IRET: Incremental Resolution Enhancing Transformer

Banafsheh Saber Latibari*, Soheil Salehi†, Houman Homayoun*, Avesta Sasan*
*University of California, Davis, CA, USA
†University of Arizona, Tucson, AZ, USA
{bsaberlatibari,hhomayoun,asasan}@ucdavis.edu
ssalehi@arizona.edu

ABSTRACT

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. This is especially useful for edge devices, where accuracy and complexity could be dynamically traded based on factors such as battery life, reliability, etc.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

Vision Transformer, Token Dropping, Attention, Focus

ACM Reference Format:
Banafsheh Saber Latibari, Houman Homayoun, and Avesta Sasan. 2024. IRET: Incremental Resolution Enhancing Transformer In *GLSVLSI '24: Great Lakes Symposium on VLSI, June 12–15, 2024, Tampa, Florida, USA. ACM, New York, NY, USA, 6 pages.*

1 INTRODUCTION

Recent advancements in deep learning and GPU capabilities 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. 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, primarily due to their reliance on complex global attention mechanisms and MLPs. To mitigate these demands, various strategies like downsampling, token dropping, early prediction, and softmax elimination have been researched. These approaches are summarized in Section 3. We briefly review these solutions in Section 3. 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 enhance their efficiency with minimal accuracy impact, broadening their application scope. We present the Incremental Resolution Enhancing Transformer (IRET), a transformative model architecture that employs attention-based input sampling. 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.

2 BACKGROUND

Fig. 1.(left) shows the Visual Transformer (ViT) [6] 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 [21]. 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 [5], 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

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3660380

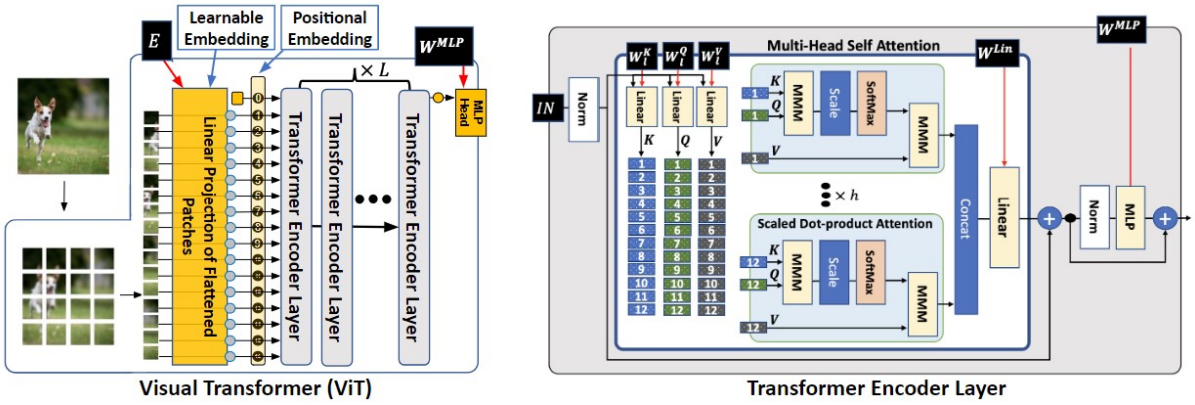

Figure 1: (Left): Overall structure of original Visual Transformer (ViT) in [6]. (right): Encoder solution used in ViT, illustrating the implementation details of Multi-Head Self Attention (MSA) from h scaled dot-product attention units.

y. Layernorm (LN) is applied before and residual connections after every block.

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

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

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

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

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; ....; x_p^N E] + E_{pos}, \quad (4)$$

$$E \in R^{(P^2.C)} \times D, E_{pos}$$

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1}, \qquad l = 1...L \quad (5)$$

$$z_l = MLP(LN(z'_l)) + z'_l, \qquad l = 1...L \quad (6)$$

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

The classification head is attached to z_L^0 and implemented by an MLP with one hidden layer at pre-training and a 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 [2] exploits a pure Transformer to create an end-to-end object detection framework. Taking a different approach, DeiT [20] enhances ViT by introducing the distillation token, and leverages a teacher model to decrease the necessary training data.

3 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.

Multiscale Viosn Transformers: To reduce vision transformer complexity, a pyramid-style processing approach has been adopted, processing input images at multiple scales to capture key contextual information [7]. Models like Pyramid Vision Transformer [23], Swin Transformer [14], Multi-scale Vision Transformer (MViT) [7], PVT v2 [24], and Wave-ViT [27] are based on this strategy. PVT utilizes a convolution-free pyramid structure with spatial-reduction attention, while Swin Transformer features a hierarchical architecture with shifted windows to manage complexity. MViT adapts multi-scale features, altering resolution and channel dimensions for detailed pattern recognition.

Patch and Token Pruning: Various studies have explored addressing attention matrix sparsity in transformer models by implementing token pruning methods to boost efficiency. These methods fall into two primary categories: consistent approaches across input types and adaptive strategies based on input characteristics. DynamicViT by Rao et al. [17] introduces a predictive module for token importance scoring and hierarchical pruning using Gumbel-Softmax. Wang et al. [25] developed DVT, which employs early exiting and feature reuse for varying token counts in images. Fayyaz et al. [8] presented ATS for dynamic token selection using attention scores. Liu et al. [13] discussed PatchDropout for training standard ViT models efficiently, and Meng et al. [15] introduced AdaViT for autonomous patch, self-attention head, and layer utilization. Yin et al. [28] proposed A-ViT with adaptive inference to optimize resource usage. Our approach, while similar to this group, is unique in incorporating a focus concept. It starts with smaller image embeddings to reduce encoding layer complexity and progressively increases the embedding size of attended tokens based on attention scores. Simultaneously, it drops unattended tokens, thereby boosting both the efficiency and effectiveness of the model.

Early Termination: Researchers have developed the concept of anytime prediction in computer vision by adding early-exit branches to models, particularly useful for IoT applications with variable latency constraints [1, 10]. This idea has been adapted to transformers, with depth-adaptive transformers by Elbayad et al. allowing predictions at different network stages. He et al. introduced Magic Pyramid (MP) which combines token pruning with early exit strategies for computational efficiency [9]. Liao et al. proposed a global early exit approach that leverages information from multiple layers [12]. The Dynamic Transformer by Wang et al. autonomously adjusts the number of tokens for processing images, enabling flexible inference based on prediction confidence [25]. Bakhtiarnia et al. explored seven early exit designs in vision transformer backbones, optimizing the trade-off between accuracy and inference speed for tasks like image classification and crowd counting [1]. Our approach is orthogonal to this solution.

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 [3, 11, 19, 22]. 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 [16]. This approach is also orthogonal to our approach.

Our approach to managing transformer model complexity differs markedly from existing methods. We employ a unique strategy of incrementally supplying information to the model, controlled by its attention mechanism. Adjusting the attention threshold allows us to dynamically balance focus and forgetfulness, achieving a crucial balance between accuracy and complexity, especially for edge applications. Our solution is also compatible with, and can enhance, previously discussed methods. The specifics of our technique are elaborated in the next section.

4 IRET: PROPOSED METHOD

4.1 Architecture of IRET

The high-level architecture of IRET is shown in Fig. 2. The innovation in IRET is the ability to focus on attended tokens in addition to forgetting unattended tokens. As illustrated in Fig. 2, IRET replaces several transformer encoder layers with IRET encoders. The architecture of an IRET encoder is shown in Fig. 3. IRET encoder pre-processes the tokens for token dropping and token focusing before performing the encoding. More specifically, similar to prior work in [8, 17], IRET performs the token dropping based on CLS token attention scores, dropping tokens with low attention scores to prune the computational tree.

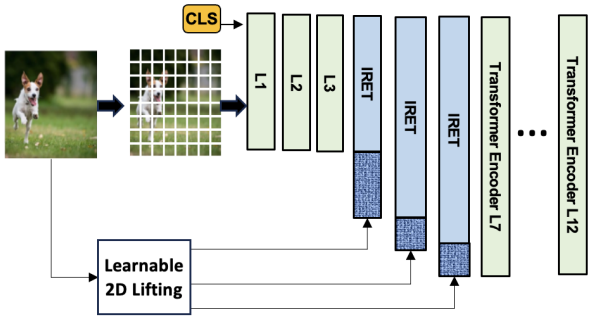

Figure 2: The IRET architecture processes input through four sampling steps: initially with a scaled low-pass filter and then three times using learnable 2-D lifting schemes. With each IRET layer, the embedding size of each token increases as it assimilates additional information. Concurrently, before 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.

However, as illustrated in Fig. 3 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 [18] 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 each patch in the newly sampled sub-band to be ignored or forwarded to the linear projection unit for embedding. If the corresponding token coming from the previous encoder has an attention score above desired threshold, the

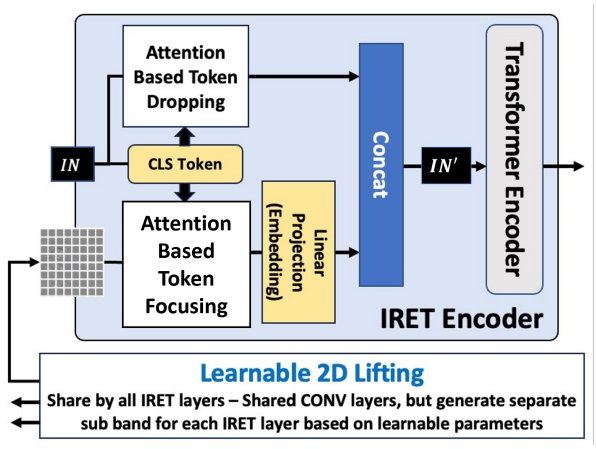

Figure 3: The IRET layer architecture utilizes the CLS token to identify unattended tokens, employing a token dropping method to remove them. Additionally, it determines which tokens require more focus based on the CLS token. This process involves filtering patches from the input sample created by the 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.

patch is deemed useful and is subjected to embedding. The embedded information for each sub-band patch 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 patch. The size of each encoder layer in Fig. 2 corresponds to the embedding size of its token. Using this illustration, as shown in Fig. 2, each IRET encoder layer (shown in blue) increases the embedding size of tokens (shown in dark blue), while each regular transformer encoder layer maintains the embedding size.

The architecture of the 2D-lifting scheme [18] used in the IRET layer is shown in Fig. 4. 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 signal x is partitioned into two components: an even component and an odd component. In following equations ${x_o}^{L_U}[n] = x_o[n - L_U]$, and $x_o[n - L_U + 1], ..., x_o[n - L_U - 1], x_o[n + L_U]$ are the sequence of $2L_U + 1$ adjacent odd polyphase samples of $x_e[n]$. In the prediction stage $c^{L_P}[n] = c[n - L_P], c[n - L_P + 1], ..., c[n + L_P - 1], c[n + L_P]$ are a sequence of $2L_P + 1$ approximation coefficients.

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

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

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

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

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

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

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. This approach allows each IRET layer to adaptively incorporate new and unique features, differentiating them from previously sample 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 the their embedding. Prior to adopting learnable layers, we explored different sampling techniques for the input image, like discrete wavelet transformation (DWT), using each sub-band as a separate input to the feature encoding layer. However, our findings indicated that a learnable lifting scheme, which learns features based on model loss and trained alongside the main model, yields the highest accuracy.

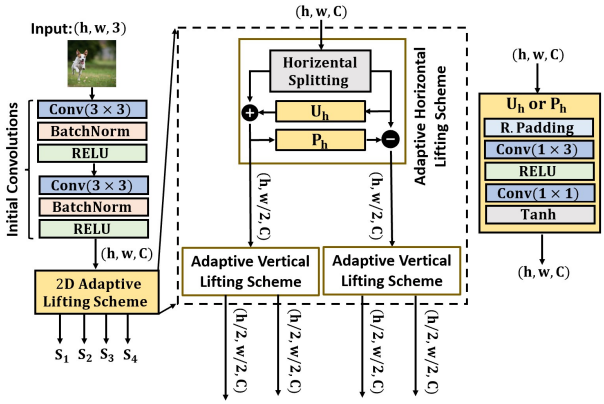

Figure 4: The Architecture of Learnable 2D Lifting Scheme. It receives the original image and learns four output samples. S_1, S_2 and, S_3 are used in the architecture of IRET.

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 number of tokens and working with a smaller number of tokens 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. 5 visualizes the pre-processing function for token dropping and token focusing in an IRET encoder layer. The first row of the attention matrix corresponds to the CLS token. The CLS token is the token used in the last layer of transformers for classification. Hence, the attention that CLS pays to other tokens, reflects the importance of each token. 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

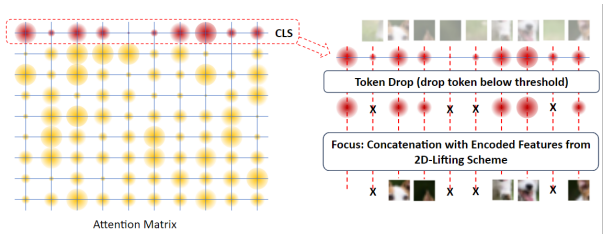

Figure 5: The IRET layer features two main functions: attention-based token dropping and focusing. It eliminates unattended tokens using attention scores to simplify computation and enlarges the embedding size for attended tokens with extra features from a 2-D lifting scheme. This process, akin to human brain focusing, allows IRET to selectively prioritize certain tokens, thereby boosting accuracy and lowering computational complexity.

through the use of a 2D-lifting scheme. As illustrated in Fig. 5, an IRET layer first drops unattended tokens, and then increases the embedding size to attended tokens analogous to increasing focus. Fig 6 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.

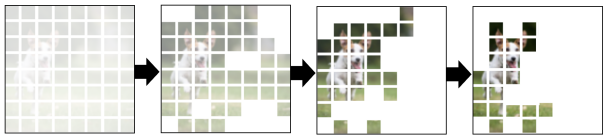

Figure 6: In IRET, the 'forget and focus' concept hinges on CLS token attention values. Tokens with attention below a threshold are dropped ('forget'), while those above the threshold see increased embedding size ('focus') via a 2D-lifting scheme. Concept of focus is shown by increased resolution.

5 EXPERIMENTS

Our model was developed based on the Facebook DeiT [20] small model with hard distillation, utilizing the Timm library [26]. We conducted our experiments on the ImageNet dataset [4] using Nvidia A100 as the training platform. The model inputs are 112×112 pixels, with IRET layers receiving 112×112 sub-bands generated by the 2D-lifting scheme [18]. 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 in the first and second IRET layers, and 4) Including all three sub-band samples.

Fig. 7 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

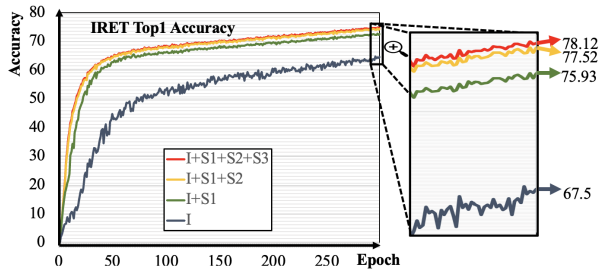

Figure 7: The top-one training accuracy of the IRET model, highlighting its improved accuracy with additional input samples, is evaluated across four scenarios: 1) Only the input image used (I), 2) Image plus one sub-band sample in the first IRET layer used (I+S1), 3) Two sub-band samples in the first and second IRET layers (I+S1+S2) used, and 4) All three sub-band samples (I+S1+S2+S3) used.

boosts accuracy, but subsequent samples yield lesser improvements. This observation made us to limit the number of IRET layers. The embedding size distribution across sub-bands also affects incremental learning rate and final accuracy. Due to space constraints, this paper introduces the concept and reserves detailed design space exploration for future work. As shown, the model's top-1 accuracy improves from 67.5% to 75.93%, 77.52%, and 78.12% with the addition new information extracted from sampled sub-bands. As previously mentioned, 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.

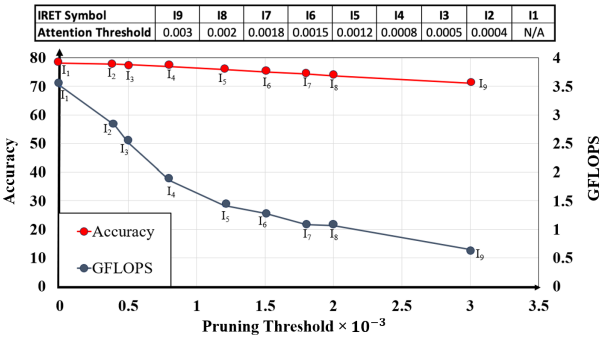

Figure 8: Change in accuracy and flop count as a function of attention threshold for token dropping and token focusing.

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 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. 8 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.

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

Figure 9 shows the average pruning results based on the different threshold values and the number of dropped tokens inside each layer averaged over ImageNet Test set. 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 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.

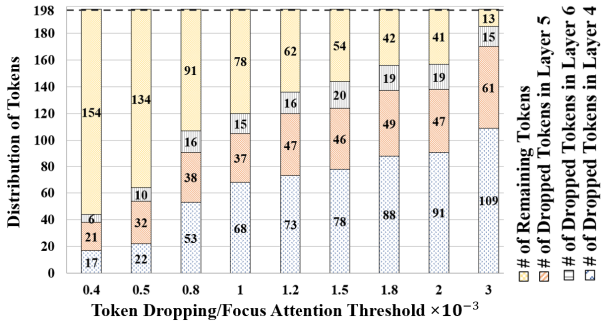

Figure 9: Token dropping of layers with pruning policy based on different threshold values. For smaller number of thresholds the model drops fewer tokens.

Fig. 10 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, DeiT, ResNet, and AdaVIT 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's 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.

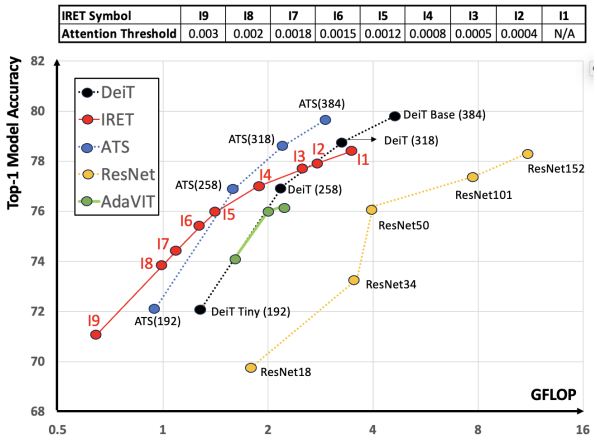

Figure 10: Comparing the tradeoff between accuracy and flop count in IRET with that of prior art solutions. Adopting the concept of Focus allows the IRET to enjoy a gentler drop in accuracy while increasing the attention threshold used for token dropping and focusing.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under award #2203399.

6 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. This unique feature renders IRET ideal for applications needing to balance accuracy with energy and latency considerations.

REFERENCES

[1] Arian Bakhtiarnia et al. 2021. Multi-exit vision transformer for dynamic inference. *arXiv preprint arXiv:2106.15183* (2021).
[2] Nicolas Carion et al. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European conf. , Glasgow, UK, August 23–28, 2020, Proc. , Part I 16*. Springer, 213–229.
[3] Krzysztof Choromanski et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794* (2020).
[4] Jia Deng et al. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conf. on computer vision and pattern recognition*. Ieee, 248–255.
[5] Jacob Devlin et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]
[6] Alexey Dosovitskiy et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
[7] Haoqi Fan et al. 2021. Multiscale vision transformers. In *Proc. of the IEEE/CVF Int. conf. on Computer Vision*. 6824–6835.
[8] Mohsen Fayyaz et al. 2022. Adaptive token sampling for efficient vision transformers. In *European conf. on Computer Vision*. Springer, 396–414.
[9] Xuanli He et al. 2021. Magic pyramid: Accelerating inference with early exiting and token pruning. *arXiv preprint arXiv:2111.00230* (2021).
[10] Gao Huang et al. 2017. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844* (2017).
[11] Soroush Abbasi Koohpayegani et al. 2022. SimA: Simple Softmax-free Attention for Vision Transformers. *arXiv preprint arXiv:2206.08898* (2022).
[12] Kaiyuan Liao et al. 2021. A global past-future early exit method for accelerating inference of pre-trained language models. In *Proc. of the 2021 conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013–2023.
[13] Yue Liu et al. 2022. PatchDropout: Economizing Vision Transformers Using Patch Dropout. *arXiv preprint arXiv:2208.07220* (2022).
[14] Ze Liu et al. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proc. of the IEEE/CVF Int. conf. on Computer Vision (ICCV)*.
[15] Lingchen Meng et al. 2022. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *Proc. of the IEEE/CVF conf. on Computer Vision and Pattern Recognition (CVPR)*. 12309–12318.
[16] Zhen Qin et al. 2022. cosFormer: Rethinking Softmax in Attention. *arXiv preprint arXiv:2202.08791* (2022).
[17] Yongming Rao et al. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems* 34 (2021), 13937–13949.
[18] Maria Ximena Bastidas Rodriguez et al. 2020. Deep adaptive wavelet network. In *Proc. of the IEEE/CVF Winter conf. on Applications of Computer Vision*. 3111–3119.
[19] Jacob R. Stevens et al. 2021. Softermax: Hardware/Software Co-Design of an Efficient Softmax for Transformers. In *2021 58th ACM/IEEE Design Automation conf. (DAC)*. 469–474. https://doi.org/10.1109/DAC18074.2021.9586134
[20] Hugo Touvron et al. 2021. Training data-efficient image transformers & distillation through attention. In *Int. conf. on machine learning*. PMLR, 10347–10357.
[21] Ashish Vaswani et al. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[22] Ihor Vasyltsov et al. 2021. Efficient Softmax Approximation for Deep Neural Networks with Attention Mechanism. *arXiv preprint arXiv:2111.10770* (2021).
[23] Wenhai Wang et al. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proc. of the IEEE/CVF Int. conf. on computer vision*. 568–578.
[24] Wenhai Wang et al. 2022. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media* 8, 3 (2022), 415–424.
[25] Yulin Wang et al. 2021. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems* 34 (2021), 11960–11973.
[26] Ross Wightman. 2019. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models. https://doi.org/10.5281/zenodo.4414861
[27] Ting Yao et al. 2022. Wave-ViT: Unifying Wavelet and Transformers for Visual Representation Learning. *arXiv preprint arXiv:2207.04978* (2022).
[28] Hongxu Yin et al. 2022. A-ViT: Adaptive Tokens for Efficient Vision Transformer. In *Proc. of the IEEE/CVF conf. on Computer Vision and Pattern Recognition (CVPR)*.