IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 January 2025, accepted 3 February 2025, date of publication 10 February 2025, date of current version 13 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3540399

RESEARCH ARTICLE

Optimizing Vision Transformers: Unveiling 'Focus and Forget' for Enhanced Computational Efficiency

BANAFSHEH SABER LATIBARI, HOUMAN HOMAYOUN, AND AVESTA SASAN, (Senior Member, IEEE)
Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616, USA
Corresponding author: Banafsheh Saber Latibari (bsaberlatibari@ucadvis.edu)

This work was supported by the National Science Foundation under Award 2233893.

ABSTRACT Vision Transformers are renowned for their accuracy in computer vision tasks but are computationally and memory expensive, making them challenging to deploy on resource-constrained edge devices. In our research paper, we introduce a revolutionary approach to designing energy-aware dynamically prunable Vision Transformers for use in edge applications. Our solution denoted as Incremental Resolution Enhancing Transformer (IRET), works by the sequential sampling of the input image. However, in our case, the embedding size of input tokens is considerably smaller than prior-art solutions. This embedding is used in the first few layers of the IRET vision transformer until a reliable attention matrix is formed. Then the attention matrix is used to sample additional information using a learnable 2D lifting scheme only for important tokens and IRET drops the tokens receiving low attention scores. Hence, as the model pays more attention to a subset of tokens for its task, its focus and resolution also increase. This incremental attention-guided sampling of input and dropping of unattended tokens allow IRET to significantly prune its computation tree on demand. By controlling the threshold for dropping unattended tokens and increasing the focus of attended ones, we can train a model that dynamically trades off complexity for accuracy. Moreover, using early exiting our model is capable of doing anytime prediction. This is especially useful for real-word energy-sensitive edge devices, where accuracy and complexity could be dynamically traded based on factors such as battery life, reliability, etc.

INDEX TERMS Computer vision, deep learning, pruning, vision transformer.

I. INTRODUCTION

Recent advancements in deep learning and GPU capabilities [12] have significantly improved computer vision's detection and prediction. A major innovation is the use of transformer models, first for Natural Language Processing (NLP) in 2017 and later for visual tasks [7]. Visual transformers, especially those developed by Google Brain in 2020, have outperformed traditional CNNs in accuracy, especially with large datasets. However, their high computational and memory requirements pose challenges for edge device deployment [32], [78], primarily due to their reliance on complex global attention mechanisms and MLPs. To mitigate these demands, various strategies like multi-scale processing, token dropping, early prediction, softmax elimination, and efficient attention approaches have been researched. These approaches are summarized in Section III. While these solutions address certain aspects of the computational challenges, they fall short of fully optimizing context-aware computation. The main contributions of this paper are as follows:

1) This paper introduces a novel context-aware approximation technique for dynamic pruning of computational trees in transformer models, diverging significantly from existing methods. We identify an underutilized potential in transformers for context-based approximation, which we argue can greatly

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

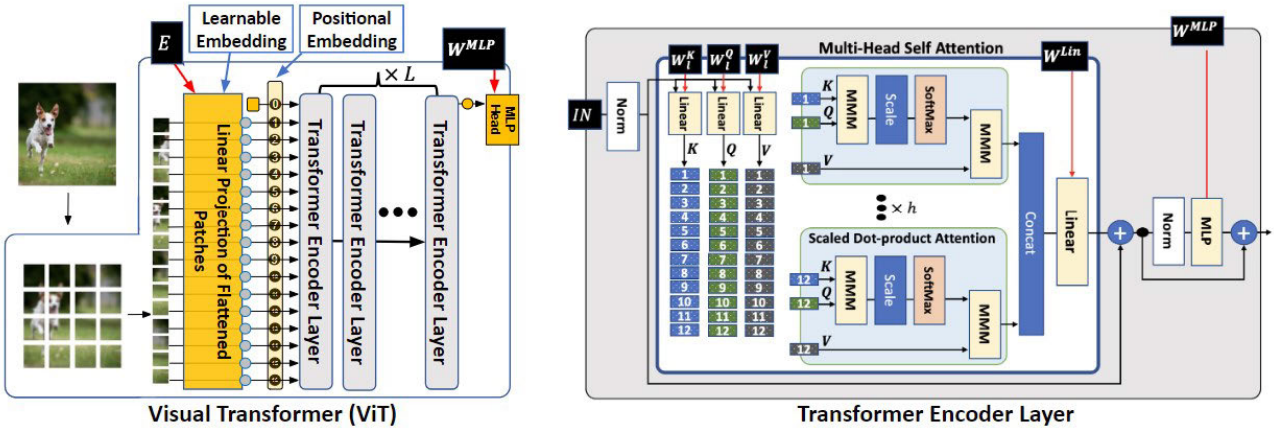


FIGURE 1. (Left): Overall structure of original Visual Transformer (ViT) in [10]. (right): Encoder solution used in ViT, illustrating the implementation details of Multi-Head Self Attention (MSA) from *h* scaled dot-product attention units.

enhance their efficiency with minimal accuracy impact, broadening their application scope.

2) We present the Incremental Resolution Enhancing Transformer (IRET), a transformative model architecture that employs attention-based input sampling.
3) Utilizing learnable 2D lifting schemes, IRET processes three input samples incrementally, thereby building contextual awareness early. This architecture allows IRET to use temporal attention scores for two key functions: a) **forget**: discarding unattended tokens, and b) **focus**: selectively enhancing the embedding size of attended tokens by merging existing features with new ones from a 2D lifting scheme output.

This approach mirrors human visual perception, starting with a broad context understanding and then focusing on more pertinent image aspects. IRET thus uses minimal information initially for context comprehension, subsequently concentrating on key image tokens through incremental sampling while ignoring less relevant ones. The remainder of the paper is structured as follows: Section II covers background information. Section III reviews related work. Section IV details the IRET architecture. Section V presents experimental evaluations. Finally, Section VI concludes the paper.

II. BACKGROUND

Fig. 1. (left) shows the Visual Transformer (ViT) [10] architecture, and Fig. 1. (right) captures the structure of its encoder layer. In ViT the input image is split into fixed-size patches by reshaping the image $x \in R^{H\times W\times C}$ into a sequence of flattened 2D patches $x_p \in R^{N\times(P^2.C)}$. The (H, W) is the image resolution, C is the number of channels, (P, P) is the image patch resolution, and $N = HW/P^2$ is the number of patches. The attention mechanism used in the encoder is scaled dot-product attention suggested in [56]. The inputs are queries Q and keys K of dimension d_k, and values V of dimension d_v. The encoder is designed to linearly project the queries, keys, and values h times with different learned linear projections to d_k, d_k, and d_v dimensions, respectively. As shown in Fig. 1(right), each encoder layer uses h scaled dot-product attention heads. Scaled dot-product attention heads compute the matrix in Eq. 1 yielding d_v-dimensional output values that are later concatenated and projected. The Multi-Head Self Attention (MSA), the function of which is captured in Eq. 2, allows the model to jointly attend to information from different representation subspaces at different positions. Similar to BERT's class token [9], ViT prepends a learnable embedding to embedded patches ($z_0^0 = x_{class}$), whose state at the output of the encoder (z_L^0) serves as the image representation y. Layernorm (LN) is applied before and residual connections after every block.

$$Attention(Q, K, V) = Softmax(QK^T/\sqrt{d_k})V \quad (1)$$

$$MSA(Q, K, V) = Concat(head_i, \ldots, head_h)W^O, \quad (2)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

The Visual transformer function is captured using equations 4 through 7:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{pos}, \quad E \in R^{(P^2.C)} \times D, E_{pos} \quad (4)$$

$$z_l' = MSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1 \ldots L \quad (5)$$

$$z_l = MLP(LN(z_l')) + z_l', \quad l = 1 \ldots L \quad (6)$$

$$y = LN(z_l^0) \quad (7)$$

The classification head is attached to z_L^0 and implemented by an MLP with one hidden layer at pre-training and one linear layer at fine-tuning. 1-Dimensional Position embedding is added to the patch embeddings to retain positional information. In a similar vein, DETR [4] exploits a pure transformer to create an end-to-end object detection framework. Taking a different approach, DeiT [55] enhances ViT by introducing the distillation token, and leverages a teacher model to decrease the necessary training data.

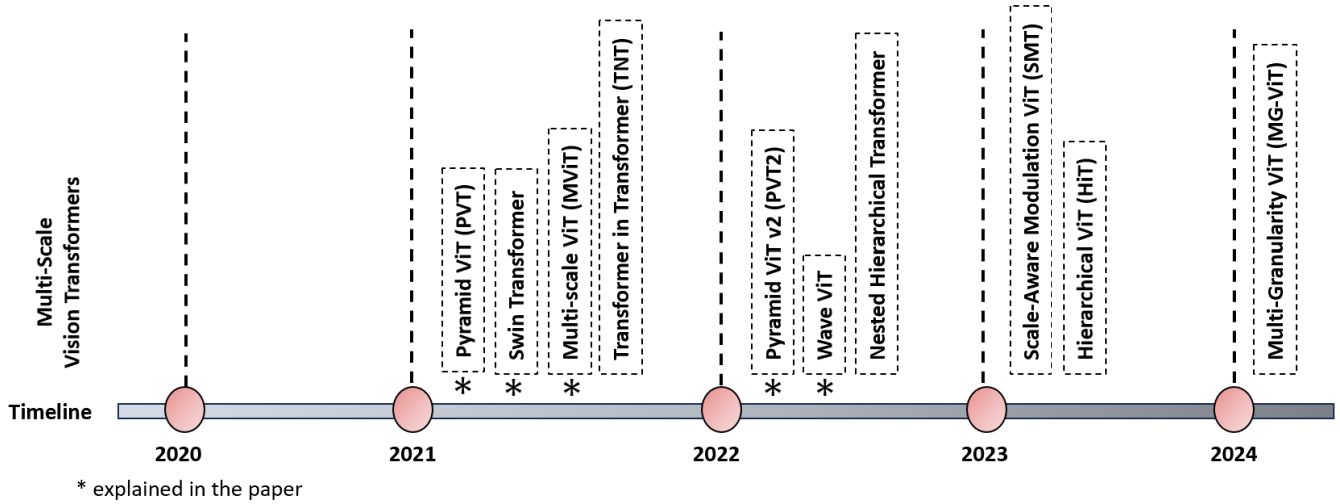


FIGURE 2. Timeline illustrating the proposed Multi-Scale Vision Transformer architectures [13], [17], [23], [34], [40], [60], [61], [75], [81], [82].

III. RELATED WORKS

Several studies have focused on reducing the high computational complexity of vision transformers, resulting in models with similar accuracy but lower complexity. This section offers a brief overview of these approaches.

A. MULTISCALE VIOSN TRANSFORMERS

A widely adopted strategy for addressing the computational complexity of vision transformers is pyramid-style processing. This technique processes input images at multiple scales, effectively capturing both coarse and fine contextual information [13], [42], [44], [68]. Fig. 2. provides a summary of the key approaches within this category. Numerous models have successfully implemented this strategy, including: Pyramid Vision Transformer (PVT) [60], Swin Transformer [40], Multi-scale Vision Transformer (MViT) [13], PVT v2 [61], and Wave-ViT [75].

PVT takes in detailed image patches to effectively capture fine-grained information for high-resolution representation. PVT employs a pyramid structure that gradually reduces in size, helping manage computational complexity in deeper layers. The authors introduce a spatial-reduction attention layer, which plays a role in conserving resources during computation.

The Swin Transformer introduces a hierarchical transformer architecture that utilizes shifted windows for representation computation. This approach ensures linear computational complexity with respect to image size. The model's early stages involve the processing of small patches, with the gradual merging of neighboring patches in deeper layers. The Swin Transformer adopts a shared key set among patches within the same window, effectively mitigating latency concerns associated with earlier sliding window-based self-attention methods.

MViT incorporates several channel-resolution stages, each serving a distinct purpose. In the initial layers, the model operates at an elevated spatial resolution coupled with a constrained channel dimension. As the network delves deeper, spatial resolution diminishes while channel dimensions expand significantly. This ingenious design culminates in the formation of a feature pyramid, effectively encompassing a comprehensive spectrum of features across different scales.

PVT v2 offers reduced computational complexity while preserving local image continuity. Additionally, the model features a flexible position encoding scheme. In the Wave-ViT, they utilize a down-sampling technique based on wavelet transforms and integrate it with self-attention learning. In this architecture, the wavelet block plays a central role. It employs Discrete Wavelet Transform (DWT) to process the key and value inputs separately, dividing them into four distinct subbands. These subbands are then stacked together, followed by a convolutional operation that maintains locality within each subband. The output of this convolutional layer feeds into both the multi-head attention and inverse DWT layers.

While multi-scale designs effectively capture contextual information at various resolutions, they typically rely on fixed embedding dimensions and lack mechanisms for incremental refinement or adaptive token management. Our proposed solution addresses these gaps by employing smaller initial embedding dimensions that incrementally increase based on attention-driven sampling, enabling dynamic refinement of resolution as the model processes the input. Additionally, our approach incorporates token pruning guided by attention scores, ensuring that computational resources are focused on the most relevant features. This integration of context-aware approximation with incremental resolution enhancement complements existing multi-scale methods.

B. PATCH AND TOKEN PRUNING

Numerous studies have highlighted the sparse nature of attention matrices within transformer models and identified instances of token redundancy that don't significantly contribute to final predictions. Building upon these observations,

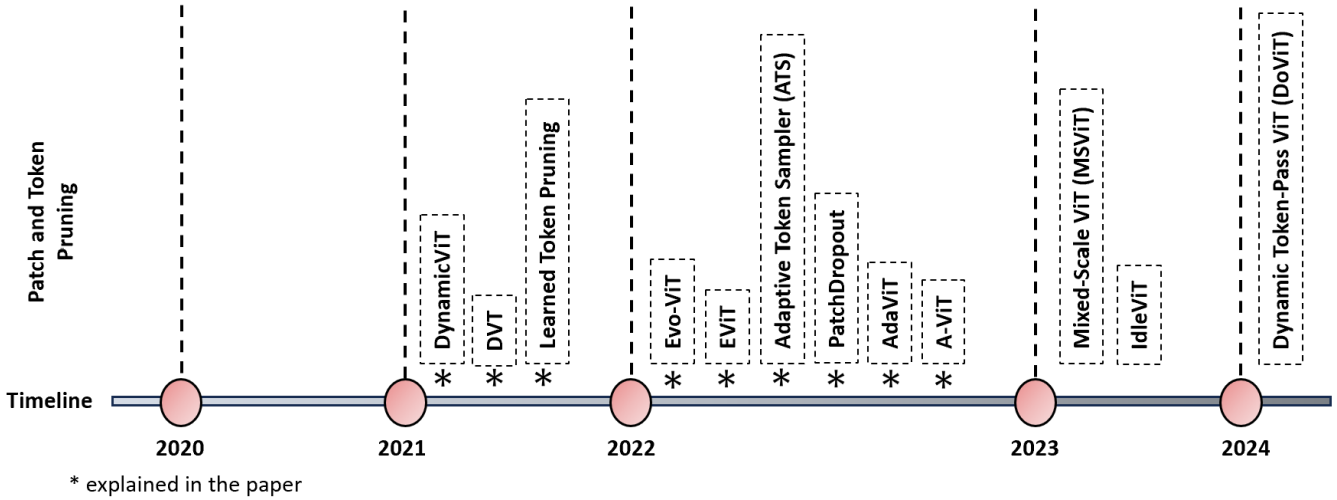


FIGURE 3. Timeline illustrating the proposed token and patch pruning approaches for transformer architectures [14], [19], [26], [31], [37], [38], [41], [48], [63], [73], [74], [76].

various strategies have been introduced to trim these redundant tokens and enhance efficiency [5], [25], [28], [39], [59], [67]. Unlike convolutional models, transformers can leverage unstructured pruned inputs to their advantage. The proposed techniques for token pruning can be classified into two primary categories: static pruning and dynamic pruning techniques. Certain methods apply a consistent approach across different types of inputs. Other techniques adapt their strategies based on the specific characteristics of different inputs. Rao et al. [48], introduced DynamicViT. This approach centers around enhancing the transformer architecture by integrating a predictive module into specific layers. The purpose of this module is to forecast the importance score assigned to each token. Consequently, tokens with lower scores undergo a hierarchical pruning process. This strategy entails disconnecting tokens that have been pruned from the remaining tokens within the attention matrix. This task is achieved using the Gumbel-Softmax masking strategy.

Wang et al. [63] demonstrated a correlation between the complexity of input images and the required number of tokens for accurate predictions. This suggests that simpler images can be accurately predicted using fewer tokens. They introduced DVT, a sequential transformer model designed to process images with varying token counts. In their approach, the concept of early exiting is utilized. This involves halting computation if accurate predictions are achieved, thereby bypassing the use of models with higher token counts. To optimize the utilization of upper-level transformer models and prevent computational inefficiencies, they reused features and relationships.

Kim et al. [26], introduced Learned Token Pruning (LTP). This approach involves dynamically removing tokens that are deemed less significant. This determination is made by utilizing a threshold value, which the model learns independently for each layer. Xu et al. [74] proposed an approach called Evo-ViT in which they introduced the concept of slow-fast updating. This involves employing separate computational pathways to update tokens based on their informational relevance. This way, informative and uninformative tokens are processed differently, helping to maintain the spatial structure of the data during updates. Liang et al. [31] proposed EViT, which utilizes a token reorganization strategy for detecting attentive tokens while merging inattentive ones into a singular token. The unique feature of this model is its independence from the necessity of a fully trained ViT. Nevertheless, adjusting the target ratio would still mandate retraining the model. Fayyaz et al. [14] presented ATS, a distinctive module referred to as a differentiable and parameter-free Adaptive Token Sampler. This module facilitates the dynamic selection of tokens from input images based on attention scores, allowing for variability in the number of chosen tokens for each image. ATS can be seamlessly incorporated into a pre-trained model to enhance its performance. Liu et al. [37] established that employing a technique known as PatchDropout, which involves the random omission of input image patches, enables efficient training of standard ViT models at high resolutions. This method achieves a significant reduction of at least 50% in both FLOPs and memory consumption on typical datasets with natural images.

Meng et al. [41] introduced AdaViT, a framework that autonomously determines the utilization of patches, self-attention heads, and layers within ViT. The core innovation involves integrating a lightweight multi-head subnetwork (referred to as the decision network) into each transformer block of the backbone network. This auxiliary network learns to predict binary choices concerning patch embedding incorporation, self-attention head engagement, and block omission across the network. Yin et al. [76] introduced A-ViT, which

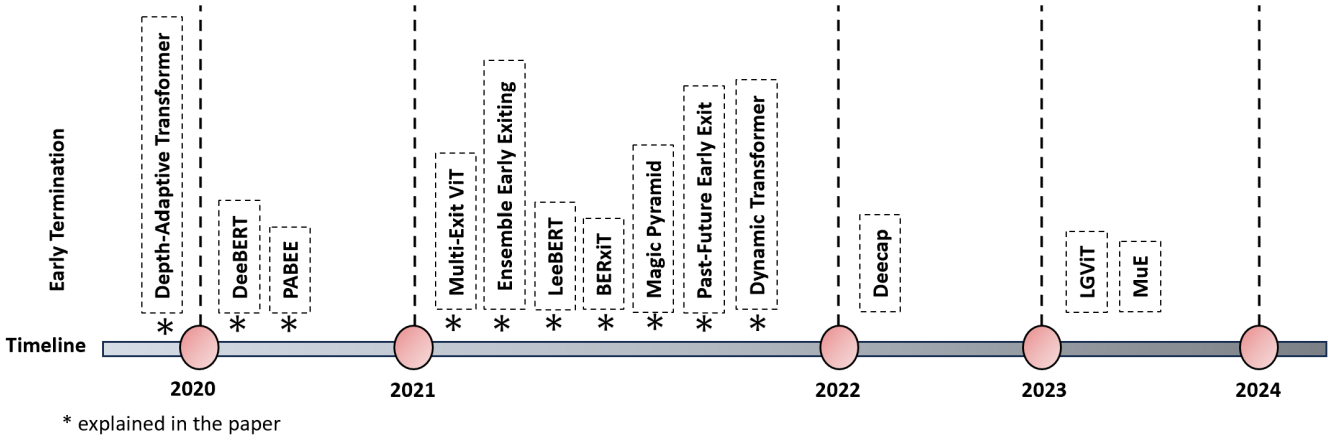


FIGURE 4. Timeline of proposed early exiting approaches for transformers [3], [11], [15], [21], [33], [52], [53], [63], [70], [71], [72], [83], [84].

involves an adaptive inference mechanism. This mechanism intelligently stops the computation process for various tokens at different depths, focusing computational resources only on tokens that contribute significantly to discrimination. This approach dynamically adjusts the allocation of computation resources. A-ViT integrated with high-performance hardware. This is facilitated by the removal of paused tokens from ongoing computations, resulting in enhanced computational efficiency. The entire halting process can be acquired through the model's existing parameters.

These methods effectively reduce computational complexity but are not without challenges. Many approaches rely heavily on retraining or manual parameter adjustments, which limits their scalability and adaptability. Static methods often lack contextual awareness, whereas dynamic methods can introduce significant overhead or fail to maintain spatial and semantic coherence. Our proposed architecture bridges these gaps by incorporating a context-aware approximation mechanism with incremental resolution enhancement. Unlike static approaches, it dynamically adjusts embedding dimensions and utilizes token pruning based on attention scores, ensuring computational resources are dedicated to the most informative tokens. Additionally, the token pruning mechanisms discussed in this subsection seamlessly integrate with our model, boosting its efficiency and adaptability.

C. EARLY TERMINATION

Earlier, scholars introduced the concept of anytime prediction in the realm of computer vision. Multiexit architectures can be created from deep neural networks by introducing branches that exit early after certain intermediate layers. This transformation enables the inference process to adapt dynamically, which proves beneficial for IoT applications with strict latency demands. These applications often face fluctuating communication and computation resources [3]. Building upon this idea, certain studies have extended this approach to transformers, effectively striking a favorable balance between prediction speed and accuracy. Elbayad et al. [11] introduced a depth-adaptive transformer in which predictions occur at different network stages, with network length and computation adjusting according to input sequences. They trained the decoder using aligned and mixed methods, examining sequence-specific versus token-specific depth prediction approaches. Xin et al. [70], revealed that different layers of BERT exhibit varying behaviors, with some layers being redundant. As a solution, they introduced DeeBERT, which enables samples to exit earlier through off-ramps to improve efficiency. Zhou et al. [83], introduced a solution known as Patience-based Early Exit (PABEE), which involves incorporating an internal classifier within each layer of a pretrained language model (PLM). This approach is designed to halt the model's inference process when the internal classifier's accuracy remains consistent for a predetermined period, effectively mitigating unnecessary processing. Sun et al. [52], constructed an ensemble model that utilized internal classifiers. This approach capitalizes on the internal classifiers being trained for predictions on the same task. They introduced a voting strategy that leverages predictions from all preceding internal classifiers. This strategy aids in determining both the optimal timing for exiting the process and the corresponding label assignment. Li et al [30] introduced an approach aimed at expediting the inference process of a pre-trained sequence labeling model. The proposed methodology includes two distinct components: SENTEE, which operates at the sentence level and enables early exiting in sequence labeling, and TOKEE, an early-exit mechanism functioning at the token level. LeeBERT [84] introduces a training approach in which every exit learns not only from the final layer but also from each other. In the BERxiT paper [71], the focus was on rectifying drawbacks in earlier early exit methods for BERT. These methods were restricted to classification tasks and couldn't fully leverage BERT's potential due to limited fine-tuning strategies. The solution introduced was a "learning-to-exit" module, extending early exits to diverse tasks and enhancing BERT's utilization.

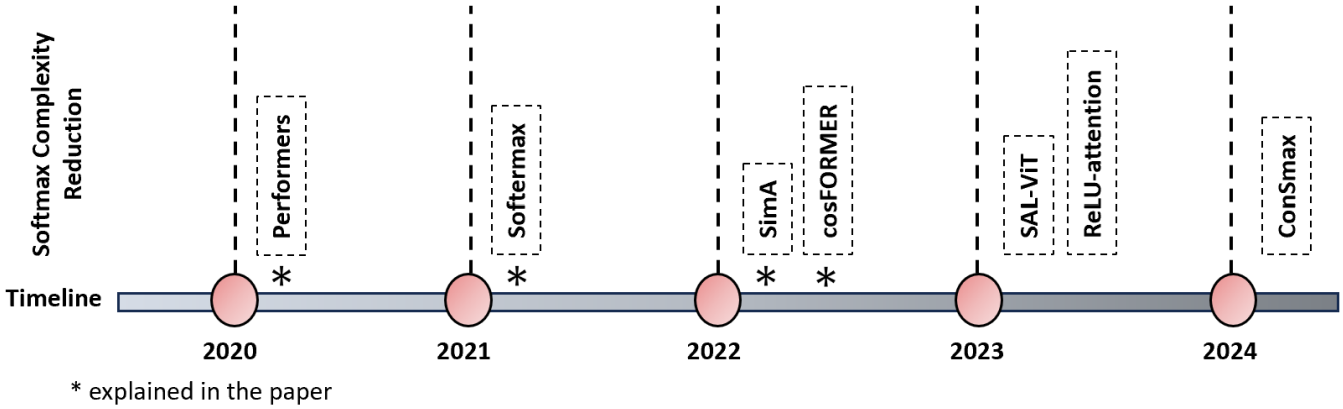


FIGURE 5. Softmax complexity reduction approaches in transformers [6], [27], [35], [46], [51], [65], [80].

He et al. [21], presented Magic Pyramid (MP) which utilizes token pruning for width-wise computation reduction and incorporates early exit strategies to address depth-wise computation reduction, ultimately leading to improved efficiency in model inference. Liao et al. [33] presents a solution for global early exits, utilizing information from both preceding and subsequent layers to facilitate the exit process. The Dynamic transformer [63] was introduced as an approach aimed at automatically determining the optimal number of tokens necessary for processing each input image. This was accomplished by utilizing a series of interconnected transformers, with each one accommodating an increasing number of tokens. Throughout the testing phase, these transformers were sequentially activated in an adaptable fashion. In essence, the inference process would conclude once a prediction of sufficient confidence was generated. In this study [3], seven distinct designs for early exit branches are introduced, which can be integrated into ViT backbones. By conducting comprehensive experiments involving tasks such as image classification and crowd counting - the latter involving regression, it is demonstrated that these architectures offer valuable options for striking a balance between classification accuracy and inference speed, depending on the specific task.

To conclude this subsection, we emphasize that the early termination mechanisms discussed here offer a variety of strategies for balancing computational efficiency and predictive accuracy. In our proposed architecture, classification heads have been integrated into the IRET model to facilitate real-time predictions, providing an inherent mechanism for early exits. Furthermore, the diverse early prediction techniques reviewed can be seamlessly incorporated into our framework.

D. SOFTMAX COMPLEXITY REDUCTION

The softmax operation in transformers is a major computational bottleneck, especially with longer sequences. It relies on costly exponential functions, and achieving numerical stability often involves extra steps. Efforts to speed up, approximate, or eliminate softmax have been made [6], [27], [51], [57]. Choromanski et al. [6] proposed Performers, for estimating regular full-rank-attention transformers with linear space and time complexity. Unlike traditional methods that rely on priors like sparsity or low-rankness, performers employ fast attention to approximate softmax attention kernels. This enables scalable and efficient modeling of attention mechanisms beyond softmax, addressing the quadratic complexity challenge of conventional transformers.

Koohpayegani et al. [27] presentd SimA, an attention block that replaces the softmax layer with L1-norm normalization, simplifying attention to matrix multiplications. SimA maintains comparable accuracy to state-of-the-art transformer variants like DeiT, XCiT, and CvT while removing the computational overhead of Softmax.

Stevens et al. [51] introduced Softermax, an optimized softmax algorithm designed for hardware efficiency. Softermax integrates base replacement, low-precision softmax computations, and an online normalization calculation. By leveraging the fine-tuning principles of transformer-based networks, they apply Softermax-aware fine-tuning to minimize accuracy loss without imposing additional training burdens. Furthermore, they provided insights into the microarchitecture required for implementing Softermax in an inference accelerator. Qin et al. proposed cosFORMER, which leverages non-negativity and a non-linear re-weighting scheme in the softmax attention matrix to create a linear transformer [46].

The softmax overhead reduction methods discussed provide efficient alternatives to traditional softmax, improving scalability and accuracy. By integrating these techniques into our proposed model, we can further decrease computational overhead and enhance performance.

E. NOVEL ATTENTIONS

The computational challenge of quadratic complexity in self-attention has long been a prominent obstacle when applying the models to tasks in computer vision. A primary research focus in enhancing the efficiency of vision

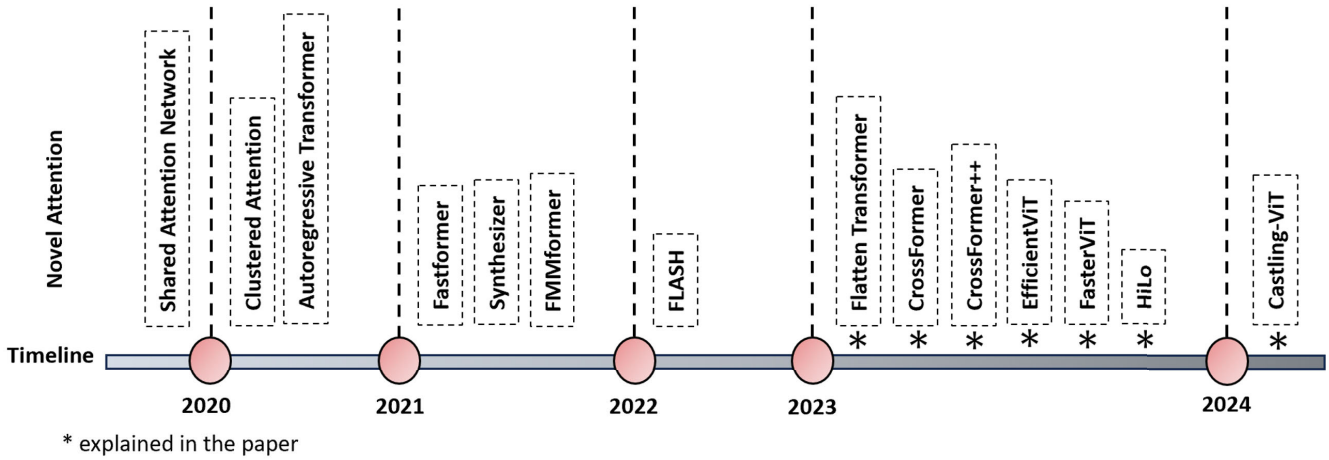


FIGURE 6. Attention optimization approaches for transformer models [16], [18], [22], [24], [36], [43], [45], [54], [58], [62], [66], [69], [77].

transformers involves the reduction of computational costs associated with self-attention modules [2], [29], [47], [50], [79].

In Flatten Transformer, the focused linear attention module addresses the computational complexity issue of self-attention by introducing a module that combines high efficiency with expressiveness [16]. CrossFormer and its enhanced version, CrossFormer++, explicitly leverage features of different scales, while addressing issues such as self-attention map enlargement and amplitude explosion [62]. EfficientViT introduces a new building block and cascaded group attention module to improve memory efficiency and computational redundancy in transformer models, achieving a commendable trade-off between speed and accuracy [36]. FasterViT [18], a hybrid CNN-ViT neural network amalgamates the swift local representation learning of CNNs with ViT's global modeling capabilities. Within FasterViT, the Hierarchical Attention (HAT) technique efficiently breaks down the quadratic complexity of global self-attention into multi-level attention mechanisms, effectively curtailing computational costs. By harnessing efficient window-based self-attention, individual windows are endowed with dedicated carrier tokens, fostering both local and global representation learning. HiLo [45] encodes high frequencies using local window self-attention and captures low frequencies through global attention within the input feature map.

Castling-ViT [77] tackles the challenge of enabling ViTs to efficiently learn both global and local context during inference. It uses linear-angular attention and masked softmax-based quadratic attention during training and switches to linear-angular attention alone during inference. The framework employs angular kernels for query-key similarity, simplified by decomposing them into linear terms and high-order residuals, and incorporates modules like depthwise convolution and masked softmax attention to efficiently learn global and local information.

The attention computation reduction methods discussed here provide innovative solutions to address the quadratic complexity of self-attention while preserving accuracy. Integrating these techniques into our proposed model can further enhance its efficiency, enabling it to handle complex tasks with reduced computational overhead and improved scalability.

IV. IRET: PROPOSED METHOD

To adapt vision transformers for resource-constrained devices, reduce their computational and memory requirements, and address the shortcomings of previous solutions discussed in the previous section, we introduce IRET.

A. ARCHITECTURE OF IRET

The high-level architecture of IRET is shown in Fig. 7. The innovation in IRET is the ability to focus on attended tokens in addition to forgetting unattended tokens.

As illustrated in Fig. 7, IRET replaces several transformer encoder layers with IRET encoders. The architecture of an IRET encoder is shown in Fig. 8. IRET encoder pre-processes the tokens for token dropping and token focusing before performing the encoding. More specifically, similar to prior work in [14] and [48], IRET performs the token dropping based on CLS token attention scores, dropping tokens with low attention scores to prune the computational tree.

However, as illustrated in Fig. 8 IRET also has an attention-based mechanism for an incremental sampling of the input image using an "attention-based focusing" module. The focusing module received a new sample of the input image using a learnable 2D-lifting scheme in [49] that is shared across IRET layers. Details of the 2D-lifting scheme will be explained later. We refer to this input image sample as a sub-band sample. Each generated sub-band is then divided into patches with a 1-to-1 mapping relationship to input image patches. Based on the attention-score of input (existing) tokens, the token focusing module then decides for

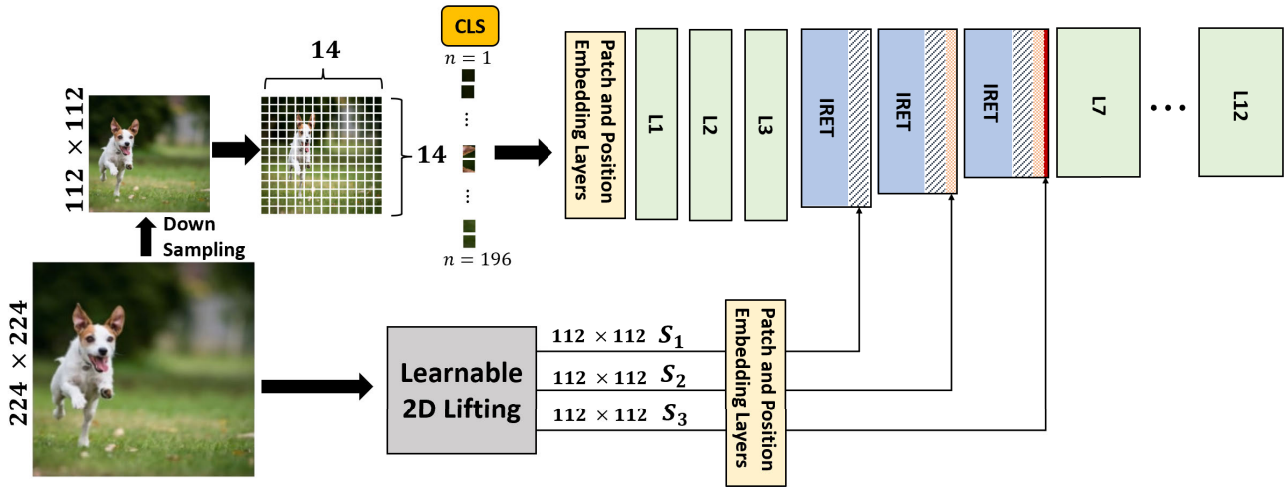


FIGURE 7. The IRET architecture processes input through four sampling steps: initially with a scaled low-pass filter and then three times using learnable 2-D lifting schemes. With each IRET layer, the embedding size of each token increases as it assimilates additional information. Concurrently, before each IRET layer, less-attended tokens are dropped. Therefore, each IRET layer has dual roles: discarding unattended tokens and focusing on attended ones through extra sampling. The transformer encoder's increasing size visualizes the growth in embedding size at each IRET encoder.

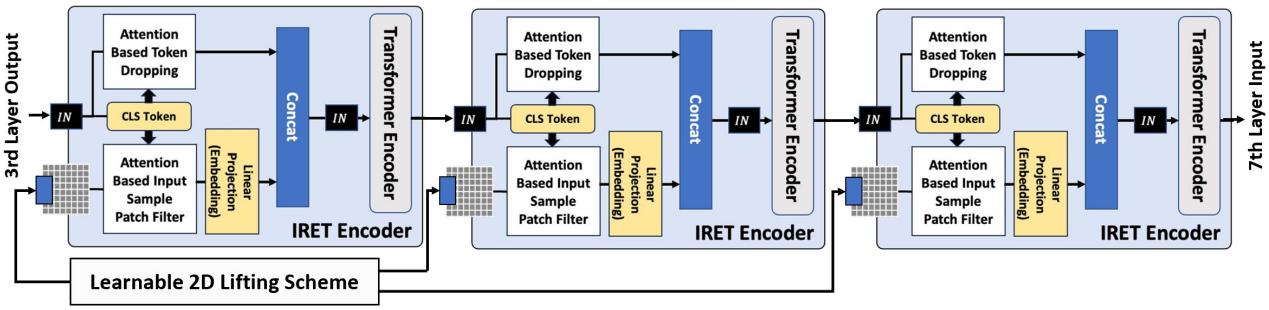


FIGURE 8. The IRET layer architecture utilizes the CLS token to identify unattended tokens, employing a token dropping method to remove them. Additionally, it determines which tokens require more focus based on the CLS token. This process involves filtering patches from the input sample created by the 2D lifting scheme, projecting these patches into new embeddings, and then concatenating new information to enhance the existing token embeddings. By enlarging the embedding size, IRET increases focus on attended tokens.

each patch in the newly sampled sub-band to be ignored or forwarded to the layer. If the corresponding token coming from the previous encoder has an attention score above the desired threshold, the token is deemed useful and is subjected to embedding. The embedded information for each sub-band that corresponds to an attended token is concatenated to the embedding of that token, increasing the embedding size, which is analogous to improving focus on that part. The size of each encoder layer in Fig. 7 corresponds to the embedding size of its token. Using this illustration, as shown in Fig. 7, each IRET encoder layer (shown in blue) increases the embedding size (shown in dark blue), while each regular transformer encoder layer maintains the embedding size.

B. INPUT SAMPLING PROCEDURE

To obtain input image samples for incorporation into IRET layers, we investigated three methodologies: 1. Utilizing the original input, 2. Employing DWT subbands, and 3. Adopting a learnable sampling approach known as the 2D-lifting Scheme. The outcomes obtained through these various sampling approaches are thoroughly examined and detailed in Section V, shedding light on the efficacy and impact of the chosen sampling methods on the overall performance of the model.

Original Input- In this experiment which is the baseline, we feed the downsampled original input three more times to the model using the IRET layers. The goal of this experiment is to check the ability of the model to learn new features from the new embedded samples of the original input.

DWT Subbands- An alternative sampling approach involves the utilization of Discrete Wavelet Transform (DWT). Numerous investigations have harnessed the capabilities of DWT within the realm of computer vision to augment diverse facets of image analysis and processing. DWT, a mathematical technique adept at decomposing signals or images into their fundamental frequency components, provides a unique pathway for feature extraction, representation, and manipulation. We employ DWT to produce four

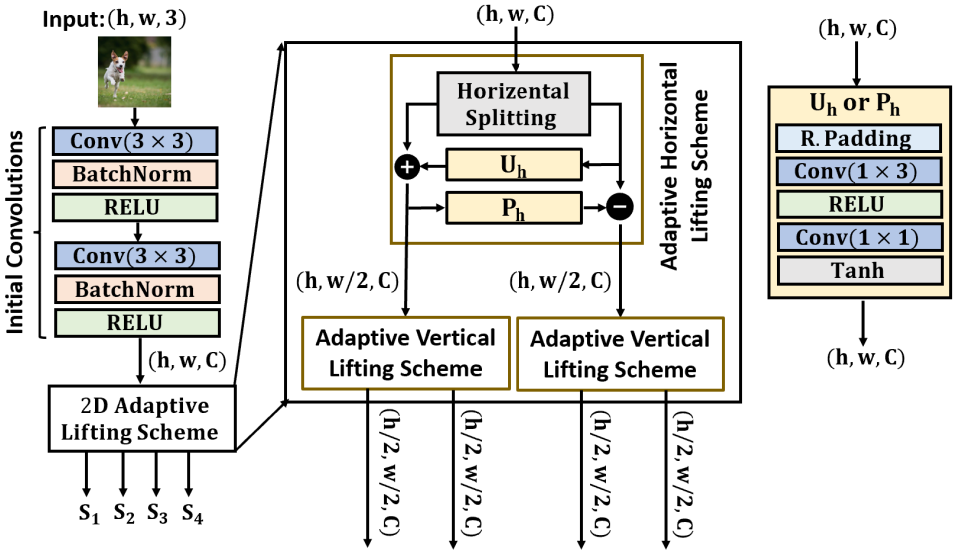


FIGURE 9. The Architecture of Learnable 2D Lifting Scheme. It receives the original image and learns four output samples. S_1, S_2 and, S_3 are used in the architecture of IRET.

subbands of images, each sized 112*112, denoted as LL, LH, HL, and HH. Due to the LL subband containing a greater entropy of information, it is assigned to the initial layer. Simultaneously, the LH, HL, and HH subbands are sequentially inserted into the subsequent IRET layers for further processing. This distribution ensures an effective utilization of the information content across the layers of the model.

2D-lifting Scheme- The architecture of the 2D-lifting scheme [49] used in the IRET layer is shown in Fig. 9. The lifting scheme is designed to take a signal, denoted as x, as its input and produce two key outputs: the approximation sub-band (c) and the details sub-band (d) of the wavelet transform. The process of designing this lifting scheme involves three distinct stages: Splitting the signal, Updater, and Predictor. Eq. 8 through 10 describes the functionality of these stages. The predictor and updater components are implemented using CNN layers. Each CNN consists of two layers: the first layer uses a kernel size of 1×3 (for horizontal processing) or 3×1 (for vertical processing) and employs a ReLU activation to enhance non-linear representations. The second layer uses a 1×1 convolution with a tanh activation to stabilize the range of outputs. The signal x is partitioned into two components in splitting stage: an even component and an odd component. The even component consists of all the values located at even positions in the sequence and in the update operation is often used as the basis for creating the approximation sub-band c, which captures the low-frequency details of the signal. The odd component consists of all the values located at odd positions in the sequence and in the prediction operation is used to derive the detail sub-band d, which captures the high-frequency details of the signal.

$$x_e[n] = x[2n], x_o[n] = x[2n+1], x : input\ signal \quad (8)$$

$$c[n] = x_e[n] + U(x_o{}^{L_U}[n]),\ U(.) = update\ operator \quad (9)$$

$$d[n] = x_o[n] - P(c^{L_P}[n]),\ P(.) = prediction\ operator \quad (10)$$

The loss function of learnable updater and predictor is defined as.

$$Loss(P) = \Sigma_n(P(c^{L_P}[n]) - x_o[n])^2 \quad (11)$$

$$Loss(U) = \Sigma_n(U(x_o^{L_U}[n]) - (x_o[n] - x_e[n]))^2 \quad (12)$$

For the predictor, the loss function minimizes the mean squared error between the predicted and actual odd samples and the updater's loss function minimizes the error in approximating the difference between the odd and even components.

The initial convolutional layers extract discriminative features from the data before downsampling. This is done using two sequences of convolution, batch normalization, and ReLU with a kernel size of 3×3.

It's important to note that to minimize overhead, a portion of the 2D-lifting scheme is shared across IRET encoder layers. Nonetheless, each IRET encoder layer is fed by a unique segment of the 2D-lifting scheme, ensuring it receives a distinct sample. Additionally, this 2D-lifting scheme is designed to be learnable, enabling its integration and training alongside the rest of the model in an end-to-end manner. During the training phase, the lifting scheme is trained as an integral component of the IRET architecture. This approach allows each IRET layer to adaptively incorporate new and unique features, differentiating them from previous sampled information for each token.

To maintain the positional information of patches in newly sampled images we employ a position embedding layer to add this data to their embedding. Prior to adopting learnable layers, we explored different sampling techniques for the input image, like DWT, using each sub-band as a separate input to the feature encoding layer. However, our

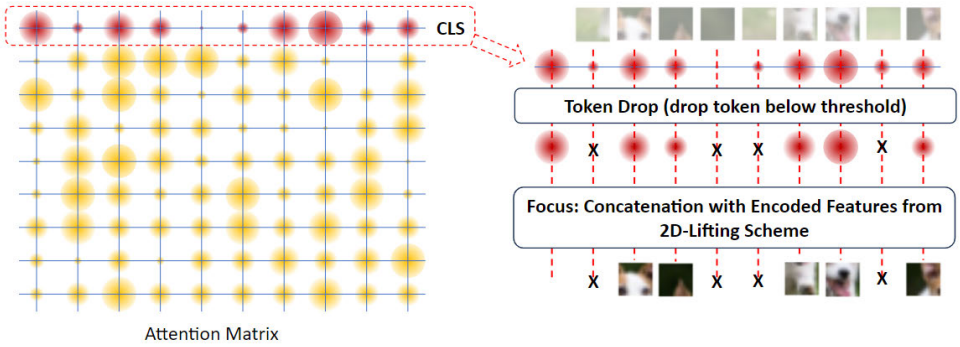


FIGURE 10. The IRET layer features two main functions: attention-based token dropping and focusing. It eliminates unattended tokens using attention scores to simplify computation and enlarges the embedding size for attended tokens with extra features from a 2-D lifting scheme. This process, akin to human brain focusing, allows IRET to selectively prioritize certain tokens, thereby boosting accuracy and lowering computational complexity.

findings indicated that a learnable lifting scheme, which learns features based on model loss and trained alongside the main model, yields the highest accuracy.

Also, note that the input to IRET is a scaled version of the input image. For example, input to the embedding layer of DeiT is a 224 × 224 pixel image. For IRET, we take a scaled 112 × 112 pixel image as input and also reduce the embedding size of the first layer from 384 to 192. subsequently in each IRET layer (that in the variant shown in this proposal is positioned in layers 4, 5, and 6, the embedding size of features is increased from 192 to 294, 348, and 384 respectively bringing in additional 102, 54, and 36 embedding dimensions with each added IRET layer. Starting with a smaller embedding size and working with a smaller embedding size in the first 6 layers of the IRET layers allows a significant reduction of the computation. By working with a smaller embedding size, IRET first decides where to look for information in the input image. As the attention scores highlight the importance of various input tokens, then IRET layers stop processing unattended tokens, and more importantly, bring in additional details for the features in attended tokens.

Fig. 10 visualizes the pre-processing function for token dropping and token focusing in an IRET encoder layer. The attention threshold, which is predefined, plays a crucial role in determining the tokens to be dropped or focused. In the left part of the figure, the attention matrix is depicted, with the first row highlighted in red corresponding to the CLS token. This token is essential for classification in the last layer of transformers, as its attention to other tokens reflects their importance. The attention scores in the CLS row are what we use in IRET to decide if a token is to be forgotten (drop) or focused by bringing additional information through the use of a 2D-lifting scheme. In the right part of the figure, the token dropping process is illustrated. At the first layer, tokens with attention scores below the predefined threshold are dropped. In the subsequent layer, the remaining tokens are concatenated with their corresponding tokens from the 2D-lifting scheme. This process increases the embedding dimension, enabling the model to focus more on these tokens, which is visualized by a clearer token at the end.

Fig 11 is another visualization of the token dropping and focusing concept in an IRET encoder. As illustrated, each IRET layer increases the details of each token with a high attention score (this is visualized by increasing image resolution, but in reality, this is achieved by increasing embedding size), while dropping the unattended tokens.

C. IMPROVED EARLY PREDICTION THROUGH MULTI-EXIT ARCHITECTURE IN IRET

As explained in section III, early termination is one of the proposed methods for expediting model prediction. We have further investigated and incorporated this feature to enhance the speed of IRET. As depicted in Fig. 12, various input images present distinct content and pose challenges, showcasing classification tasks with varying levels of complexity. Therefore, our goal is to formulate the IRET with multiple exit options, enabling it to perform early classification after each encoder layer. The decision to advance to the next encoder level is based on how sure the model is about its prediction and our predetermined confidence threshold. This formulation additionally enables the application of IRET in low-power and real-time systems.

Fig. 13 depicts the real-time behavior of the IRET system, wherein the model's execution is halted based on a predefined deadline. Consequently, the model adjusts its complexity by reducing the number of layers when faced with tighter deadlines, and conversely, it utilizes more layers when more time is available for computation. In scenarios with low-power models, computations can be halted to adhere to energy constraints.

It's crucial to consider that the energy constraints play a pivotal role in determining at which layers IRET concludes its operations. This variability stems from the diverse number of skip computations influenced by context-aware computation

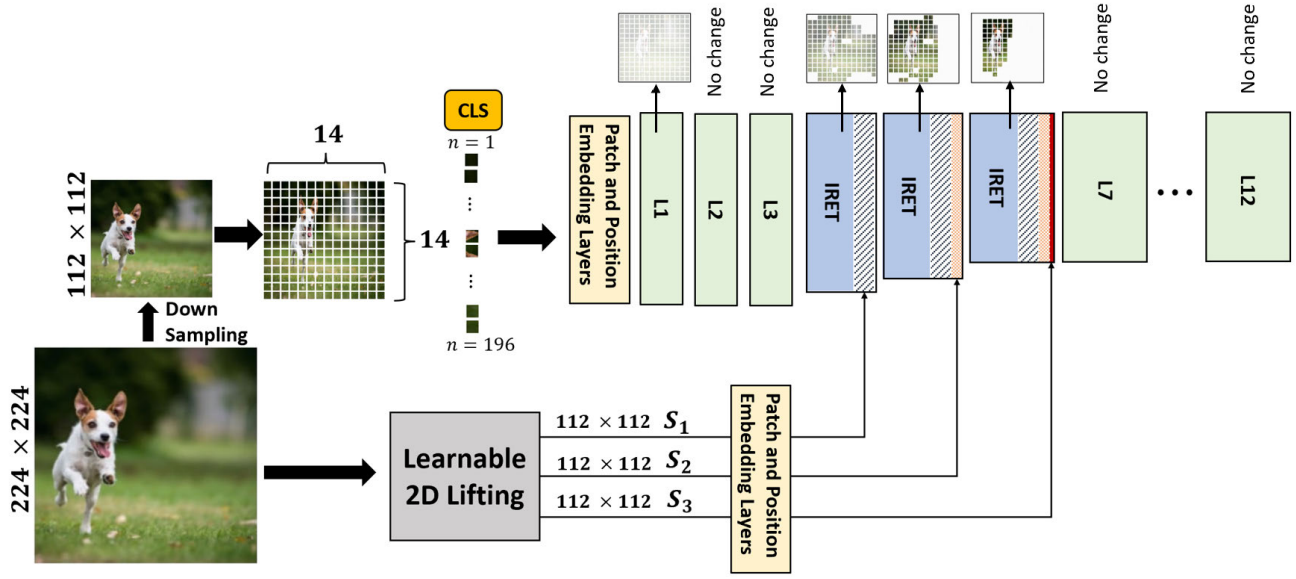


FIGURE 11. In IRET, the 'forget and focus' concept hinges on CLS token attention values. Tokens with attention below a threshold are dropped ('forget'), while those above the threshold see increased embedding size ('focus') via a 2D-lifting scheme. The concept of focus is shown by increased resolution.

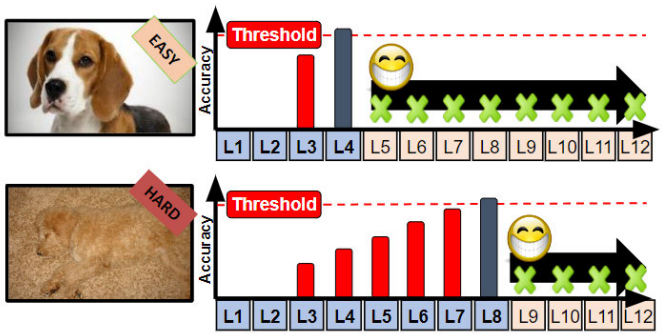


FIGURE 12. IRET exit behavior in low-power and energy aware system.

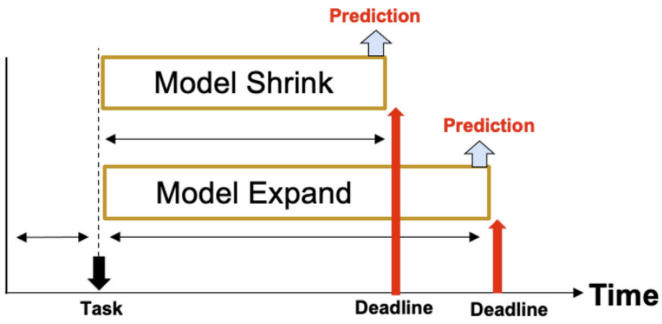


FIGURE 13. IRET exit behavior in real-time system.

and patch elimination, especially when dealing with different images. Figure 14 shows the overview of muti-exit IRET. Extra heads are added to the model after each encoder layer from layers 4 to 12. Classification before the first 3 layers is not useful because the model has more understanding after the 3 first layers.

V. EXPERIMENTS

Our model was developed based on the Facebook DeiT [55] small model with hard distillation, utilizing the Timm library [64]. We conducted our experiments on the ImageNet dataset [8] using Nvidia A100 as the training platform.

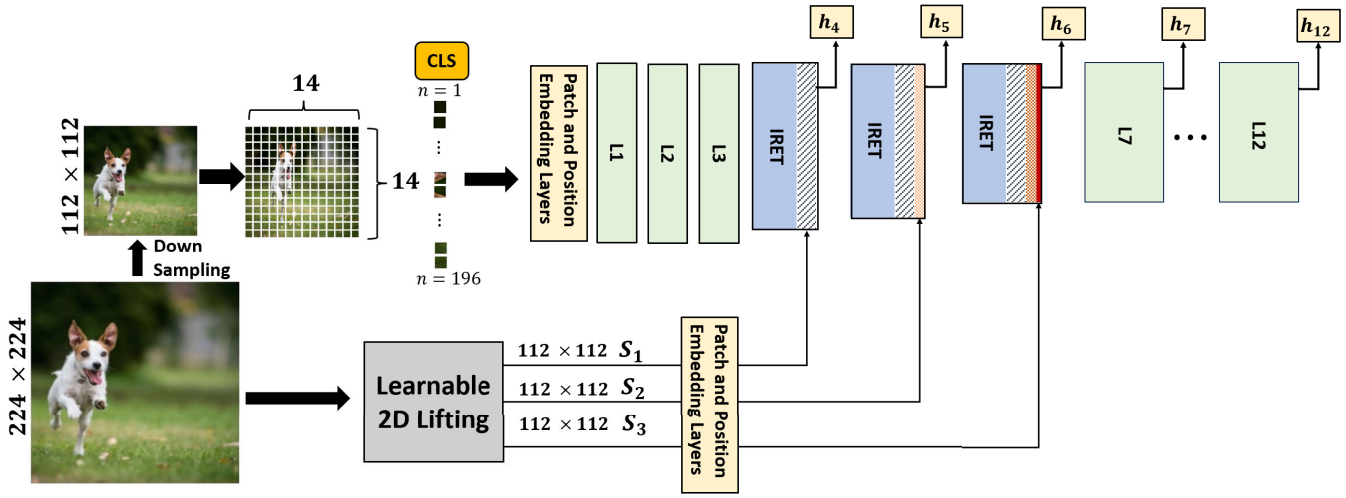


FIGURE 14. The IRET multi-exit architecture. Employing MLP heads at layers 4 to 12 enables the model to make early predictions, effectively bypassing the need for computations in subsequent layers. This strategic approach enhances efficiency and accelerates decision-making in IRET.

To explore the most effective input sampling strategies, three distinct approaches were evaluated. These experiments aimed to address the challenge of balancing computational efficiency with model performance by identifying a subsampling mechanism that retains essential information while reducing redundancy. The performance of each sampling method was assessed using top-1 accuracy metric. Accuracy was defined as the proportion of correctly classified images over the total test set.

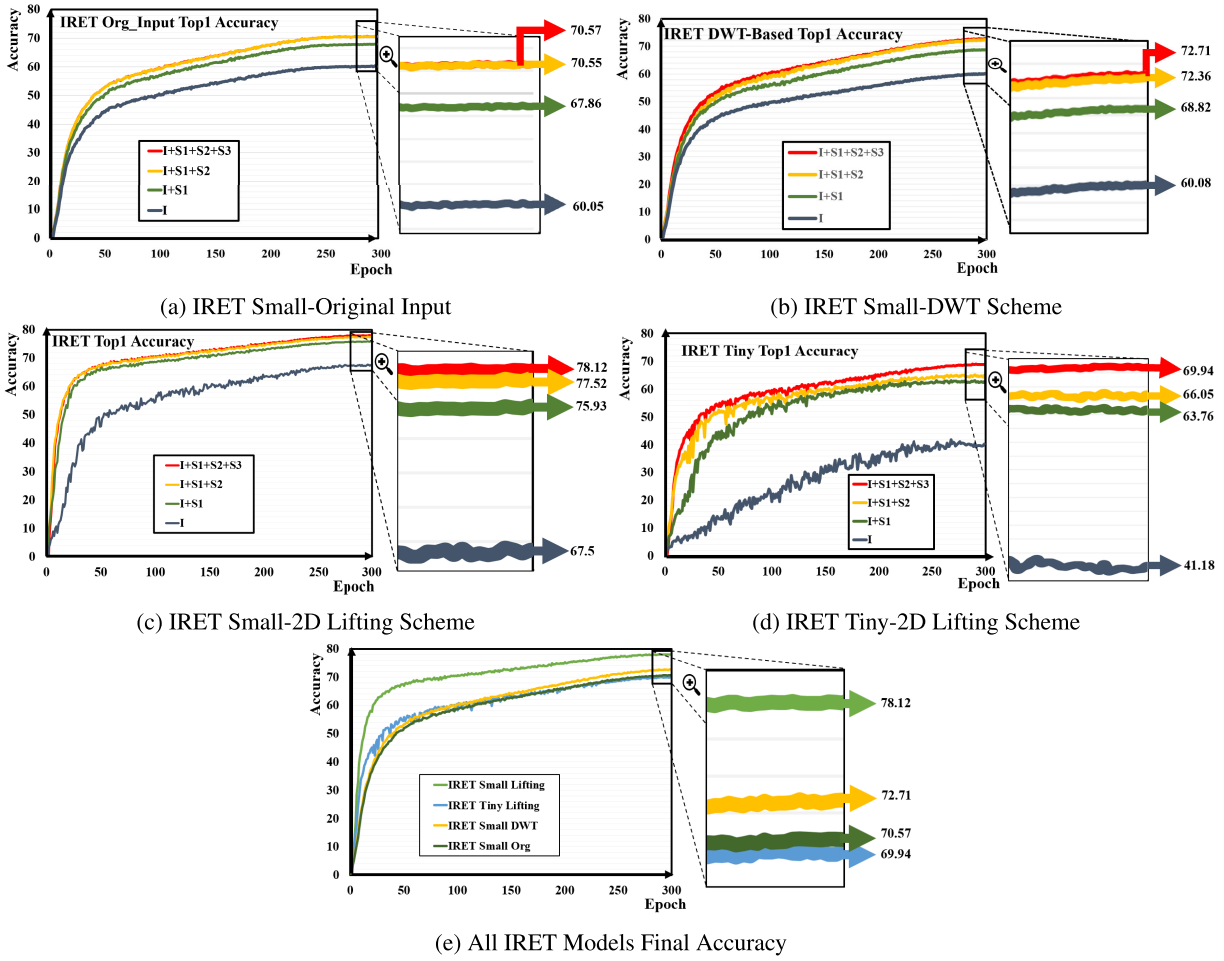
The experimental results demonstrated that the 2D-lifting scheme significantly outperformed the other methods, emerging as the most effective and impactful sampling approach. This scheme was designed to dynamically adapt the subsampling process to capture and preserve essential image features, enabling improved learning by the IRET layers.

For the first sampling mechanism, we simply fed the downsampled image to IRET layers as well Fig. 15a shows the results. In implementing the second sampling approach, the Pytorch Wavelet library was utilized [1]. However, this method proved to be ineffective as the model struggled to learn from subsequent samples introduced to the IRET layers. Consequently, no notable improvement in accuracy was observed following the insertion of LH, HL, and HH subbands. The result of this method is illustrated in Fig. 15b.

These two sampling experiments served as motivation to adopt a trainable subsampling approach. By doing so, the model can gain valuable information from the inputs to the IRET layers, potentially enhancing its learning capabilities.

A learnable approach for subsampling images, such as the 2D-lifting scheme discussed, introduces adaptability and flexibility into the subsampling process. Unlike fixed or predetermined subsampling methods, a learnable approach allows the model to dynamically adjust and optimize the subsampling strategy during training. Learnable subsampling enables the model to adapt its sampling strategy based on the specific characteristics and requirements of the given task. Different tasks or datasets may benefit from different subsampling patterns, and a learnable approach allows the model to learn the most effective strategy for the task at hand. Moreover, Images often contain complex relationships and structures that may vary across different regions. A learnable approach allows the model to capture and exploit these intricate relationships during subsampling, potentially leading to the extraction of more relevant and discriminative features. Also, traditional subsampling methods may discard certain information during the downsampling process. A learnable approach has the potential to minimize information loss by intelligently selecting which details to retain or discard based on the model's learning experience. In essence, a learnable subsampling approach empowers the model to actively participate in the decision-making process, learning and refining its subsampling strategy during training. This adaptability can lead to more effective feature extraction, better task performance, and improved generalization capabilities. Each sample focuses on acquiring insights into a distinct aspect of the image, thereby introducing new features to enhance the model's understanding.

The final model inputs are 112 × 112 pixels, with IRET layers receiving 112 × 112 sub-bands generated by the 2D-lifting scheme [49]. To enhance trainability, we integrated three additional classification heads, each corresponding to a CLS token of an IRET layer. These heads contribute to the total classification error during backpropagation, accelerating the training of the IRET layer and 2D-lifting scheme. These heads are removed post-training for inference. Training lasts for 300 epochs or until accuracy plateaus. Data augmentation included randomly omitting information from the 2D-lifting scheme to assess IRET's incremental learning capability. We evaluated IRET's performance in four scenarios: 1) Using only the input image, 2) Adding the first sub-band sample to the first IRET layer, 3) Incorporating two sub-band samples



(a) IRET Small-Original Input

(b) IRET Small-DWT Scheme

(c) IRET Small-2D Lifting Scheme

(d) IRET Tiny-2D Lifting Scheme

(e) All IRET Models Final Accuracy

FIGURE 15. Top-1 accuracy of the IRET for different input sampling mechanisms and different combinations of inputs. In each experiment, we have tested four combinations: Only the downsampled input I, downsampled input plus one sample $I + S1$, downsampled input plus two samples $I + S1 + S2$, and downsampled input plus three samples $I + S1 + S2 + S3$. (a) Original downsampled input is given to the IRET layers. (b) DWT is used to generate the samples. (c,d) 2D Lifting mechanism is used for sample generation for DeiT small and tiny respectively. (e) Comparing the top1 accuracy of all the experiments using all samples as input.

TABLE 1. IRET's accuracy, FLOP count, and parameter count based on various attention thresholds in the IRET layer, which affect token dropping and focusing.

Attention Threshold	Top-1	Top-5	GFLOPs	Params(M) Used
IRET (Base)	78.12	93.28	3.51	17.24
0.0004	77.86	93.05	2.82	13.75
0.0005	77.68	92.92	2.51	12.24
0.0008	76.98	92.61	1.86	9.07
0.0012	75.98	92.01	1.42	6.93
0.0015	75.3	91.56	1.28	6.24
0.0018	74.36	91.01	1.08	5.27
0.002	73.76	90.64	1.02	5.17
0.003	71.11	88.97	0.65	3.17

in the first and second IRET layers, and 4) Including all three sub-band samples.

Fig. 15c presents the top-1 training accuracy of IRET across these scenarios. The figure shows IRET's proficiency in incremental learning, with diminishing accuracy gains upon adding more sub-band samples. The first sub-band's addition notably boosts accuracy, but subsequent samples yield lesser improvements. This observation made us limit the number of IRET layers. The embedding size distribution across sub-bands also affects incremental learning rate

IRET Symbol	I9	I8	I7	I6	I5	I4	I3	I2	I1
Attention Threshold	0.003	0.002	0.0018	0.0015	0.0012	0.0008	0.0005	0.0004	N/A

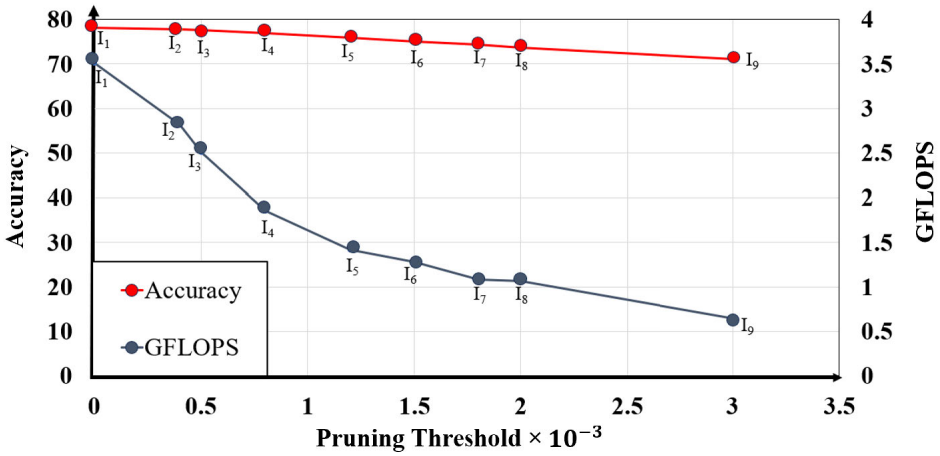


FIGURE 16. Change in accuracy and flop count as a function of attention threshold for token dropping and token focusing.

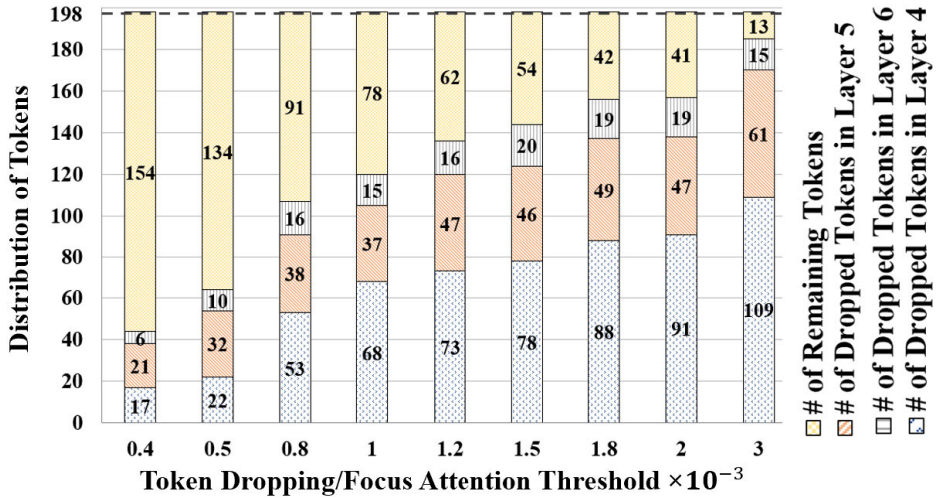


FIGURE 17. Token dropping of layers with pruning policy based on different threshold values. For smaller threshold values, the model drops fewer tokens.

and final accuracy. As shown, the model's top-1 accuracy improves from 67.5% to 75.93%, 77.52%, and 78.12% with the addition of new information extracted from sampled sub-bands. We have also developed the IRET-tiny using DeiT tiny model. Fig. 15d shows the IRET-tiny accuracy using lifting-scheme with the top-1 accuracy close to 70%. Finally, in Fig. 15e, the top-1 accuracy of the experiments is compared when they receive all samples. This serves as evidence of the superiority of the DeiT small model and the utilization of the 2D lifting scheme.

As mentioned above, the IRET layer facilitates a dynamic balance between computational complexity and model accuracy. In the realm of approximate computing, the ideal scenario is achieving a substantial reduction in computational complexity with only a minor impact on performance. IRET exemplifies this by enabling dynamic observation of such trade-offs. The token dropping and focusing attention threshold in each IRET layer is the control knob for this trade-off. The threshold could be different for each IRET encoder. However, for simplicity in this study, we apply a uniform attention threshold across all IRET layers, leaving detailed exploration of threshold variations for future research.

Table 1 presents the top-1 and top-5 accuracy, FLOP count, and parameters of IRET under various attention thresholds for token dropping and focusing. The IRET's parameter count remains constant at 17.24M, but attention thresholding reduces the number of parameters actively used by discarding those related to dropped tokens. It's important to differentiate between used parameters and those loaded from memory, as data movement depends on the hardware

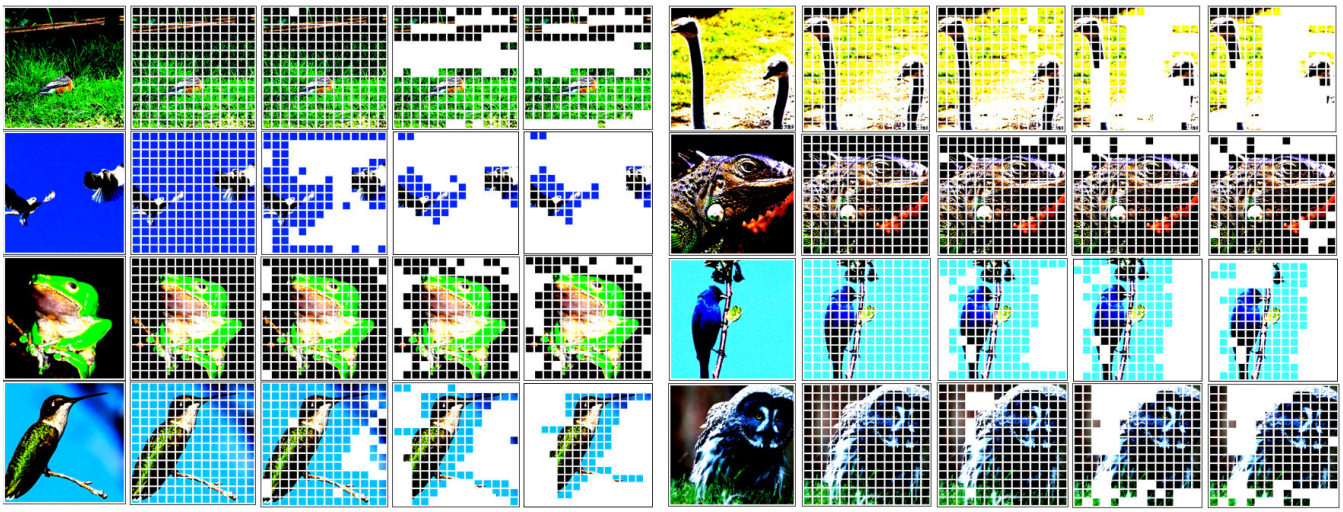


FIGURE 18. Visualizing forget and focus with attention threshold of 0.0004 for samples from the ImageNet dataset.

accelerator's architecture, including buffer sizes and mapping solutions. Reduction in used parameters leads to decreased data movement in the hardware accelerator, which we plan to explore further in future work. Fig. 16 visualizes how increasing the threshold size effectively prunes the model with minimal impact on accuracy. This balance is achieved by the token-dropping module reducing complexity and the focusing module maintaining accuracy.

Fig. 17 presents the average pruning results across different threshold values, illustrating how the number of dropped tokens varies within each layer, as averaged over the ImageNet test set. This figure demonstrates the sensitivity of our pruning strategy to the attention threshold. In the IRET model presented in this paper, there are 3 IRET encoder layers. As illustrated, by increasing the pruning threshold, the number of dropped tokens in each layer and total number of dropped tokens increases. In the extreme case, with the attention pruning threshold of 3E-3, as illustrated in this figure, 109 tokens are dropped in layer 4 (IRET layer 1), 61 in layer 5, and 15 in layer 6. In this case, from table 1, the top-1 accuracy of 71.11 and top-5 accuracy of 88.97 is achieved by focusing on only 13 tokens.

Fig. 18 showcases the practical impact of our pruning strategy on individual samples. This figure visually compares the original token distribution with the pruned results at different layers, emphasizing how the model selectively focuses on the most critical tokens. The visualizations provide a clear depiction of how irrelevant or less important tokens are gradually removed as the layers progress, allowing the model to allocate more resources to the most informative regions.

Fig. 19 illustrates the trade-off between computational complexity and accuracy for IRET, comparing it to prior art solutions. Increasing the attention threshold in IRET leads to a gradual decline in accuracy but with a significant reduction in computational complexity. It's crucial to note that the data points for ATS [14], DeiT [55], ResNet [20], and AdaVIT [41] represent different models. For ResNet, the accuracies correspond to models with varying depths from 18 to 152 layers. DeiT and ATS models differ in embedding sizes (384, 318, 258, 192), meaning each point reflects a distinct model architecture optimized for specific accuracy. In contrast, all IRET data points are derived from the same architecture, starting with an embedding size of 192 and incrementally increasing it through the IRET encoder layers to 294, 348 and 384 respectively. The variations in IRET's FLOP count and accuracy are due to different attention thresholds for token dropping, assumed uniform across all layers in this study. Adjusting these thresholds layer-wise in IRET, with incremental increases, could further enhance accuracy.

It is also worth noting that in IRET, token focusing and dropping occur in layers 4, 5, and 6 (IRET layers), whereas in ATS, token dropping is applied in all layers past the third encoder. Combining IRET and ATS could potentially yield higher accuracy. This approach, alongside the exploration of various thresholds, learnable thresholds, and the integration of ATS with other pruning techniques, will be a focus of our future work. As shown, IRET initially has slightly lower accuracy than ATS and DeiT without token dropping. However, with the implementation of the Focus concept and increased token dropping, IRET achieves better accuracy than ATS and DeiT at similar FLOP counts for higher attention thresholds. IRET's consistent architecture and the FLOP reduction achieved solely through threshold control, coupled with its superior accuracy in lower FLOP count regions, positions it as an efficient solution for edge applications balancing accuracy with computational complexity, enabling its use in energy and latency-sensitive applications.

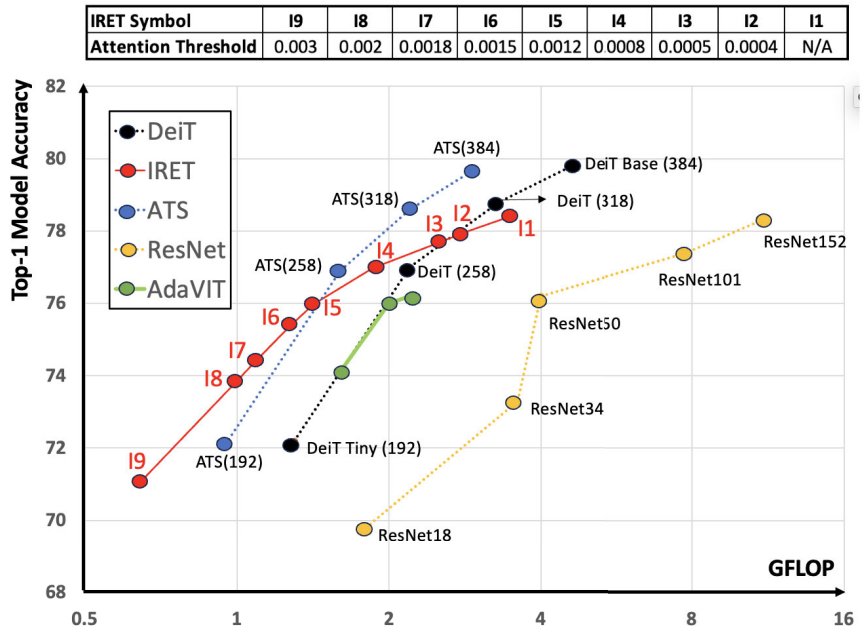


FIGURE 19. Comparing the tradeoff between accuracy and flop count in IRET with that of prior art solutions. Adopting the concept of Focus allows the IRET to enjoy a gentler drop in accuracy while increasing the attention threshold used for token dropping and focusing.

TABLE 2. IRET with multi-head architecture Top-1 accuracy of heads.

Attention Threshold	$head_4$	$head_5$	$head_6$	$head_7$	$head_8$	$head_9$	$head_{10}$	$head_{11}$	$head_{12}$
IRET(Base)	49.592	56.866	63.976	73.68	75.768	76.708	77.442	77.862	78.12
0.0004	49.446	56.604	63.574	72.826	75.346	76.482	77.148	77.562	77.86
0.0005	49.438	56.356	63.234	72.494	75.114	76.204	76.932	77.384	77.68
0.0008	49.168	55.952	62.294	71.728	74.7	75.73	76.466	76.768	76.98
0.0012	48.448	54.978	60.52	70.466	73.426	74.606	75.32	75.674	75.98
0.0015	48.326	54.428	59.314	69.624	72.746	73.976	74.62	75	75.3
0.0018	48.004	53.552	57.85	68.766	71.87	72.93	73.73	74.08	74.36
0.002	47.706	52.798	56.936	68.058	71.212	72.264	73.146	73.446	73.76
0.003	47.352	50.198	53.192	65.308	68.424	69.694	70.438	70.74	71.11

TABLE 3. IRET with multi-head architecture GFLOPS count based on different confidence threshold.

Attention Threshold	$C_t = 60$	$C_t = 65$	$C_t = 70$	$C_t = 75$	$C_t = 80$	$C_t = 85$	$C_t = 90$	$C_t = 95$	$C_t = 100$
IRET(Base)	1.002	1.284	1.487	1.782	2.737	3.444	3.499	3.514	3.515
0.0004	0.898	1.122	1.294	1.530	2.287	2.772	2.812	2.819	2.819
0.0005	0.852	1.050	1.195	1.410	2.166	2.474	2.505	2.511	2.511
0.0008	0.742	0.890	0.984	1.170	1.675	1.836	1.853	1.856	1.856
0.0012	0.670	0.780	0.857	1.027	1.365	1.410	1.419	1.421	1.421
0.0015	0.658	0.730	0.798	0.964	1.250	1.270	1.277	1.278	1.278
0.0018	0.638	0.684	0.728	0.855	1.070	1.080	1.084	1.085	1.085
0.002	0.640	0.679	0.737	0.874	1.054	1.064	1.066	1.067	1.067
0.003	0.519	0.537	0.585	0.629	0.645	0.646	0.646	0.646	0.646

A. MUTIHEADED IRET ENERGY-AWARE BEHAVIOR

Table 2 shows the Top-1 accuracy of MLP heads for different variants of IRET. As you can see from $head_7$ we have an acceptable prediction accuracy for all variants of IRET, so the model gives us the possibility to do the prediction earlier, especially for simple images, and avoid the computation of final layers of the model and save power and energy. For the real-time scenario, with a preset deadline, the model can do the prediction with high confidence and meet the deadline. The proposed multi-headed architecture provides an energy-effective solution. This means that for some applications using this architecture, we can do earlier predictions when we reach the desired confidence level. We have tested the effect of different confidence values on IRET. Fig. 20 shows the behavior of the IRET variant with attention threshold = 0.0004 based on different confidence values. As the figure shows for smaller confidence values like 60 the model can do the prediction for all the images using heads 4-6 so the

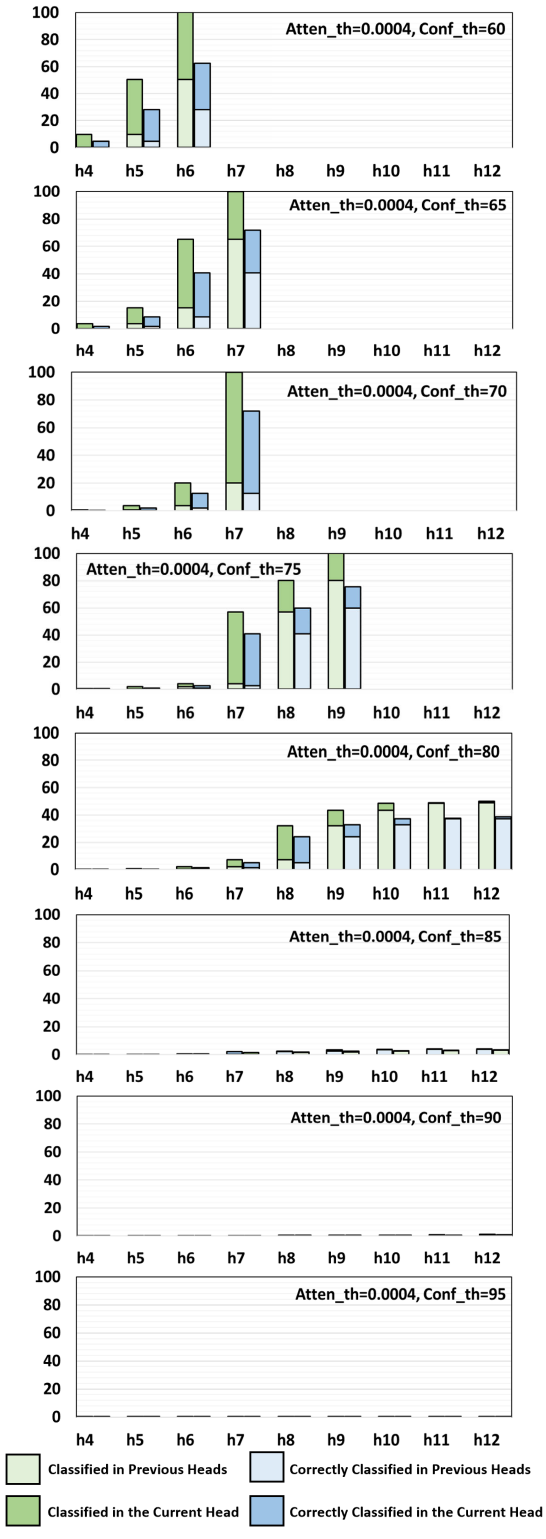


FIGURE 20. The effect of different confidence thresholds on the multi-exit IRET. The attention threshold is 0.0004.

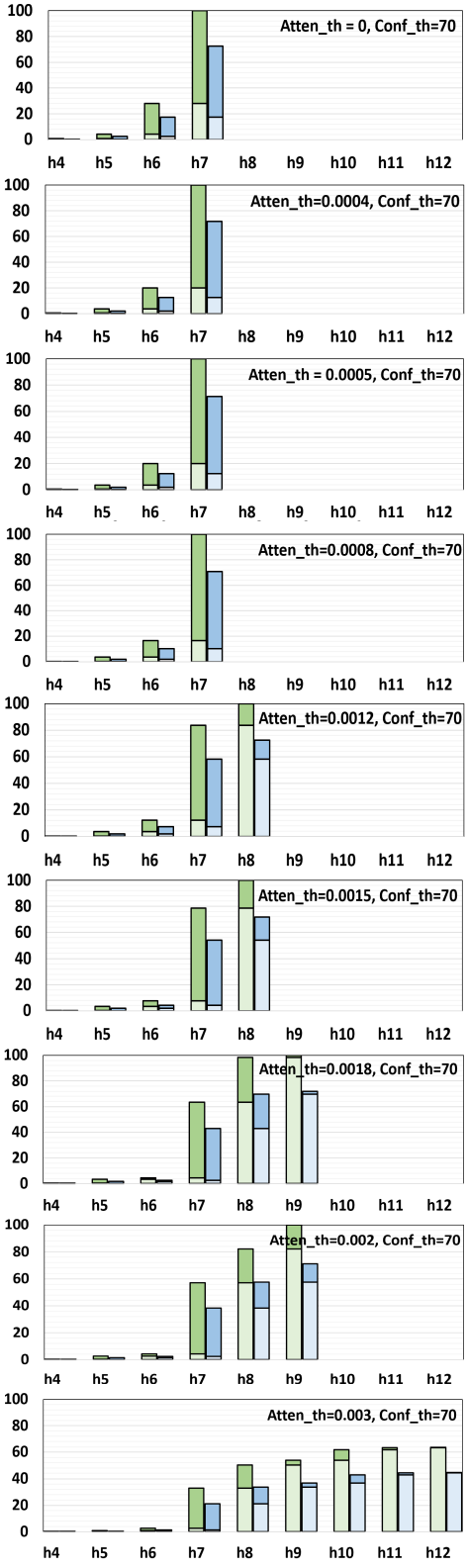


FIGURE 21. The effect of different attention thresholds on the multi-exit IRET. The confidence threshold is 70.

computation of the reset of layers is skipped. The dark orange color shows the percentage of classified images using the current head and the light orange shows the percentage of classified images in the previous layers. But these predictions are based on the confidence threshold so not all of the images are classified correctly. The dark gray shows the percentage

of the correctly classified images in each head and the light gray shows the percentage of the correctly classified images in previous heads. By increasing the confidence threshold the ability of the model to do the prediction in earlier layers decreases and the models switch to later layers for prediction so the computation requirement of the model and as a result of that the energy requirement of it increases by increasing the confidence. Moreover, the gap between the images classified in each head and the number of correctly classified images in each head decreases. Fig. 21 shows the effect of different Attention threshold values for a constant confidence threshold = 70. By increasing the Attention threshold more tokens drop so the model needs more layers to extract the features and do the classification. In both experiments, some of the images remain unclassified. The gap between 100 and the last bar shows the number of remaining images that are not classified. Table 3 shows the flop count of the different variants of multi-head IRET based on different confidence values. For smaller confidence values the GFLOPS of the model reduces significantly. With a smaller confidence value, the model can perform the prediction at an earlier head and avoid the rest of the computation. However, increasing confidence needs more understanding of the image features by the model and more number layers and as a result more computation.

VI. CONCLUSION

In this study, we introduced the IRET encoder, a novel encoder layer that not only drops unattended tokens but also enhances the model's focus on attended ones using incremental input sampling and increased embedding size. IRET transformer, constructed using a mix of IRET and basic transformer encoders. Based on the choice of attention threshold for token dropping and token focusing, IRET allows us to trade accuracy for computational complexity. The IRET's ability to focus on attended tokens using incremental input sampling allows a more graceful degradation in accuracy in the result of dropping tokens compared to prior art solutions. Notably, its computational complexity is modulated through attention threshold adjustments, rather than changes in embedding size or model architecture. The combination of this unique feature alongside early exiting renders IRET ideal for applications requiring a balance between accuracy, energy efficiency, and latency considerations.

While IRET offers promising advancements, there are areas for further improvement. Future work will explore incorporating token pruning mechanisms, such as mixing IRET with ATS, to enhance accuracy. Additionally, we aim to investigate new attention computation methods to further optimize performance. To address practical implementation challenges, we plan to pursue a hardware-software co-design approach, exploring both the hardware and algorithmic aspects to make IRET more efficient and scalable across diverse deployment scenarios. Moreover, we plan to leverage the IRET architecture for tasks that demand high-resolution inputs or involve datasets with limited structural regularities. We will assess its performance and fine-tune it to ensure optimal functioning for these specific applications.

REFERENCES

[1] *Pytorch Wavelets*. Accessed: Mar. 1, 2024. [Online]. Available: https://pytorch-wavelets.readthedocs.io/en/latest/readme.html

[2] J. Alman and Z. Song, "Fast attention requires bounded entries," in *Advances in Neural Information Processing Systems*, vol. 36, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., Red Hook, NY, USA: Curran Associates, 2023, pp. 63117–63135.

[3] A. Bakhtiarnia, Q. Zhang, and A. Iosifidis, "Multi-exit vision transformer for dynamic inference," 2021, *arXiv:2106.15183*.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th Eur. conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Jan. 2020, pp. 213–229.

[5] J. Choi, S. Lee, J. Chu, M. Choi, and H. J. Kim, "Vid-TLDR: Training free token merging for light-weight video transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 18771–18781.

[6] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, "Rethinking attention with performers," 2020, *arXiv:2009.14794*.

[7] M. Dehghani, B. Mustafa, J. Djolonga, J. Heek, M. Minderer, M. Caron, A. Steiner, J. Puigcerver, R. Geirhos, I. M. Alabdulmohsin, A. Oliver, P. Padlewski, A. Gritsenko, M. Lucic, and N. Houlsby, "Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–23.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Jan. 2018.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[11] M. Elbayad, J. Gu, E. Grave, and M. Auli, "Depth-adaptive transformer," 2019, *arXiv:1910.10073*.

[12] H. Falahati, M. Sadrosadati, Q. Xu, J. Gómez-Luna, B. S. Latibari, H. Jeon, S. Hesaabi, H. Sarbazi-Azad, O. Mutlu, M. Annavaram, and M. Pedram, "Cross-core data sharing for energy-efficient GPUs," *ACM Trans. Archit. Code Optim.*, vol. 21, no. 3, pp. 1–32, Sep. 2024.

[13] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6804–6815.

[14] M. Fayyaz, S. A. Koohpayegani, F. R. Jafari, S. Sengupta, H. R. V. Joze, E. Sommerlade, H. Pirsiavash, and J. Gall, "Adaptive token sampling for efficient vision transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jan. 2022, pp. 396–414.

[15] Z. Fei, X. Yan, S. Wang, and Q. Tian, "DeeCap: Dynamic early exiting for efficient image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12206–12216.

[16] D. Han, X. Pan, Y. Han, S. Song, and G. Huang, "FLatten transformer: Vision transformer using focused linear attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 5938–5948.

[17] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 15908–15919.

[18] A. Hatamizadeh, G. Heinrich, H. Yin, A. Tao, J. M. Alvarez, J. Kautz, and P. Molchanov, "FasterViT: Fast vision transformers with hierarchical attention," 2023, *arXiv:2306.06189*.

[19] J. D. Havtorn, A. Royer, T. Blankevoort, and B. E. Bejnordi, "MSViT: Dynamic mixed-scale tokenization for vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 838–848.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[21] X. He, I. Keivanloo, Y. Xu, X. He, B. Zeng, S. Rajagopalan, and T. Chilimbi, "Magic pyramid: Accelerating inference with early exiting and token pruning," 2021, *arXiv:2111.00230*.
[22] W. Hua, Z. Dai, H. Liu, and Q. V. Le, "Transformer quality in linear time," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 9099–9117.
[23] B. Kang, X. Chen, D. Wang, H. Peng, and H. Lu, "Exploring lightweight hierarchical vision transformers for efficient visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 9612–9621.
[24] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are RNNs: Fast autoregressive transformers with linear attention," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2020, pp. 5156–5165.
[25] M. Kim, S. Gao, Y.-C. Hsu, Y. Shen, and H. Jin, "Token fusion: Bridging the gap between token pruning and token merging," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 1372–1381.
[26] S. Kim, S. Shen, D. Thorsley, A. Gholami, W. Kwon, J. Hassoun, and K. Keutzer, "Learned token pruning for transformers," 2021, *arXiv:2107.00910*.
[27] S. Abbasi Koohpayegani and H. Pirsiavash, "SimA: Simple softmax-free attention for vision transformers," 2022, *arXiv:2206.08898*.
[28] S. Lee, J. Choi, and H. J. Kim, "Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 15741–15750.
[29] G. Li, Z. Cui, M. Li, Y. Han, and T. Li, "Multi-attention fusion transformer for single-image super-resolution," *Sci. Rep.*, vol. 14, no. 1, p. 10222, May 2024.
[30] X. Li, Y. Shao, T. Sun, H. Yan, X. Qiu, and X. Huang, "Accelerating BERT inference for sequence labeling via early-exit," 2021, *arXiv:2105.13878*.
[31] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," 2022, *arXiv:2202.07800*.
[32] Y. Liang, Z. Wang, X. Xu, Y. Tang, J. Zhou, and J. Lu, "MCUFormer: Deploying vision tranformers on microcontrollers with limited memory," in *Advances in Neural Information Processing Systems*, vol. 36, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., Red Hook, NY, USA: Curran Associates, 2023, pp. 8501–8512.
[33] K. Liao, Y. Zhang, X. Ren, Q. Su, X. Sun, and B. He, "A global past-future early exit method for accelerating inference of pre-trained language models," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2021, pp. 2013–2023.
[34] W. Lin, Z. Wu, J. Chen, J. Huang, and L. Jin, "Scale-aware modulation meet transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6015–6026.
[35] S. Liu, G. Tao, Y. Zou, D. Chow, Z. Fan, K. Lei, B. Pan, D. Sylvester, G. Kielian, and M. Saligane, "ConSmax: Hardware-friendly alternative softmax with learnable parameters," 2024, *arXiv:2402.10930*.
[36] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "EfficientViT: Memory efficient vision transformer with cascaded group attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14420–14430.
[37] Y. Liu, C. Matsoukas, F. Strand, H. Azizpour, and K. Smith, "PatchDropout: Economizing vision transformers using patch dropout," 2022, *arXiv:2208.07220*.
[38] Y. Liu, Q. Zhou, J. Wang, Z. Wang, F. Wang, J. Wang, and W. Zhang, "Dynamic token-pass transformers for semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 1816–1825.
[39] Y. Liu, M. Gehrig, N. Messikommer, M. Cannici, and D. Scaramuzza, "Revisiting token pruning for object detection and instance segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2646–2656.
[40] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
[41] L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.-G. Jiang, and S.-N. Lim, "AdaViT: Adaptive vision transformers for efficient image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12299–12308.
[42] Y. Mo, P. Zuo, Q. Zhou, Z. Mo, Y. Fan, S. Zhang, and B. Kang, "PWLT: Pyramid window-based lightweight transformer for image classification," *Comput. Electr. Eng.*, vol. 116, May 2024, Art. no. 109209.
[43] T. M. Nguyen, V. Suliafu, S. Osher, L. Chen, and W. Bao, "FMMformer: Efficient and flexible transformer via decomposed near-field and far-field attention," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 29449–29463.
[44] I. Ntinoutl, E. Sanchez, and G. Tzimiropoulos, "Multiscale vision transformers meet bipartite matching for efficient single-stage action localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 18827–18836.
[45] Z. Pan, J. Cai, and B. Zhuang, "Fast vision transformers with HiLo attention," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022.
[46] Z. Qin, W. Sun, H. Deng, D. Li, Y. Wei, B. Lv, J. Yan, L. Kong, and Y. Zhong, "CosFormer: Rethinking softmax in attention," 2022, *arXiv:2202.08791*.
[47] M. M. Rahman and R. Mărculescu, "Multi-scale hierarchical vision transformer with cascaded attention decoding for medical image segmentation," in *Medical Imaging With Deep Learning* (Proceedings of Machine Learning Research), vol. 227, I. Oguz, J. Noble, X. Li, M. Styner, C. Baumgartner, M. Rusu, T. Heinmann, D. Kontos, B. Landman, and B. Dawant, Eds., Jul. 2024, pp. 1526–1544.
[48] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C. Hsieh, "DynamicViT: Efficient vision transformers with dynamic token sparsification," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 13937–13949.
[49] M. X. Bastidas Rodriguez, A. Gruson, L. F. Polanía, S. Fujieda, F. P. Ortiz, K. Takayama, and T. Hachisuka, "Deep adaptive wavelet network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 3100–3108.
[50] Y. Song, Q. Zhou, X. Li, D.-P. Fan, X. Lu, and L. Ma, "BA-SAM: Scalable bias-mode attention mask for segment anything model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 3162–3173.
[51] J. R. Stevens, R. Venkatesan, S. Dai, B. Khailany, and A. Raghunathan, "Softermax: Hardware/software co-design of an efficient softmax for transformers," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, Dec. 2021, pp. 469–474.
[52] T. Sun, Y. Zhou, X. Liu, X. Zhang, H. Jiang, Z. Cao, X. Huang, and X. Qiu, "Early exiting with ensemble internal classifiers," 2021, *arXiv:2105.13792*.
[53] S. Tang, Y. Wang, Z. Kong, T. Zhang, Y. Li, C. Ding, Y. Wang, Y. Liang, and D. Xu, "You need multiple exiting: Dynamic early exiting for accelerating unified vision language model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10781–10791.
[54] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, and C. Zheng, "Synthesizer: Rethinking self-attention for transformer models," in *Proc. Int. Conf. Mach. Learn.*, May 2021, pp. 10183–10192.
[55] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. conf. Mach. Learn.*, Jul. 2021, pp. 10347–10357.
[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.
[57] I. Vasyltsov and W. Chang, "Efficient softmax approximation for deep neural networks with attention mechanism," 2021, *arXiv:2111.10770*.
[58] A. Vyas, A. Katharopoulos, and F. Fleuret, "Fast transformers with clustered attention," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 21665–21674.
[59] H. Wang, B. Dedhia, and N. K. Jha, "Zero-TPrune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16070–16079.
[60] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.
[61] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.
[62] W. Wang, W. Chen, Q. Qiu, L. Chen, B. Wu, B. Lin, X. He, and W. Liu, "CrossFormer++: A versatile vision transformer hinging on cross-scale attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3123–3136, May 2024.

[63] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16×16 words: Dynamic transformers for efficient image recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 11960–11973.
[64] R. Wightman. (2019). *Pytorch Image Models*. [Online]. Available: https://github.com/rwightman/pytorch-image-models
[65] M. Wortsman, J. Lee, J. Gilmer, and S. Kornblith, "Replacing softmax with ReLU in vision transformers," 2023, *arXiv:2309.08586*.
[66] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," 2021, *arXiv:2108.09084*.
[67] J. Wu, B. Duan, W. Kang, H. Tang, and Y. Yan, "Token transformation matters: Towards faithful post-hoc explanation for vision transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 10926–10935.
[68] C. Xia, X. Wang, F. Lv, X. Hao, and Y. Shi, "ViT-CoMer: Vision transformer with convolutional multi-scale feature interaction for dense predictions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 5493–5502.
[69] T. Xiao, Y. Li, J. Zhu, Z. Yu, and T. Liu, "Sharing attention weights for fast transformer," 2019, *arXiv:1906.11024*.
[70] J. Xin, R. Tang, J. Lee, Y. Yu, and J. Lin, "DeeBERT: Dynamic early exiting for accelerating BERT inference," 2020, *arXiv:2004.12993*.
[71] J. Xin, R. Tang, Y. Yu, and J. Lin, "BERxiT: Early exiting for BERT with better fine-tuning and extension to regression," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 91–104.
[72] G. Xu, J. Hao, L. Shen, H. Hu, Y. Luo, H. Lin, and J. Shen, "LGViT: Dynamic early exiting for accelerating vision transformer," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 9103–9114.
[73] X. Xu, C. Li, Y. Chen, X. Chang, J. Liu, and S. Wang, "No token left behind: Efficient vision transformer via dynamic token idling," in *Proc. Australas. Joint Conf. Artif. Intell.* Cham, Switzerland: Springer, Jan. 2023, pp. 28–41.
[74] Y. Xu, Z. Zhang, M. Zhang, K. Sheng, K. Li, W. Dong, L. Zhang, C. Xu, and X. Sun, "Evo-ViT: Slow-fast token evolution for dynamic vision transformer," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, Jun. 2022, pp. 2964–2972.
[75] T. Yao, Y. Pan, Y. Li, C.-W. Ngo, and T. Mei, "Wave-ViT: Unifying wavelet and transformers for visual representation learning," 2022, *arXiv:2207.04978*.
[76] H. Yin, A. Vahdat, J. M. Alvarez, A. Mallya, J. Kautz, and P. Molchanov, "A-ViT: Adaptive tokens for efficient vision transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10799–10808.
[77] H. You, Y. Xiong, X. Dai, B. Wu, P. Zhang, H. Fan, P. Vajda, and Y. C. Lin, "Castling-ViT: Compressing self-attention via switching towards linear-angular attention at vision transformer inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14431–14442.
[78] S. Yun and Y. Ro, "SHViT: Single-head vision transformer with memory efficient macro design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 5756–5767.
[79] Q. Zhang, J. Zhang, Y. Xu, and D. Tao, "Vision transformer with quadrangle attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3608–3624, May 2024.
[80] Y. Zhang, D. Chen, S. Kundu, C. Li, and P. A. Beerel, "SAL-ViT: Towards latency efficient private inference on ViT using selective attention search with a learnable softmax approximation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 5116–5125.
[81] Y. Zhang, Y. Liu, D. Miao, Q. Zhang, Y. Shi, and L. Hu, "MG-ViT: A multi-granularity method for compact and efficient vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–20.
[82] Z. Zhang, H. Zhang, L. Zhao, T. Chen, S. Ö. Arık, and T. Pfister, "Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, Jun. 2022, pp. 3417–3425.
[83] W. Zhou, C. Xu, T. Ge, J. McAuley, K. Xu, and F. Wei, "BERT loses patience: Fast and robust inference with early exit," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 18330–18341.
[84] W. Zhu, "LeeBERT: Learned early exit for BERT with cross-level optimization," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 2968–2980.

BANAFSHEH SABER LATIBARI received the B.Sc. degree in computer engineering from the K. N. Toosi University of Technology, in 2014, and the M.Sc. degree in computer architecture from the Sharif University of Technology, in 2017. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at Davis (UC Davis). Her research interests include computer vision, machine learning, security, and hardware design.

HOUMAN HOMAYOUN received the B.S. degree in electrical engineering from the Sharif University of Technology, in 2003, the M.S. degree in computer engineering from the University of Victoria, in 2005, and the Ph.D. degree in computer science (CS) from the University of California at Irvine (UCI), in 2010. He is currently working as a Professor with the Department of Electrical and Computer Engineering (ECE), University of California at Davis (UC Davis). Prior to that, he was an Associate Professor at the Department of ECE, George Mason University (GMU). From 2010 to 2012, he spent two years at the University of California at San Diego, as an NSF Computing Innovation Fellow (CIFellow) awarded by CRA-CCC. He is the Director of UC Davis Accelerated, Secure, and Energy-Efficient Computing Laboratory (ASEEC). He conducts research in hardware security and trust, data-intensive computing, and heterogeneous computing.

AVESTA SASAN (Senior Member, IEEE) received the B.Sc. degree in computer engineering and the M.Sc. and Ph.D. degrees in electrical and computer engineering (ECE) from the University of California at Irvine (UCI), in 2005, 2006, and 2010, respectively. In 2010, he joined the Office of CTO, Broadcom Company, working on the physical design and implementation of ARM processors, working as a Physical Designer, a Timing Signoff Specialist, and the Lead of signal and power integrity signoff in this team. In 2014, he was recruited by Qualcomm Office of VLSI Technology, where he developed different methodologies and in-house EDAs for accurate signoff and analysis of hardened ASIC solutions. He joined the Department of ECE, George Mason University, in 2016, while simultaneously working as an Associate Chair for Research with the Department of ECE. In 2021, he joined as a Faculty Member with the Department of ECE, University of California at Davis. His research interests include hardware security, machine learning hardware, efficient learning on edge, low-power design, approximate computing, and the IoT.