Differentiable Weightless Neural Networks (DWNs) [14], a recent work in this direction, defined Extended Finite Difference (EFD) based gradients for LUT-entries and inputs, and Learnable Mapping technique for learning LUTs connectivity, allowing for multi-layer models of LUT neurons to be directly trained using gradient descent and backpropagation.

All these LUT-neuron based models significantly excel in terms of model size, latency and energy efficiency, as demonstrated on small-scale datasets including JSC [44], MNIST [45] and NID [46]. However, these models perform poorly on image classification tasks such as the CIFAR-10 dataset [18], as these do not implement an architecture that can learn positional independence of features. While these prior works were largely tiny discriminator-based models that directly predicted the maximum likelihood class when presented with an input image, in this work we seek to design LUT-based channel mixers, as intermediate layers within a complex model architecture.

III. LL-ViT: Learned-LUT Vision Transformers

A. Workload Analysis & Motivation

We study a range of vision transformer models to characterize the computations and memory usage in their constituent layers with an aim to identify bottlenecks in model inference efficiency, and summarize the findings in Table I. As illustrated here, and as alluded in Fig. 1, particularly of interest is the fact that across vision transformer models, channel mixers (MLPs) contribute over 60% of the total model weights. At the same time, these blocks also contribute to a significant fraction of the overall compute – ranging between 50-70% of the total MAC (multiply-accumulate) operations. These findings are also corroborated in ViTA [7]. This analysis motivates us to look at alternate layer architectures to implement channel mixing in vision transformers, as an efficient channel mixer would translate to significant overall efficiency improvements.

As compared to MLP-only models, prior works on LUT-neuron based models have demonstrated iso-accuracy performance with significant hardware performance improvements [43], [14]. This suggests that a newly designed LUT-based channel-mixer would be able to learn the patterns originally represented by the MLP block, thus motivating our design. For this work, we choose to adopt a WNN-based approach for the LUT-based channel-mixer design. This is owing to the LUT differentiation techniques proposed in DWN [14], which would enable us to construct a multi-layer LUT-based network with arbitrary connections. The superior performance of DWNs further motivates our choice.

B. LUT-based Channel Mixer

Existing Implementation: The channel mixer transforms the feature channels while treating each token independently. In vision transformers, this is typically implemented as a two-layer MLP with an intermediate GELU non-linearity:

$$\mathbf{CM}(\mathbf{z}) = \text{GELU}(\mathbf{z}\mathbf{W}_1)\mathbf{W}_2 \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^{N \times D}$ is the input token feature (activation) matrix, with N tokens and D feature dimensions. The first linear transformation weight matrix is $\mathbf{W}_1 \in \mathbb{R}^{D \times D_h}$, which expands the feature dimension to D_h. The GELU activation function is applied element-wise after this transformation. Finally, $\mathbf{W}_2 \in \mathbb{R}^{D_h \times D}$ projects the hidden representation back to the original feature dimension.

Design Challenges: As mentioned previously, prior work on LUT-based WNNs have proposed models that directly compute the scores for discriminators corresponding to each class. On the contrary, within the vision transformer architecture, channel mixer is an intermediate block, and hence the proposed LUT-based channel mixer would have to present real-valued outputs for all the feature dimensions – while also ensuring that the output layer is fully-differentiable. To address this, we propose a conditional summation layer following the final layer of LUT neurons, that adds higher precision encoded values based on the output requirements from the block.

Conditional Summation Layer: We propose a conditional summation layer that operates as follows: if the output from

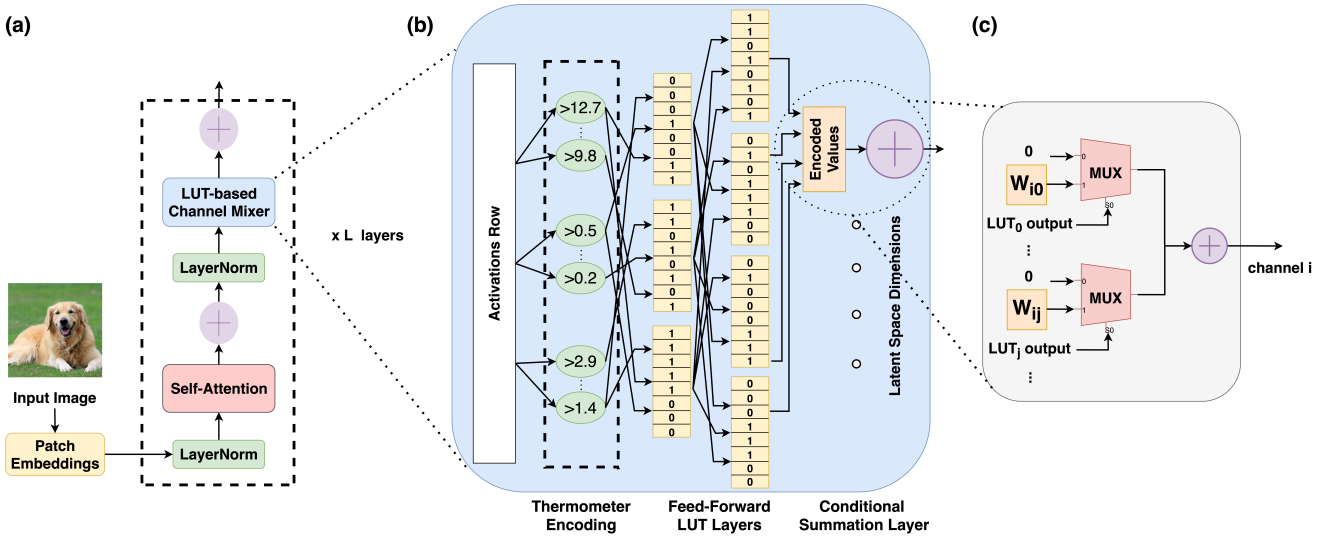

Fig. 5: Proposed Learned LUT-based Vision Transformer Design – (a) Overall Design, (b) a LUT-based Channel Mixer within the encoder block, (c) the conditional summation layer implementation for a particular channel. This is repeated in each encoder.

the LUT neuron in the last layer is a 1, an encoded value is added to the corresponding output, and if it is 0, it is skipped:

$$y_i = \sum_j \begin{cases} W_{ij}, & \text{if } x_j = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where y_i is the output of the i-th channel, x_j is the output of the j-th LUT in the last layer, and W_{ij} represents the corresponding encoded value associated with the j-th LUT for the i-th channel. The summation ensures that only the value W_{ij} corresponding to active LUT outputs ($x_j = 1$) contribute to y_i. Importantly, this layer incurs no multiplication during inference, as it merely adds an encoded value based on which LUTs participate in the summation – while still maintaining the full-precision required by the model. During training, we allow the encoded values (W_{ij}) to be learned with full-precision (fp32) to ensure it is differentiable and has smooth gradients. Post-training quantization is performed on these encoded values to quantize them down to a lower precision as required by the rest of the vision transformer.

Design: Fig. 5 (b) illustrates the proposed LUT-based channel mixer block. As the channel mixer only captures interactions along the channel, and treats each token independently, we flatten input activations row-wise and pass them through these layers one row at a time. We add a thermometer encoding layer [47], [48], [14], which converts the input activations into a sequence of bit representations. This is followed by one or more configurable LUT neuron layers (feed-forward), ending with the conditional summation layer. The output layer contains a number of summation units matching the number of channels in the transformer (latent dimension, D), preserving the network's latent space dimensionality.

Differentiability: As discussed earlier, we employ the Extended Finite Difference (EFD) based technique from DWN [14] for the LUT neurons to be differentiable. The input connections to the LUT neurons are learned using DWN's Learnable Mapping technique. The gradients for the thermometer encoding layers are computed using straight-through estimators (STEs) [49], as in Binary Neural Networks (BNNs) [50], with thermometer-thresholds serving as STE-thresholds. Further, the conditional summation layer is represented as a matrix mutiplication during training, making it differentiable as well. This ensures that the proposed multiplication-free LUT-based channel mixer is fully learnable, differentiable and compatible with the vision transformer model architecture.

C. Integrated Learned LUT Vision Transformer

As the token mixer block is not dominated by weights, and predominantly involves finding relations amongst the tokens, a LUT-based neural layer for it would not yield significant returns. As such, we stick to existing multi-head self-attention based token mixer blocks, and combine them with our proposed channel mixer block to deliver LL-ViT. In doing so, we leverage the LUT neurons' hardware and energy savings advantages combined with the ability to learn positional invariance through the self-attention layers of the transformers. We note that we integrate the proposed channel mixer into each encoder in the vision transformer. With the gradients well defined for the LUTs and the conditional summation layer, we ensure that the overall model is fully-differentiable, and the entire LL-ViT is trained end-to-end by backpropagating the losses from the output – similar to how a regular vision transformer is trained. Fig. 5 (a) shows an overview of the proposed design.

D. FPGA Acceleration

Channel Mixer PE Design: For the LL-ViT accelerator design, we propose a dedicated processing element (PE) block design for the LUT-based channel mixer, shown in Fig. 6(b). As the connections between two layers could be any arbitrarily learnt mapping, all output bits from the previous layer would be

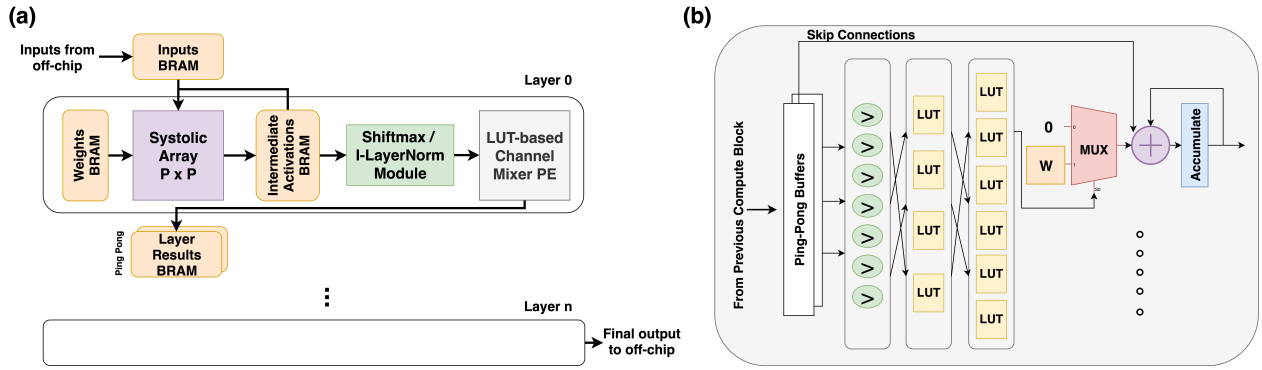

Fig. 6: Hardware Accelerator Design for LL-ViT– (a) the overall design (b) the PE design for the LUT-based channel mixer.

required to form input addresses for the next layer. Hence, to avoid stalls, we design this PE to be parallely processing all elements along the channel-dimension for a given row of input activations, with subsequent rows processed through the PE in a pipelined fashion. Ping-pong buffers at the input of the PE are used to stage the row of input activations currently being processed in one buffer, while the other buffer is simultaneously filled with the next row. The thermometer encodings are realized using comparator blocks, and the LUT neurons in the feed-forward LUT layers are implemented using the LUTs within the logic slices of the FPGA. For the conditional summation layer, we introduce an adder circuit preceded by a 2:1 MUX for each dimension in the latent space. For the specific channel, the MUX and adder circuit iteratively adds the contributions from all the LUTs in the previous layer. Note that while we could also introduce an adder tree for each of these for low-latency, this would increase the resource consumption quite significantly. Furthermore, we find that the overall throughput of the network would be bottlenecked by other slower layers, and hence this iterative approach allows us to time-balance the channel mixer stage with the other stages of the network, while conserving resources. This adder circuit also accumulates the skip connections from the ping-pong buffer.

Overall Design: Fig. 6(a) shows the overall LL-ViT accelerator design. To offer high throughput, we design dedicated blocks for implementing each layer in the encoder stack, and multiple frames are pipelined across these layers. The inputs are written into the first layer, and outputs are read back from the last layer. All model weights are read and stored in dedicated BRAMs at initialization, and remain stationary on-chip. The matrix multiplications in the MHA token-mixer are catered to by a $P \times P$ systolic array, as used in literature typically [51]. This is interfaced with the PE block designed for the channel mixer. Dedicated blocks are implemented for computing the integer-based non-linear operations, namely, I-LayerNorm and ShiftMax, as proposed in I-ViT [27].

As mentioned in Sec. I, an important aspect to note here is that by introducing LUT-based layers instead of conventional MLP layers, we are not trading off compute with additional memory. On FPGAs, compute blocks are majorly built up of DSPs and LUTs. Consequently, any processing element for conventional neural network acceleration gets mapped to these logic elements on device. In our proposed model, the LUTs in the channel mixer would get directly mapped to these LUTs on device, incurring no additional overhead on the memory. Infact on the contrary, the LUT-based implementation eliminates the memory requirement associated with those weights that would have been present in a conventional layer; while also eliminating the associated off-chip memory access.

IV. Evaluation

A. Experimental Setup

Model Backbone & Datasets: As the goal of our work is to develop an edge-optimized deployment solution for vision transformers, we consider I-ViT-T [27], an **INT8 quantized** model as our **baseline**. To evaluate the performance of our proposed channel mixer and model architecture changes, we consider four datasets for evaluation – CIFAR-10 [18], CIFAR-100, Flowers-102 [52] and TinyImageNet [19]. We particularly choose these medium-sized datasets considering that we are optimizing these models for resource-constrained environments, and inline with prior works on LUT-neuron based networks and edge efficient models [14], [53]. I-ViT-T is a fully-quantized INT8 precision model, and designing LL-ViT with this backbone allows us to gauge our design's benefits in optimizing an already quantized model. For consistency with other ViTs, we resize the images in these datasets to 224×224 and apply the same data augmentations used in DeiT [2].

Channel Mixer Configuration: As described in Section III-B, the channel mixer allows the following configurations to be specified – the number of bits of thermometer encoding, the number of layers of LUTs, the number of LUTs in each layer, and the output precision for the encoded values. The number of inputs, and the output channel dimensions would be determined by the model backbone to which the channel mixer is being integrated. The configurable parameters are indicative of the complexity of the channel mixer, and these can be scaled based on the baseline vision transformer variant it is being integrated to. With our experimental observations, we found a two LUT-layer channel mixer configuration, with $(768, 192)$ LUTs in the layers respectively, with a 8-bit thermometer encoding to

TABLE II: Inference performance comparison of LL-ViT against I-ViT-T, a fully quantized ViT baseline on the target FPGA. On CIFAR-10, CIFAR-100, Tiny-ImageNet and Flowers-102 datasets, LL-ViT achieves similar accuracies as the baseline.

Work	CIFAR-10	CIFAR-100	Tiny-ImageNet	Flowers-102	Model Size	Inference Latency	Energy/ Inference
I-ViT-T (baseline, int8)	95.4%	79.2%	60.4%	91.3%	5.06 MB	6.93 ms	4.05 mJ
LL-ViT (ours, int8)	95.5%	78.8%	60.9%	91.6%	1.93 MB	5.33 ms	2.14 mJ

be optimal for the I-ViT-T model backbone. Further, **post-training**, the **encoded values** in the conditional summation layer were **quantized to 4-bits**. The parameters were chosen using the configurations of the original MLP in the baseline as a guidance, with effectively the same number of LUT neurons being used in each layer as in the MLP.

Training Methodology: We define the LUT-based channel mixers using custom PyTorch classes, and integrate this with the MHA based token mixer and train the resultant LL-ViT. We use the default train-test split that these datasets come with, train these models on a A100 GPU, and primarily use the same learning rate, scheduler and optimizer as the baseline.

FPGA Evaluation: To evaluate the inference efficiency, we generate SystemVerilog RTL [54] for the accelerator design, with the RTL code for the PE generated from the trained model using custom mako scripts [55]. For our evaluation, we consider the systolic array to be of dimensions $P \times P = 32 \times 32$. For the baseline I-ViT-T model acceleration, we consider a pipelined 32×32 systolic array for the MLP and MHA blocks. We synthesize our design on AMD Xilinx xcvu9p-flgb2104-2-i (Virtex series) FPGA, with a target clock frequency of 200 MHz using Vivado Design Suite. For power and energy estimation, we use a default switching activity factor of 12.5% and consider the total device power using AMD Power Design Manager (PDM) [56]. We evaluate the overall model's performance in terms of latency, and energy consumed per image inference.

B. Results

Model Accuracy: We trained our model, and performed a post-training quantization on the encoded value in the conditional summation layers to int4 precision. As noted in Table II, with CIFAR-10, LL-ViT achieves an accuracy of 95.5% against the baseline (I-ViT-T) model accuracy of 95.4%, 78.8% accuracy on CIFAR-100, 60.9% on Tiny-ImageNet, and 91.3% on Flowers-102 dataset, with a 60% smaller model size. These findings suggest that LL-ViT performs comparably to the baseline in terms of model accuracy. Table III compares our model's performance in terms of accuracy, parameter count, and number of MAC operations against other common iso-accuracy vision transformer models. LL-ViT achieves better accuracy than most other works, while involving fewer parameters and less than 50% MAC operations compared to the baseline. We also note that LL-ViT performs much better compared to prior works like FINN, and LUT-neuron based works, that report accuracy on CIFAR-10 in the range of 40-88% (Table IV). By incorporating a vision transformer architecture, LL-ViT offers a solution with improved accuracy on the accuracy-energy pareto front – enabling LUT-based models to be useful for accuracy-critical applications.

TABLE III: Comparison of LL-ViT against optimized ViTs.

Work	CIFAR-10 Top 1 Accuracy	#Params (M)	#GMACs/ inference
CCT-2/3x2 [30]	89.7%	0.28	0.04
CCT-7/3x2 [30]	95.0%	3.85	0.29
CCT-7/3x1 [30]	96.5%	3.76	1.19
DeiT-T [2]	94.8%	5.3	1.31
I-ViT-T [27] (baseline)	95.4%	5.3	1.31
LL-ViT (ours)	95.5%	2.5	0.65

TABLE IV: Performance of LL-ViT in the context of prior quantized / LUT-based models – none of these implement a ViT architecture, and the accuracies are considerably low. Aided by a ViT model backbone, LL-ViT offers improved accuracy trading off model-size and energy consumption.

Work	CIFAR-10 Top 1 Accuracy	Param Size (KiB)	LUTs (K)	Energy/ inf (mJ)
TreeLUT [17]	42.9%	–	2.2	–
DiffLogicNet [39]	57.3%	250	283.3	–
DWN [14]	57.5%	23.4	45.7	0.004
LogicTreeNet-G [57]	86.29%	21183.7	–	–
FINN [53]	80.1%	183.1	46.3	0.15
FINN-R [58]	88.63%	376.5	25.8	-
H-WNN (L) [59]	88.23%	306.5	20.7	-
LL-ViT (ours)	95.5%	1976.3	587	2.14

Optimizing the Optimized: We also demonstrate how our proposed design further optimizes the tiniest variant of Compact Convolutional Transformers [30] – a model that is already optimized for small size, and well-suited for resource-constrained environments. With a LL-ViT design built on the CCT-2/3x2 backbone, we achieve an accuracy of 87.4% on the CIFAR-10 dataset while still reducing MAC operations by about $1.5\times$.

Energy Efficiency: We evaluate the energy consumption per image inference of our design on the target device. As shown in Table II, LL-ViT achieves a $1.9\times$ improvement in energy efficiency (energy per inference) over a fully-quantized baseline accelerator implemented on the same FPGA. Note that this is a conservative estimate, as it does not account for the additional energy spent in the baseline implementation to fetch MLP weights that cannot be accommodated on-chip.

Latency: As noted in Section III-D, our Processing Element is designed such that it avoids stalls due to the varied computations in the token-mixer and channel-mixer blocks. As a consequence, our design is not fully optimized for latency, and we primarily target energy efficiency. Our analysis shows that we still achieve a $1.3\times$ improvement in latency by virtue of our optimized channel mixer, as reported in Table II.

TABLE V: Comparison of our work against other prior works for IViT-T or equivalent models. (* = approximated/ not reported)

Work	Model	Platform	DSPs	LUTs (K)	FFs (K)	BRAM36s	FPS	Power(W)	Efficiency (FPS/W)
ViTA (ISCAS 23) [7]	DeiT-T, A8W8	Pynq Z1	0	22.8	5	*	19	0.88	21.60
HG-PIPE (ICCAD 24) [11]	DeiT-T, A3W3	VCK190	312	669	*	1006	7118	46.7	152.42
HeaT-ViT (HPCA 23) [33]	DeiT-T, A8W8	ZCU102	1968	137	126	355	183	9.45	19.40
Huang et al. (TCAS-I) [60]	ViT-T, A8W8	ZCU102	1268	114	168	648	245	29.6	8.27
SSR (FPGA 24) [10]	DeiT-T, A8W8	VCK190	14405*	619	849	1456	4545	46	98.8
ME-ViT (HiPC 24) [8]	DeiT-T, *	Alveo200	1024	192	132	288	352	9.3	37.87
LL-ViT (ours)	I-ViT-T, A8W8	Virtex	0	589	229	1425	1083	10.9	99.35

Comparison with prior ViT accelerators: Table V compares the FPGA acceleration of LL-ViT against other prior works on ViT accelerators. For comparable resource usage, we use systolic arrays of 16×16 dimensions, and we pick quantized I-ViT-T (or equivalently sized model configurations) with 224 × 224 sized images for fairness in evaluation. As seen, with a low power edge, our approach achieves superior efficiency (FPS/W) and a high FPS compared to most prior works. HG-PIPE [11], the only exception, consumes 4× the power as LL-ViT. We note that some of these works, including HG-PIPE, are aggressively quantized (3-bit) – we demonstrate superior performance even with a conservative 8-bit quantization scheme for our design. With our optimizations, LL-ViT fits within the on-chip memory. Further, we note that our approach is better than baseline model inference on a Jetson-TX2 GPU, which offers an efficiency of about 7FPS/W at 12W [33].

Comparison with other Multiplication-free works: In Fig. 7, we evaluate our LUT-based channel mixer's efficacy against some popular multiplication-free and approximate matrix-multiplication works that have been gathering traction of late; including Affine [34] and LUT-NN [24], in terms of the computations involved. For a channel mixer with the same number of neurons, while these works eliminate multiplications over the baseline, they do it at the cost of increased reads or adds, as each multiplication is replaced by an equivalent add or look-up operation. On the contrary, with learned LUTs, LL-ViT achieves similar model performance while reducing the number of lookup and add operations by a significant factor – demonstrating the efficacy of learning LUTs while training.

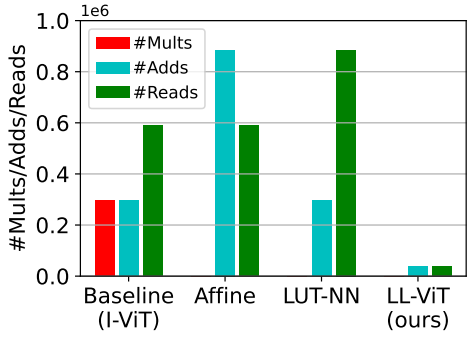

Fig. 7: Comparison of LL-ViT against multiplication-free designs in terms of the number of multiplications, additions, and read operations in the channel mixer block.

C. Sensitivity Studies

Quantized Baselines: We consider various quantization schemes for the baseline vision transformer. As shown in Fig. 8, LL-ViT designed with the quantized backbone consistently demonstrates an energy efficiency of over 1.5× regardless of the quantization scheme used (8-bit/ 4-bit/ 2-bit).

Scaling Trends: We also show that energy efficiency offered by LL-ViT scales comparably with the latent dimensions of the network (Fig. 9(a)) and the image size (Fig. 9(b)). This suggests that as we continue to scale LL-ViTs to larger or smaller models, they would consistently offer the energy savings reported.

Layerwise Energy Breakdown: Table VI indicates the breakdown of energy consumption in a single encoder block of LL-ViT against a fully-quantized baseline I-ViT design. Here, a 32x32 systolic array is used for the baseline model and for the non-weightless layers in LL-ViT. We ignore the energy consumption of the other components including SoftMax, GELU and LayerNorm for clarity in this analysis, as these were found to be minimal. This clearly indicates that LL-ViT eliminates the most energy intensive blocks of the vision transformer (MLP), and introduces an alternate energy-efficient channel mixer block.

Encoded Value Post-training Quantization: As mentioned in Sec. III-B, the encoded value in the channel mixer can be quantized post-training to match the precision desired by the rest of the network. We observe the overall accuracy of LL-ViT to be 95.6% at int8 quantization, 95.5% at int4 quantization, and 90.3% at int2 quantization. As int4 quantization offers the optimal tradeoff, we stick to it for all our evaluations.

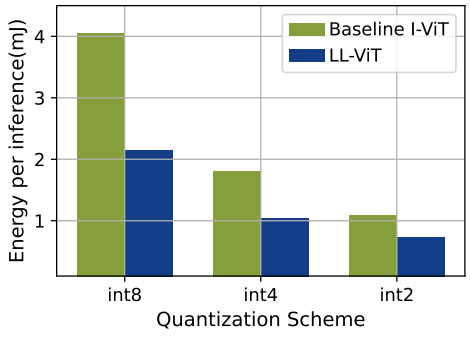

Fig. 8: Energy per inference of baseline I-ViT vs. LL-ViT for varying quantization schemes.

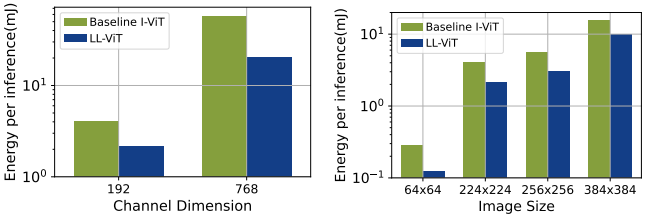

Fig. 9: (a) Energy per inference of the baseline I-ViT vs. LL-ViT with varying latent dimensions (with the same number of neurons in both models). (b) Energy per inference of the baseline I-ViT vs. LL-ViT with varying image sizes, with the model configuration remaining the same.

TABLE VI: Comparing the layerwise breakdown of energy consumption between the baseline and LL-ViT for a single encoder layer, per sample inference.

Stage	Baseline I-ViT-T	LL-ViT
Q, K, V	23.59 uJ (each)	23.59 uJ (each)
QK^T	27.52 uJ	27.52 uJ
$S.V$	27.52 uJ	27.52 uJ
Multi-Head Concat	23.59 uJ	23.59 uJ
MLP Dense 1	94.35 uJ	28.8 uJ
MLP Dense 2	94.35 uJ	
Total	**338.10 uJ**	**178.2 uJ**

V. Discussion

While we primarily demonstrate our work with a tiny ViT baseline (I-ViT-T) as we specifically target edge applications of small tasks and datasets, the technique proposed with LL-ViT could also be applied to larger variants of vision transformers. The hardware performance improvements over the baseline, would be similar for any vision transformer model, considering the structural composition of the accelerator, as shown in the case of CCT. Similarly, we also note that while our primary baseline was INT8 quantized, the proposed technique can be integrated with complementary works on aggressive quantization like BinaryViT [29], and still achieve performance improvement, as alluded in Fig. 8. We view this work as a stepping stone towards a class of learnable LUT-based tiny transformer models that are competitive to the current energy-inefficient transformers.

We also note that the LUT-based Channel Mixer is not merely trying to approximate the MLP; it instead serves as a fundamentally different, yet complete learning primitive. Both MLPs and LUT-based layers are universal function approximators, and the LUT-based channel mixer might end up learning an alternate efficient representation. In our experiments, we observe a low cosine similarity of 0.1 between the outputs of the MLP (in the baseline I-ViT) and the LUT-based channel mixer (in LL-ViT). Nevertheless, LL-ViT achieves superior overall accuracy compared to I-ViT.

VI. Conclusion

In this work we introduce Learned-LUT based Vision Transformers (LL-ViT), an effort to develop edge-efficient vision transformers targeted for FPGA acceleration. We identify an opportunity to reduce computational and memory demands by targeting channel mixers, replace these with the proposed LUT-based channel mixers, and design an accelerator for the model. Based on our model and performance evaluations, we report $1.9\times$ energy efficiency, and $1.3\times$ lower latency against quantized baselines at comparable model accuracies. We also reduce the model size by 60%, enabling the remaining model weights to be fully stationary on-chip. These results illustrate that LL-ViTs are well-positioned as a promising lightweight alternative to traditional vision transformers – paving the way for tiny ViTs deployable at the edge. Although LUT-neuron based models have been applied to many small-scale problems in the past, to the best of our knowledge, this is the first time that its usefulness in constructing a vision transformer model has been demonstrated.

Acknowledgements

This research was supported by Semiconductor Research Corporation (SRC) Task 3148.001, National Science Foundation (NSF) Grants #2326894, #2425655 (supported in part by the federal agency and Intel, Micron, Samsung, and Ericsson through the FuSe2 program), NVIDIA Applied Research Accelerator Program, and compute resources on the Vista GPU cluster through CGAI & TACC at UT Austin. Any opinions, findings, conclusions, or recommendations are those of the authors and not of the funding agencies.

References

[1] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[2] H. Touvron *et al.*, "Training data-efficient image transformers & distillation through attention," 2021.

[3] H. Thisanke *et al.*, "Semantic segmentation using vision transformers: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2305.03273

[4] Meta AI, "Imagenet image classification leaderboard," *Papers with Code*, 2024. [Online]. Available: https://paperswithcode.com/sota/image-classification-on-imagenet

[5] A. Ramesh *et al.*, "Zero-shot text-to-image generation," 2021. [Online]. Available: https://arxiv.org/abs/2102.12092

[6] T. S. Group, "Edge-ai market analysis: Applications, processors and ecosystem guide," 2025. [Online]. Available: https://theshdgroup.com/wp-content/uploads/2025/04/Edge-AI-2025-Report-Final.pdf

[7] S. Nag *et al.*, "Vita: A vision transformer inference accelerator for edge applications," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2023. [Online]. Available: http://dx.doi.org/10.1109/ISCAS46773.2023.10181988

[8] K. Marino *et al.*, "Me-vit: A single-load memory-efficient fpga accelerator for vision transformers," 2024. [Online]. Available: https://arxiv.org/abs/2402.09709

[9] Z. Li *et al.*, "Auto-vit-acc: An fpga-aware automatic acceleration framework for vision transformer with mixed-scheme quantization," in *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)*, 2022, pp. 109–116.

[10] J. Zhuang *et al.*, "Ssr: Spatial sequential hybrid architecture for latency throughput tradeoff in transformer acceleration," in *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 55–66. [Online]. Available: https://doi.org/10.1145/3626202.3637569

[11] Q. Guo *et al.*, "Hg-pipe: Vision transformer acceleration with hybrid-grained pipeline," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '24. New York, NY, USA: Association for Computing Machinery, 2025. [Online]. Available: https://doi.org/10.1145/3676536.3676681

[12] M. Andronic *et al.*, "Polylut: Learning piecewise polynomials for ultra-low latency fpga lut-based inference," in *2023 International Conference on Field Programmable Technology (ICFPT)*. IEEE, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1109/ICFPT59805.2023.00012

[13] M. Andronic *et al.*, "Neuralut: Hiding neural network density in boolean synthesizable functions," IEEE, pp. 140–148, 2024.

[14] A. T. L. Bacellar *et al.*, "Differentiable weightless neural networks," *2024 41st International Conference on Machine Learning (ICML)*, 2024. [Online]. Available: https://icml.cc/virtual/2024/poster/34511

[15] Y. Umuroglu *et al.*, "Logicnets: Co-designed neural networks and circuits for extreme-throughput applications," *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 291–297, 2020.

[16] O. Weng *et al.*, "Greater than the sum of its luts: Scaling up lut-based neural networks with amigolut," in *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 25–35. [Online]. Available: https://doi.org/10.1145/3706628.3708874

[17] A. Khataei *et al.*, "Treelut: An efficient alternative to deep neural networks for inference acceleration using gradient boosted decision trees," in *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '25, 2025.

[18] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[19] Y. Le *et al.*, "Tiny imagenet visual recognition challenge," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:16664790

[20] D. Sridhar *et al.*, "Scheme: Scalable channel mixer for vision transformers," 2024. [Online]. Available: https://arxiv.org/abs/2312.00412

[21] M. Geva *et al.*, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg *et al.*, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. [Online]. Available: https://aclanthology.org/2022.emnlp-main.3

[22] Z. Susskind *et al.*, "Weightless neural networks for efficient edge inference," in *31st International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2022.

[23] S. Nag *et al.*, "Logicnets vs. uleen : Comparing two novel high throughput edge ml inference techniques on fpga," in *2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2024, pp. 1206–1211.

[24] X. Tang *et al.*, "Lut-nn: Empower efficient neural network inference with centroid learning and table lookup," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3613285

[25] X. Zhai *et al.*, "Scaling vision transformers," 2022. [Online]. Available: https://arxiv.org/abs/2106.04560

[26] L. Papa *et al.*, "A survey on efficient vision transformers: Algorithms, techniques, and performance benchmarking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, p. 7682–7700, Dec. 2024. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2024.3392941

[27] Z. Li *et al.*, "I-vit: Integer-only quantization for efficient vision transformer inference," 2023.

[28] S. Kim *et al.*, "I-bert: Integer-only bert quantization," *International Conference on Machine Learning (Accepted)*, 2021.

[29] P.-H. C. Le *et al.*, "Binaryvit: Pushing binary vision transformers towards convolutional models," 2023.

[30] A. Hassani *et al.*, "Escaping the big data paradigm with compact transformers," 2021. [Online]. Available: https://arxiv.org/abs/2104.05704

[31] A. Shaker *et al.*, "Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[32] T. Wang *et al.*, "Via: A novel vision-transformer accelerator based on fpga," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4088–4099, 2022.

[33] P. Dong *et al.*, "Heatvit: Hardware-efficient adaptive token pruning for vision transformers," 02 2023, pp. 442–455.

[34] A. Kosson *et al.*, "Multiplication-free transformer training via piecewise affine operations," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2024.

[35] G. Park *et al.*, "Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models," 2023.

[36] A. Khataei *et al.*, "Compressedlut: An open source tool for lossless compression of lookup tables for function evaluation and beyond," in *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 2–11. [Online]. Available: https://doi.org/10.1145/3626202.3637575

[37] O. Cassidy *et al.*, "Reducedlut: Table decomposition with "don't care" conditions," in *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 36–42. [Online]. Available: https://doi.org/10.1145/3706628.3708823

[38] M. Nazemi *et al.*, "Nullanet tiny: Ultra-low-latency dnn inference through fixed-function combinational logic," in *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2021, pp. 266–267.

[39] F. Petersen *et al.*, "Deep differentiable logic gate networks," 2022. [Online]. Available: https://arxiv.org/abs/2210.08277

[40] I. Aleksander *et al.*, "Wisard· a radical step forward in image recognition," *Sensor review*, vol. 4, no. 3, pp. 120–124, 1984.

[41] V. C. Ferreira *et al.*, "A feasible FPGA weightless neural accelerator," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[42] H. C. C. Carneiro *et al.*, "The exact vc dimension of the wisard n-tuple classifier," *Neural Computation*, vol. 31, pp. 176–207, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:53715711

[43] Z. Susskind *et al.*, "Uleen: A novel architecture for ultra-low-energy edge neural networks," *ACM Trans. Archit. Code Optim.*, vol. 20, no. 4, dec 2023. [Online]. Available: https://doi.org/10.1145/3629522

[44] J. Duarte *et al.*, "Fast inference of deep neural networks in fpgas for particle physics," *Journal of Instrumentation*, vol. 13, no. 07, p. P07027–P07027, Jul. 2018. [Online]. Available: http://dx.doi.org/10.1088/1748-0221/13/07/P07027

[45] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[46] N. Moustafa *et al.*, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[47] H. Carneiro *et al.*, "Multilingual part-of-speech tagging with weightless neural networks," *Neural Networks*, vol. 66, 03 2015.

[48] A. Bacellar *et al.*, "Distributive thermometer: A new unary encoding for weightless neural networks," in *ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 01 2022, pp. 31–36.

[49] Y. Bengio *et al.*, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[50] M. Courbariaux *et al.*, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *Neural Information Processing Systems*, 2016.

[51] R. Xu *et al.*, "A survey of design and optimization for systolic array-based dnn accelerators," *ACM Comput. Surv.*, vol. 56, no. 1, aug 2023. [Online]. Available: https://doi.org/10.1145/3604802

[52] M.-E. Nilsback *et al.*, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, 2008, pp. 722–729.

[53] Y. Umuroglu *et al.*, "Finn: A framework for fast, scalable binarized neural network inference," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '17. ACM, Feb. 2017. [Online]. Available: http://dx.doi.org/10.1145/3020078.3021744

[54] "Ieee standard for systemverilog–unified hardware design, specification, and verification language," *IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012)*, pp. 1–1315, 2018.

[55] M. Bayer, "Mako templates for python," 2021. [Online]. Available: https://www.makotemplates.org/
[56] "AMD Power Design Manager (PDM)," Online (AMD Documentation), 2025, next-generation power estimation platform for AMD Versal™, UltraScale+, and Kria™ SOMs. [Online]. Available: https://www.amd.com/products/software/adaptive-socs-and-fpgas/power-design-manager.html
[57] F. Petersen *et al.*, "Convolutional differentiable logic gate networks," *Advances in Neural Information Processing Systems*, vol. 37, pp. 121 185–121 203, 2024.
[58] M. Blott *et al.*, "Finn-r: An end-to-end deep-learning framework for fast exploration of quantized neural networks," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3242897
[59] M. Jadhao *et al.*, "Hybrid weightless neural networks for efficient edge inference," in *2025 35th International Conference on Field-Programmable Logic and Applications (FPL)*. IEEE, 2025.
[60] M. Huang *et al.*, "An integer-only and group-vector systolic accelerator for efficiently mapping vision transformer on edge," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. PP, pp. 1–13, 12 2023.