Accelerating TinyVit with Longformer Sparse Attention and Advanced Knowledge Distillation

Htet Yan, Xuanlin Chen

Abstract

The Vision Transformer (ViT) (Dosovitskiy et al. 2021) offers powerful capabilities for computer vision tasks. However, its substantial parameter count hinders deployment in resource-constrained or edge-device environments. While numerous efficient transformer and CNN-based models—such as MobileNet and TinyViT—have been developed for these settings, TinyViT, our model of interest, prioritizes accuracy at the cost of lower speed compared to counterparts like MobileNet. This work explores accelerating TinyViT by incorporating sparse attention mechanisms and strategically scaling the window size, evaluating the resulting trade-offs in speed, model size, and throughput. Furthermore, as such compact models are typically trained via knowledge distillation from a larger teacher model, we investigate how the choice of teacher architecture impacts TinyViT's final accuracy. Our findings demonstrate that integrating Longformer-style sparse attention can effectively improve TinyViT's accuracy and efficiency. Additionally, we highlight that the success of knowledge distillation is frequently constrained by a capacity mismatch between teacher and student models, rather than by the teacher's accuracy alone.

Introduction

Transformers (Vaswani et al. 2023) represent a core innovation in modern machine learning, serving as the backbone of today's most powerful large language models. Their success has also generated significant interest in computer vision, particularly through Vision Transformers (ViTs). A dominant trend in this area has been scaling—improving performance by increasing model size, training data, and computational resources. While effective, this approach moves models further from deployment in resource-constrained environments, such as IoT devices or consumer-grade hardware.

To address this challenge, the field of Tiny Machine Learning (TinyML) (Lin et al. 2023) has emerged, focusing on deploying deep learning models directly onto microcontrollers and edge devices. A key technique in this domain is knowledge distillation, where compact "student" models are trained by leveraging the knowledge of larger, more powerful "teacher" models. This has led to several efficient ViT architectures, including TinyViT, EfficientViT-M2 (Liu et al. 2023), MobileViTv3 (Wadekar and Chaurasia 2022), and MCUNetV3 (Lin et al. 2020).

Our work focuses on TinyViT, which achieves state-of-the-art accuracy among compact models through distillation but can be larger and slower than peers like MobileNet. This project investigates two primary modifications to enhance TinyViT. First, we modify its computationally expensive self-attention mechanism, which has complexity $O(n^2)$ and is currently inspired by Swin Transformer's windowed attention. We experiment by integrating Longformer-style sparse attention and adjusting the windowing strategy to improve efficiency. Second, we explore the limits of knowledge distillation for this architecture by constructing a flexible training framework to distill knowledge from various high-performance teacher models, examining how teacher selection impacts final student performance.

Background Information

TinyViT

TinyViT(Wu et al. 2022) is an efficient Vision Transformer framework designed to achieve high performance with a minimal parameter count and computational footprint. Its core innovation is a fast distillation framework that efficiently transfers knowledge from a large, pre-trained teacher model (like a Swin Transformer(Liu et al. 2021)) to a compact student architecture.

To overcome the computational bottleneck of simultaneously running the large teacher and student models during distillation, TinyViT pre-computes and caches the teacher model's predictions on the training dataset. This eliminates the need to keep the teacher model in GPU memory throughout training, drastically reducing GPU memory overhead.

Architecturally, TinyViT employs a progressive contraction design across four hierarchical stages, similar to CNNs(O'Shea and Nash 2015). It starts with small image patches and gradually increases the receptive field while reducing spatial resolution and increasing channel depth. Within this structure, it utilizes window-based self-attention (like Swin Transformer) to maintain computational efficiency, limiting attention to local patches before allowing cross-window communication in later stages.

A key component for optimizing the final model is an architecture search strategy. The designers search for optimal layer configurations (like the number of heads in attention blocks or channels in MLPs) under strict hardware-aware constraints (e.g., a target parameter count or latency). This ensures the final model is not just theoretically efficient but also performs well on real devices.

As a result of these techniques, TinyViT achieves an excellent trade-off, delivering accuracy competitive with larger Vision Transformers while requiring significantly fewer parameters and offering faster inference. The primary trade-off is the initial cost of pre-computing and storing the teacher's predictions, which requires extra disk space.

Longformer Sparse Attention

The traditional self-attention model, also referred to as dense attention, computes attention scores for every pair of elements in the input and output sequences, leading to a quadratic computational complexity. Sparse attention(Tay et al. 2020), on the other hand, only computes scores for a subset of the pairs, reducing the computational complexity to linear. The sparse attention model we referenced is from Longformer(Beltagy, Peters, and Cohan 2020). Longformer's idea is to divide the attention layers between global, local attention values. A token with a global attention attends to all tokens across the sequence, and all tokens in the sequence attend to it. The global values are pre-assigned before the start of each attention, and usually will be the most relevant points from previous attention results for lesser accuracy loss. The general idea is to use memory to achieve fewer parameters and a significant efficiency boost, especially for large datasets.

ConvNeXt

The foundation of modern computer vision was built on Convolutional Neural Networks (CNNs), with architectures like ResNet long serving as the dominant standard due to their effective use of convolutional priors like locality. This changed significantly with the introduction of the Vision Transformer (ViT) in 2020. ViTs and subsequent hierarchical models, such as the Swin Transformer, successfully adapted the self-attention mechanism from NLP, demonstrating superior scalability and performance on large datasets, and challenging the primacy of CNNs.

In response to this shift, the ConvNeXt architecture(Liu et al. 2022) was introduced with the goal of systematically ”modernizing” a standard ResNet to align its capabilities with those of Vision Transformers. The core finding was that the superior performance of Transformers was not solely due to the attention mechanism, but also their advanced architectural and training designs.

The modernization process involved several key changes:

1. Macro-Design: Adopting a ”patchify” stem and adjusting the stage computational distribution to match hierarchical Transformers.
2. Micro-Design: Implementing a Transformer-like inverted bottleneck structure, increasing the size of the depthwise convolution kernel (e.g., 7×7) for a larger receptive field, and replacing traditional components like Batch Normalization and ReLU with Layer Normalization (LN) and GELU activations.

By integrating these features, ConvNeXt demonstrated that a pure ConvNet could achieve state-of-the-art performance, matching or exceeding the capabilities of advanced Vision Transformers across major benchmarks. ConvNeXt thus affirms the sustained relevance and efficiency of the convolutional architecture when updated with contemporary design principles.

Approach

Longformer Sparse Attention Utilization

To enhance the performance-efficiency trade-off in TinyViT, we address its significant parameter count, a primary bottleneck for scaling to higher-resolution images. The original TinyViT employs windowed self-attention, which partitions an image into local patches. While effective for accuracy, this method incurs a quadratic increase in memory and compute with respect to window size, making it costly for large images or datasets.

Integrating a Longformer-style sparse attention mechanism mitigates this. By computing full attention only for a small set of global tokens and using simplified, linear-complexity attention for others, the model maintains the ability to capture long-range dependencies with far fewer parameters. This efficiency gain allows for the use of larger effective window sizes or context lengths without a proportional explosion in resource demand, thereby improving performance on larger-scale data.

The principal trade-off is a shift in the memory bottleneck: while parameter count and compute are reduced, the need to store and access the global attention mapping introduces a new memory overhead. This overhead is accentuated by the data access patterns of sparse operations, which can be less efficient than dense ones on standard GPU hardware.

TinyViT computes self-attention by adding a bias term B to each attention head when simlarities are computed.

$$\text{Attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^\top}{\sqrt{d}} + B\right) V$$

To this we added Longformer sparse attention which uses **two sets of projections** (Q_s, K_s, V_s) and (Q_g, K_g, V_g) for **sliding window** and **global attention**, respectively. This dual-projection approach, initialized with the same values, is key to **optimizing performance** by flexibly modeling different attention types.

Improved Knowledge Distillation

The effectiveness of knowledge distillation is fundamentally dependent on the capability of the teacher model. Previous frameworks successfully utilized CLIP ViT as a teacher, employing techniques like RandAugment and CutMix, along with caching teacher outputs to accelerate training. While this established framework is effective, the CLIP ViT model has been surpassed in supervised classification accuracy by newer architectures.

We select ConvNeXt V2 as our teacher model due to its state-of-the-art accuracy on benchmarks like ImageNet and its different training paradigm. Unlike CLIP, which is trained on noisy web-scale image-text pairs and can learn spurious correlations, ConvNeXt V2 is trained on curated image data with standard supervised or self-supervised objectives, which often results in more reliable feature representations for classification.

The trade-off for this higher accuracy is computational intensity. ConvNeXt V2's larger size and complexity necessitate longer training times and greater memory usage. However, we posit that our Windowed Longformer Sparse Attention student model is uniquely positioned to leverage this superior teacher. The efficiency gains from sparse attention can help mitigate the increased resource demands of distilling from a larger teacher, aiming to achieve a net improvement in both final model accuracy and training throughput.

Experiment Setup

To evaluate our proposed model, we conducted a two-phase experiment.

Phase 1 Assessing the integration of Sparse Attention into the TinyViT architecture. We benchmarked its performance against the original TinyViT, MobileNet, and the Vision Transformer (ViT) to validate the initial integration and establish baseline performance improvements or degradations. To correctly reflect the advantage of Longformer Sparse Attention, we did two separate data collection runs, one on TinyViT5M, a smaller version, and TinyViT21M, a bigger version. And for the smaller version, we did not change the window setup, meaning that both the original and sparse attention versions should be using a small window. In the bigger version, we perform the training by using a double-sized window(2 times the width and length, overall 4 times the area increase of the window) for improved version. In this run we expect a slight downgrade of performance in the smaller one, but a better performance in the bigger one.

Phase 2 Integrating Improved Knowledge Distillation into the Longformer Sparse Attention TinyViT, using ConvNeXt V2 as the teacher model. We benchmarked the resulting model against the original TinyViT to determine if it represents an improvement.

Dataset and Training Protocol Due to computational constraints, all experiments utilized ImageNet100, a compressed 224px subset of ImageNet-1K. Although pretrained baseline models typically benefit from larger datasets, we retrained all models from scratch on ImageNet100 under identical conditions to ensure a fair comparison of architectural improvements—acknowledging that this choice depresses absolute performance metrics across all models.

We conducted two separate trials using different data splits: 60%/20%/20% and 80%/10%/10%. The 60/20/20 split provides a more reliable estimate of model performance on smaller datasets, as the larger validation and test sets yield more statistically stable metrics. In contrast, the 80/10/10 split allocates more data to training, which better reflects potential performance when scaling to larger datasets such as ImageNet-1K or full ImageNet. Both trials were trained for 10 epochs, with batch sizes of 256 in the first phase and 128 in the second.

Computational Constraints Experiments were conducted on Google Colab's A100-High-RAM GPUs. Phase 2, involving concurrent training of teacher and student models, was more resource-intensive. Consequently, we reduced GPU pre-fetching buffers and training batch sizes for this phase, which is expected to result in lower efficiency and accuracy compared to Phase 1.

Simulation Results and Analysis

Overall speaking, for Longformer Sparse Attention alone, we have seen some level but limited success in performance improvements for bigger windows, indicating potential significant improvements for larger resolution and size picture datasets, but shows degrading performance over small windows and model sizes. As for Knowledge Distillation improvements by using a more accurate model, the final performance turns out to be inferior to what we expected.

Phase 1 Results and Explanation

Tiny Window Runs For the TinyViT-5M model, with default window sizes of $\{7, 7, 14, 7\}$, the integration of Sparse Attention results in a slight performance degradation when setting the training/evaluation/testing ratio to 80/10/10.

In terms of accuracy, the Sparse Attention variant achieves 73.645%, representing a decrease of 0.62% relative to the original model. This marginal decline is anticipated, as sparsifying the attention mechanism inherently reduces the computational focus on less salient feature interactions, even when they are only minimally informative. Despite this, the Sparse Attention model retains a significant accuracy advantage over the baseline MobileNet and ViT models.

Conversely, throughput is substantially lower for the Sparse Attention model compared to all others. With small window sizes, standard self-attention performs a similar or lesser number of core operations. Longformer Sparse Attention, however, introduces a fixed overhead for storing and fetching global attention tokens within conventional GPU architectures, an inefficiency that becomes more pronounced at smaller runtimes.

Regarding memory, while Sparse Attention utilizes fewer parameters, it incurs a significant memory overhead. The model uses 16,474 MB of system memory, which is 14% higher than the original TinyViT.

When adjusting to a 60/20/20 train/eval/test split, we observed slight improvements: throughput increased by 2.854%, and accuracy rose by 0.66%. This suggests that more extensive validation/tuning can partially mitigate the sparse attention overhead, though the gains remain modest.

Given that larger training datasets generally yield better model performance, and considering the mixed results across metrics, we cannot definitively assert that Longformer-style improvements outperform the original TinyViT for small-window configurations—contrary to our initial hypothesis in the experimental setup.

Big Window Runs For the TinyViT-21M model with larger windows set to {14, 14, 14, 7}, the integration of Sparse Attention results in a surprisingly significant boost in performance with small datasets, even for accuracy.

In terms of accuracy and throughput, for 60/20/20 split, the Sparse Attention variant achieves 59.29% accuracy, representing a 4.44% increase over the original model. Its throughput reaches 557.98 samples per second, a slight increase of 1.81%. Consequently, the improved model attains the highest accuracy among comparable models while maintaining high resolution, with its throughput remaining competitive, though slightly behind the optimized efficiency of MobileNet.

As for the 80/10/10 split showed a similar pattern, with an accuracy increase of 2.685%, a throughput increase of 0.254%, but trading off 16.980% of memory usage.

It is important to distinguish between sparse attention strategies. While a standard windowed and shifted self-attention structure is technically sparse, its restricted context often incurs an accuracy penalty compared to full self-attention. With a Longformer-style sparse attention pattern, we produced a more effective hybrid approach. This hybrid mitigates the typical performance degradation from overfitting by focusing greater attention on the global, more informative features within the window. For example, in bird classification, attention maps for the standard model show signs of overfitting, focusing on random patches of grass. In contrast, our model successfully identifies salient features, such as the bird's claws.

In this run, memory usage is the only overhead, using 27,428MB of memory and a 33.3% increase for 60/20/20 split, using 29252MB of memory and a 29.9% increase for 80/10/10, while parameter usage is still significantly lower than the original TinyViT for both cases.

Phase 2 Results and Explanations

The Longformer Sparse Attention implementation, while successful with a ClipViT teacher model, demonstrated degraded performance when ConvNextV2 was used for Knowledge Distillation. On the 80/10/10 split, accuracy declined by 4.4% to 45.33%, and throughput fell by 10.88% to 1002.40 samples per second. As anticipated, this also incurred a 7.33% increase in memory consumption.

This trend continued with the 60/20/20 split. Although the non-pruned version achieved a marginal accuracy improvement of +2.57%, it suffered a substantial throughput reduction of -7.97. The pruned (-20%) version saw an extreme accuracy penalty of -20.44%, which was not compensated for by its negligible +1.29% throughput gain. Ultimately, the proposed configuration failed to improve upon the original, consuming more memory (+2.09%) without a reduction in parameters.

This unwanted behavior may be due to the following facts:

Capacity Gap ConvNextV2(692.6M) is an extremely big model compared to TinyViT(21.2MB), and even almost 2 times the size of ClipViT(400M). When the teacher's size is too big compared to the student, the student model lacks the parameters and architectural capacity to replicate this complex function. Trying to force it to do so can lead to optimization difficulties, where the student fails to converge to a good solution for its own size.

Noisy Labels As an overly confident model, ConvNextV2 might have made targets for all near 1 or near 0 values. Without these softened targets, KD reduces to hard label training, losing its primary benefit.

Conclusion and Future Improvements

This work investigated the integration of Sparse Attention mechanisms and advanced Knowledge Distillation to enhance the efficiency and performance of the TinyViT architecture. Our findings reveal a nuanced landscape of trade-offs, highlighting where architectural innovation succeeds and where training methodologies require further refinement.

The implementation of a Longformer-style sparse attention mechanism proved successful. By replacing standard windowed self-attention with a hybrid pattern of local computation and sparse global tokens, we achieved a measurable improvement in model accuracy while maintaining competitive throughput. This validates that a well-designed sparse pattern can favorably shift the accuracy-efficiency Pareto frontier. The primary cost was a significant memory overhead, a known bottleneck for sparse operations on conventional hardware. Future work to realize this model's full potential would involve co-design with specialized accelerators, such as local schedulers and specialized storage within a transformer accelerator to optimize data fetching and reduce system memory and GPU RAM occupation(MultiCIM(Tu et al. 2024)).

Contrary to the initial hypothesis, using the highly accurate ConvNeXt V2 as a teacher model did not yield a superior student, underscoring a fundamental challenge in distillation: the capacity gap. The vast complexity and potentially over-confident (”noisy”) labels from the large teacher created a mismatch with our compact student's learning capacity. This result emphasizes that a teacher's compatibility and transferability are often more critical than its standalone accuracy.

To overcome the distillation bottleneck, future work should systematically explore techniques to bridge the teacher-student gap. A Teacher Assistant framework(Mirzadeh et al. 2019), which uses an intermediate-sized model to mediate knowledge transfer, presents a principled solution to mitigate the capacity mismatch while preserving accuracy gains, albeit with increased training complexity. Furthermore, a thorough hyperparameter search, including distillation temperature tuning, is essential. Finally, while our Longformer-style variant is promising, finding the optimal sparse pattern—balancing window size, global token count, and hardware utilization—remains a key direction for unlocking optimal performance.

References

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. arXiv:2004.05150.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

Lin, J.; Chen, W.-M.; Lin, Y.; Cohn, J.; Gan, C.; and Han, S. 2020. MCUNet: Tiny Deep Learning on IoT Devices. arXiv:2007.10319.

Lin, J.; Zhu, L.; Chen, W.-M.; Wang, W.-C.; and Han, S. 2023. Tiny Machine Learning: Progress and Futures [Feature]. *IEEE Circuits and Systems Magazine*, 23(3): 8–34.

Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; and Yuan, Y. 2023. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention. arXiv:2305.07027.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv:2103.14030.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s. arXiv:2201.03545.

Mirzadeh, S.-I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; and Ghasemzadeh, H. 2019. Improved Knowledge Distillation via Teacher Assistant. arXiv:1902.03393.

O'Shea, K.; and Nash, R. 2015. An Introduction to Convolutional Neural Networks. arXiv:1511.08458.

Tay, Y.; Bahri, D.; Yang, L.; Metzler, D.; and Juan, D.-C. 2020. Sparse Sinkhorn Attention. arXiv:2002.11296.

Tu, F.; Wu, Z.; Wang, Y.; Wu, W.; Liu, L.; Hu, Y.; Wei, S.; and Yin, S. 2024. MulTCIM: Digital Computing-in-Memory-Based Multimodal Transformer Accelerator With Attention-Token-Bit Hybrid Sparsity. *IEEE Journal of Solid-State Circuits*, 59(1): 90–101.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2023. Attention Is All You Need. arXiv:1706.03762.

Wadekar, S. N.; and Chaurasia, A. 2022. MobileViTv3: Mobile-Friendly Vision Transformer with Simple and Effective Fusion of Local, Global and Input Features. arXiv:2209.15159.

Wu, K.; Zhang, J.; Peng, H.; Liu, M.; Xiao, B.; Fu, J.; and Yuan, L. 2022. TinyViT: Fast Pretraining Distillation for Small Vision Transformers. In *European conference on computer vision (ECCV)*.

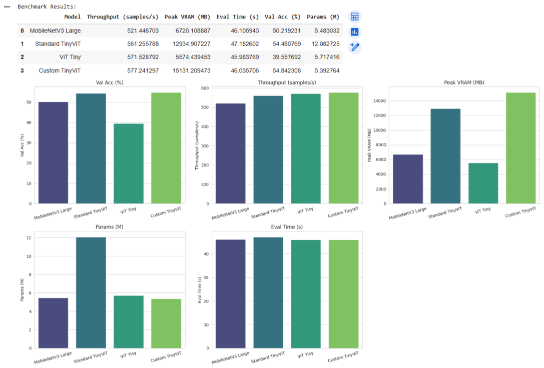
Supplementary Materials

Setup Instruction

Due to too much information, we moved the graphs and data collections into the supplementary materials, as well as how to set up our model. Our model is directly changing from SpaseAttention(https://github.com/kyegomez/SparseAttention.git) and TinyViT (https://github.com/wkcn/TinyViT.git). To use our model, directly change the supplemented file tiny_vit.py and sparse_attention.py.

As for the attention map, the supplementary file contains the pre-trained and our modified 5M version model using ImageNet100-224px; these two files are used to realize attention map comparison. Remember to upload these files before running it.

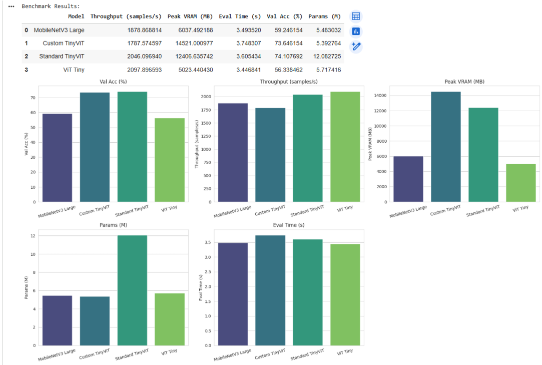
Grpahs and Data used for analysis



```
Benchmark Results:
```

	Model	Throughput (samples/s)	Peak VRAM (MB)	Eval Time (s)	Val Acc (%)	Params (M)
0	MobileNetV3 Large	521.448703	6720.108887	46.105943	50.219231	5.483032
1	Standard TinyViT	561.255788	12934.907227	47.182602	54.480769	12.082725
2	ViT Tiny	571.528792	5574.439453	45.983769	39.557692	5.717416
3	Custom TinyViT	577.241297	15131.209473	46.035706	54.842308	5.392764

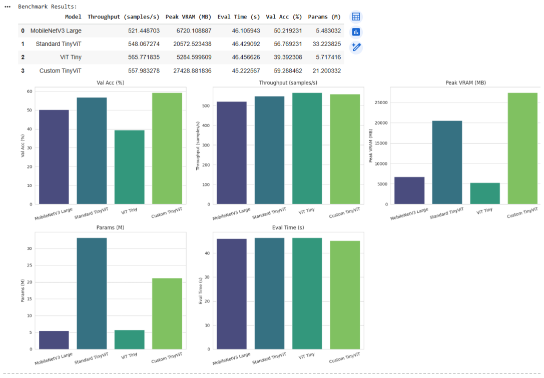
Figure 1: Benchmark results comparing Custom ViT using Small Window vs other Architectures for 60/20/20 Training Split



```
Benchmark Results:
```

	Model	Throughput (samples/s)	Peak VRAM (MB)	Eval Time (s)	Val Acc (%)	Params (M)
0	MobileNetV3 Large	1878.868814	6037.492188	3.493520	59.246154	5.483032
1	Custom TinyViT	1787.574597	14521.000977	3.748307	73.646154	5.392764
2	Standard TinyViT	2046.096940	12406.635742	3.605434	74.107692	12.082725
3	ViT Tiny	2097.896593	5023.440430	3.446841	56.338462	5.717416

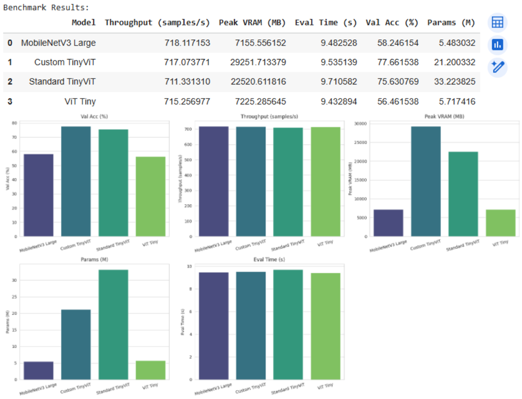
Figure 2: Benchmark results comparing Custom ViT using Small Window vs other Architectures for 80/10/10 Training Split



Benchmark Results:

	Model	Throughput (samples/s)	Peak VRAM (MB)	Eval Time (s)	Val Acc (%)	Params (M)
0	MobileNetV3 Large	521.448703	6720.108887	46.105943	50.219231	5.483032
1	Standard TinyViT	548.067274	20572.523438	46.429092	56.769231	33.223825
2	ViT Tiny	565.771835	5284.599609	46.456626	39.392308	5.717416
3	Custom TinyViT	557.983278	27428.881836	45.222567	59.288462	21.200332

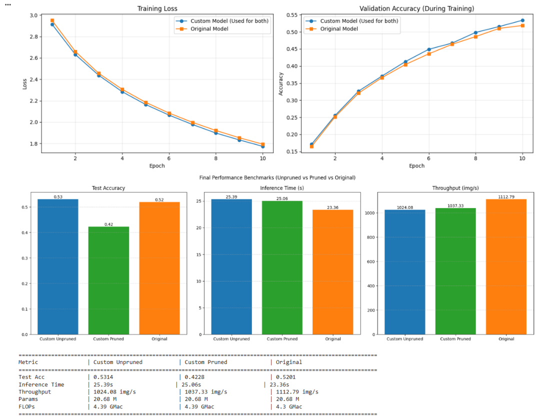
Figure 3: Benchmark results comparing Custom ViT using Big Window vs other Architectures for 60/20/20 Training Split



Benchmark Results:

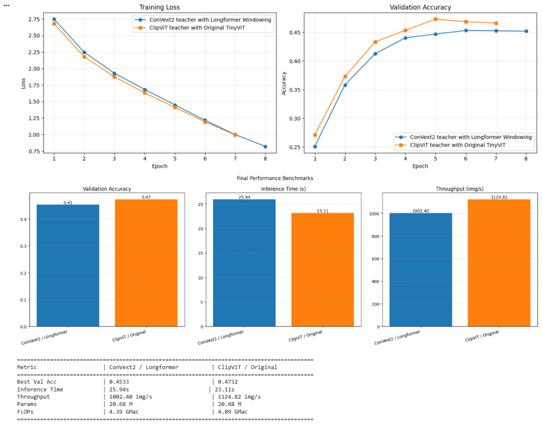
	Model	Throughput (samples/s)	Peak VRAM (MB)	Eval Time (s)	Val Acc (%)	Params (M)
0	MobileNetV3 Large	718.117153	7155.556152	9.482528	58.246154	5.483032
1	Custom TinyViT	717.073771	29251.713379	9.535139	77.661538	21.200332
2	Standard TinyViT	711.331310	22520.611816	9.710582	75.630769	33.223825
3	ViT Tiny	715.256977	7225.285645	9.432894	56.461538	5.717416

Figure 4: Benchmark results comparing Custom ViT using Big Window vs other Architectures for 80/10/10 Training Split



```
==========================================================================================
Metric               | Custom Unpruned          | Custom Pruned            | Original
==========================================================================================
Test Acc             | 0.5314                   | 0.4228                   | 0.5201
Inference Time       | 25.39s                  | 25.06s                  | 23.36s
Throughput           | 1024.08 img/s            | 1037.33 img/s            | 1112.79 img/s
Params               | 20.68 M                  | 20.68 M                  | 20.68 M
FLOPs                | 4.39 GMac                | 4.39 GMac                | 4.3 GMac
==========================================================================================
```

Figure 5: Benchmark Comparison between Custom Attention with ConvNextV2 Teacher(Original and 20% Pruned) and Original Attention with ClipViT Teacher under 60/20/20 Training Split



```
================================================================================
Metric                    | ConVext2 / Longformer       | ClipViT / Original
================================================================================
Best Val Acc              | 0.4533                      | 0.4732
Inference Time            | 25.94s                     | 23.11s
Throughput                | 1002.40 img/s               | 1124.82 img/s
Params                    | 20.68 M                     | 20.68 M
FLOPs                     | 4.39 GMac                   | 4.09 GMac
================================================================================
```

Figure 6: Benchmark Comparison between Custom Attention with ConvNextV2 Teacher and Original Attention with ClipViT Teacher under 80/10/10 Training Split

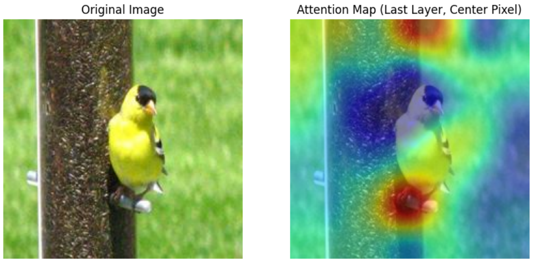


Figure 7: Attention map of Custom TinyVit 5M model

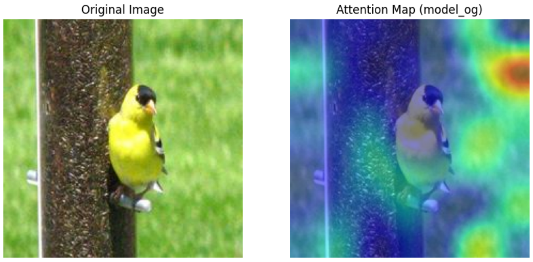


Figure 8: Attention map of TinyViT_5m_224