UNIFORM: A REUSE ATTENTION MECHANISM FOR EFFICIENT TRANSFORMERS ON RESOURCE-CONSTRAINED EDGE DEVICES

Seul-Ki Yeom and Tae-Ho Kim
Nota AI GmbH
Friedrichstrasse 200, 10117 Berlin, Germany
{skyeom,thkim}@nota.ai

ABSTRACT

Transformer-based architectures have demonstrated remarkable success across various domains but remain challenging to deploy on edge devices due to high memory and computational demands. In this paper, we propose *UniForm* (**Uni**fied Trans**form**er), a novel transformer architecture that unifies multi-head attention computations into a shared attention mechanism, *Reuse Attention*, and integrates it into a lightweight, scalable backbone for efficient inference on edge devices, without compromising accuracy. By consolidating redundant operations into a unified representation, UniForm effectively reduces memory overhead and computational complexity, enabling seamless deployment in resource-constrained environments. Experiments on ImageNet-1K and downstream tasks show that UniForm achieves state-of-the-art accuracy while improving inference speed and memory efficiency. Notably, UniForm-l attains 76.7% Top-1 accuracy on ImageNet-1K with a 21.8ms inference time on Jetson Nano, achieving up to a 5x speedup over competing benchmarks. These results highlight UniForm's versatility across GPUs and edge platforms, demonstrating its potential for real-time AI applications in low-resource settings. Code available at https://github.com/seulkiyeom/uniform.

1 INTRODUCTION

Transformers have revolutionized neural networks, excelling in NLP, computer vision, and speech recognition (Vaswani et al., 2017; Devlin et al., 2018). At their core, the Attention mechanism enables effective contextual learning but incurs high computational and memory costs, posing challenges for real-time deployment, especially on resource-constrained edge devices. While Vision Transformers (ViTs) extend Transformers to computer vision, these inefficiencies persist across all Transformer models (Dosovitskiy et al., 2021). Deploying Transformers on devices like Raspberry Pi and Jetson Nano introduces severe constraints such as limited memory, low energy consumption, and strict real-time processing (Yang et al., 2023b). A major bottleneck is the memory bandwidth gap between high-performance GPUs and edge devices, as shown in Figure A.1 (Li et al., 2022). For instance, GPUs like the NVIDIA H100 offer 3.35 TB/s bandwidth, while edge devices such as Raspberry Pi 3B operate at just 17 GB/s—nearly 197 times lower—causing severe memory access inefficiencies. This resource gap significantly impacts inference performance, even with optimized GPUs like the NVIDIA A100 (Liang et al., 2023).

Various optimization techniques, including pruning, quantization, and Neural Architecture Search (NAS), have been explored to mitigate these challenges (Chen et al., 2022a; Jaiswal et al., 2023). However, they fail to address the fundamental memory access inefficiencies in attention mechanisms, which become a primary bottleneck in low-bandwidth environments. Studies such as MCUNet and NAS for efficient edge inference (Lin et al., 2020; Zoph & Le, 2017) show that models optimized for GPUs often struggle to scale effectively on edge hardware, underscoring the need for edge-specific transformer designs.

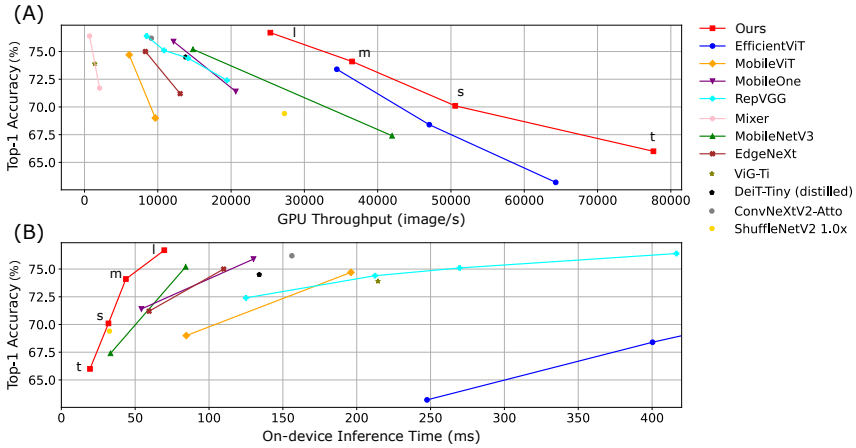

Figure 1: Comparison of speed and accuracy between UniForm and other efficient CNN and ViT models, evaluated using the ImageNet-1K dataset in terms of (A) GPU throughput on an NVIDIA A100 and (B) on-device inference time on a Raspberry Pi 4B.

To address this, we propose *Reuse Attention*, a novel mechanism that reuses a single shared attention matrix across multiple heads, significantly reducing redundant computations and minimizing memory access by up to 80%. Unlike traditional Multi-Head Attention (MHA), which computes independent attention matrices per head, our approach consolidates memory operations and enhances arithmetic intensity, enabling real-time AI inference on low-resource hardware (see Figure A.2). As shown in Figure 1, Reuse Attention maintains strong representational power while substantially improving computational efficiency, making it ideal for edge deployment. Despite these optimizations, our method retains high accuracy across tasks like object detection and segmentation, ensuring broad applicability. Experimental results demonstrate that *UniForm*, our newly designed transformer architecture built around Reuse Attention, achieves a 5× speedup on Jetson Nano while maintaining a new state-of-the-art Top-1 accuracy on ImageNet-1K, outperforming both ViT and CNN-based models in inference efficiency. By reducing memory access by up to 80%, UniForm significantly enhances efficiency and scalability for real-time edge AI applications.

In summary, our contributions are threefold:

- *Efficient Attention Mechanism*: Reduces redundant computations by reusing a shared attention matrix, minimizing memory access, and improving edge-device efficiency.
- *Improved Throughput on Diverse Hardware*: enhances inference speed on both high-performance GPUs and resource-constrained devices, surpassing conventional attention mechanisms.
- *Versatility in Downstream Tasks*: demonstrates strong performance in object detection and segmentation while maintaining efficiency and scalability for real-world applications.

These advancements highlight the potential of Reuse Attention in bridging the gap between high-end GPUs and edge devices, making real-time transformer deployment on constrained hardware a practical reality.

2 Proposed Method

In this section, we introduce Reuse Attention with multi-scale processing, a novel mechanism designed to minimize the memory and computational overheads of conventional Multi-Head Attention (MHA). As illustrated in Figures 2 and A.3, our approach reuses a single unified attention matrix across heads and integrates multi-scale value processing within a hierarchical backbone, substantially improving efficiency while preserving representational capacity.

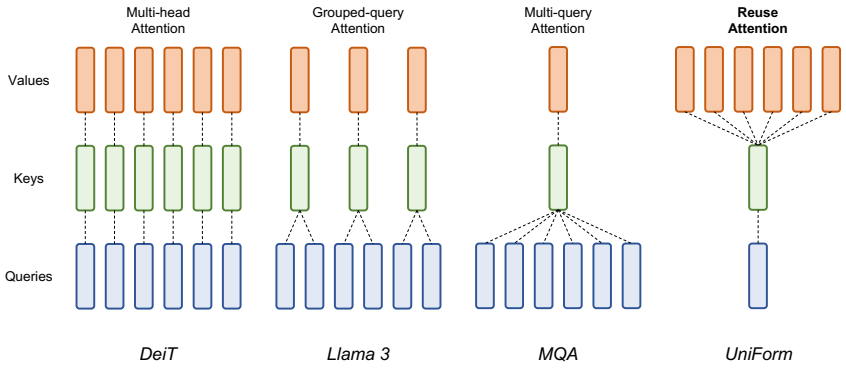

Figure 2: Overview of the Proposed Method compared to the previous attention mechanisms

Mitigating Redundancy in Attention Mechanisms Recent studies have underscored the computational and memory overheads arising from redundant attention maps across heads in Vision Transformers (ViTs). Liu et al. (2023) demonstrated that many heads learn highly similar attention patterns, suggesting that distinct attention matrices for each head may be unnecessary. This observation aligns with findings in Mehta et al. (2021), where fixed or synthetic attention mechanisms sometimes maintained or improved performance, pointing to opportunities for shared attention computations.

Building on these insights, our Reuse Attention mechanism calculates a single shared attention matrix:

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{D}}\right), \tag{1}$$

where $Q = XW^Q$ and $K = XW^K$. By reusing A across all heads, our design circumvents repetitive attention computations, dramatically reducing both arithmetic operations and memory traffic. As shown in Table A.1, this reuse strategy cuts down memory access costs, addressing the bandwidth bottleneck that is particularly problematic on resource-constrained devices.

Enhancing Value Projections with Multi-Scale Processing While queries and keys can often be reduced in dimensionality with minimal accuracy degradation, values typically carry the bulk of feature information. Yang et al. (2023a) found that compressing value representations caused a more severe performance degradation compared to reducing queries or keys. To preserve rich feature information, we retain the original dimensionality of value projections and introduce multi-scale processing through depthwise convolutions:

$$V_h = \text{DWCONV}_{k_h}(X_h W^{V_h}), \tag{2}$$

where each head h employs a distinct kernel size k_h. Inspired by multi-scale designs such as MixConv (Tan & Le, 2019) and Inception modules (Szegedy et al., 2015), this approach enables each head to capture diverse receptive fields without escalating memory bandwidth demands. The shared attention matrix A then aggregates all processed values: $O_h = AV_h$. Finally, the outputs O_h from all heads are concatenated before passing through a final linear projection:

$$\text{Output} = \text{Concat}(O_1, O_2, \ldots, O_H)W^O, \tag{3}$$

This multi-scale design retains high expressiveness while remaining efficient in both computation and memory transfer—an imperative for real-time inference on edge devices.

2.1 Comparative Analysis of Attention Mechanisms

Figure 2 compares our Reuse Attention with existing approaches such as standard MHA (e.g., DeiT), Grouped-Query Attention (e.g., Llama 3), and Multi-Query Attention (e.g., Falcon, PaLM). Unlike these methods, which typically compute separate attention matrices for each head, our design reuses a single attention matrix and focuses on multi-scale value projections. Notably, this kind of attention reuse has shown promise in large language models (LLMs) (Ribar et al., 2024) and speech

Table 1: Performance comparison with the state-of-the art CNN and ViT models on ImageNet-1K.

Model	Base architecture	Top-1 acc. (%) ↑	Top-5 acc. (%) ↑	Throughput (images/s) GPU ↑	CPU ↑	FLOPs (M) ↓	Params (M) ↓
EfficientViT-M0 (Liu et al., 2023)	Transformer	63.2	85.4	64293	450	79	2.3
UniForm-t	**Transformer**	**66.0**	**86.6**	**77625**	**544**	**74**	**1.8**
MobileNetV3-small (Howard et al., 2019)	CNN	67.4	87.4	41965	360	57	2.5
EfficientViT-M1 (Liu et al., 2023)	Transformer	68.4	88.7	47045	220	167	3.0
MobileViT-XXS (Mehta & Rastegari, 2022)	Transformer	69.0	88.9	9663	59	410	1.3
ShuffleNetV2 1.0x (Ma et al., 2018)	CNN	69.4	88.9	27277	138	146	2.3
UniForm-s	**Transformer**	**70.1**	**89.3**	**50582**	**231**	**164**	**2.4**
EdgeNeXt-XXS (Maaz et al., 2022)	Both	71.2	-	13051	121	261	1.3
MobileOne-S0 (Vasu et al., 2023)	CNN	71.4	89.8	20642	26	275	2.1
Mixer-B/16 (Tolstikhin et al., 2021)	MLP	71.7	-	2057	6	12610	59.8
RepVGG-A0 (Ding et al., 2021)	CNN	72.4	-	19450	61	1366	8.3
SHViT (Yun & Ro, 2024)	Transformer	72.8	91.0	33489	143	241	6.3
EfficientViT-M3 (Liu et al., 2023)	Transformer	73.4	91.4	34427	166	263	6.9
ViG-Ti (Han et al., 2022)	GNN	73.9	92.0	1406	6	1300	7.1
UniForm-m	**Transformer**	**74.1**	**91.9**	**36507**	**174**	**251**	**5.6**
RepVGG-A1 (Ding et al., 2021)	CNN	74.4	-	14155	39	2362	12.7
DeiT-Tiny (distilled) (Touvron et al., 2021)	Transformer	74.5	-	13785	63	1085	5.9
MobileViT-XS (Mehta & Rastegari, 2022)	Transformer	74.7	92.3	6098	13	986	2.3
ShuffleNetV2 2.0x (Ma et al., 2018)	CNN	74.9	92.4	12910	67	591	7.4
EdgeNeXt-XS (Maaz et al., 2022)	Both	75.0	-	8312	69	538	2.3
RepVGG-B0 (Ding et al., 2021)	CNN	75.1	-	10868	30	15824	14.3
MobileNetV3-large (Howard et al., 2019)	CNN	75.2	91.3	14798	69	217	5.4
MobileOne-S1 (Vasu et al., 2023)	CNN	75.9	92.5	12150	22	825	4.8
ConvNeXtV2-Atto (Woo et al., 2023)	CNN	76.2	93.0	9120	73	552	3.7
Mixer-L/16 (Tolstikhin et al., 2021)	MLP	76.4	-	688	2	44570	208.2
RepVGG-A2 (Ding et al., 2021)	CNN	76.4	-	8483	20	5123	25.4
UniForm-l	**Transformer**	**76.7**	**93.2**	**25356**	**113**	**467**	**10.0**

recognition (Shim et al., 2023), indicating broader applicability across diverse Transformer-based architectures. By alleviating the high memory bandwidth demand characteristic of MHA, Reuse Attention is especially advantageous in resource-constrained scenarios.

3 Experimental Results

3.1 Image Classification

Performance on High-Performance Hardware (GPU/ CPU) The UniForm models (Tiny, Small, Medium, and Large) consistently surpass state-of-the-art (SOTA) models across different scales, demonstrating superior accuracy and computational efficiency. As shown in Figure 1 and Table 1, UniForm achieves up to **76.7%** Top-1 accuracy while maintaining significantly higher throughput compared to conventional CNN and ViT architectures. Specifically, UniForm-s achieves **70.1%** Top-1 accuracy, outperforming MobileNetV3-small (67.4%), EfficientViT-M1 (68.4%), and MobileViT-XXS (69.0%), while also offering a superior throughput on CPU (**231 images/s**) and GPU (**50,582 images/s** vs. 41,965 images/s for MobileNetV3-small and 47,045 images/s for EfficientViT-M1). Moreover, UniForm-l also achieves **76.7%** accuracy with **25,356 images/s** throughput on GPU, significantly outperforming Mixer-L/16 (688 images/s) and ViG-Ti (1,406 images/s) while maintaining higher accuracy.

This trend is consistent across all UniForm variants, confirming that UniForm not only provides higher accuracy but also achieves faster throughput on both GPU and CPU compared to other models of similar sizes. These results establish UniForm as a compelling choice for both large-scale and edge-device environments, where efficient computation and high accuracy are critical (see also Figure C.1.

Inference Speed and Efficiency on Edge Devices As shown in Table 2 and C.1, UniForm demonstrates remarkable efficiency on resource-constrained devices, significantly reducing inference latency while maintaining competitive accuracy. On the Jetson Nano, UniForm-t achieves a notable **11.9ms** inference time, representing a **5x speedup** over EfficientViT-M0 (56.8ms) while delivering improved accuracy (66.0% vs. 63.2%). Similar trends extend across Raspberry Pi and Jetson platforms, reinforcing UniForm's suitability for edge computing applications. These results highlight UniForm as an optimal solution for real-time deployment scenarios, where computational efficiency and predictive performance are critical (see Figure C.1).

Table 2: Inference speed and accuracy comparison between UniForm and state-of-the-art CNN and ViT models across various edge devices.

Model	Top-1 Accuracy	Jetson-Nano	RaspberryPi4B	RaspberryPi3B Plus	RaspberryPi3B	RaspberryPi2B	Jetson-Xavier	Jetson-Tx2	Jetson-Nx	RaspberryPi5	Jetson AGX Orin
EfficientViT-M0	63.2	56.8	247.6	521.1	528.8	1908.3	10.5	25.3	14.1	93.3	6.0
UniForm-t	**66.0**	**11.9**	**19.4**	**35.7**	**40.5**	**141.7**	**4.4**	**7.7**	**5.8**	**6.4**	**2.0**
MobileNetV3-small	67.4	7.3	33.3	65.4	73.5	233.7	1.9	4.0	2.6	12.2	1.1
EfficientViT-M1	68.4	74.7	400.3	839.1	863.1	3162.2	12.0	32.3	17.8	147.7	6.7
MobileViT-XXS	69.0	75.1	84.6	156.8	172.0	549.4	5.0	10.5	7.1	31.7	2.1
ShuffleNetV2 1.0x	69.4	8.6	32.6	62.2	69.6	224.9	2.1	4.2	3.0	9.7	1.1
UniForm-s	**70.1**	**13.5**	**31.9**	**60.4**	**67.0**	**244.0**	**4.4**	**8.0**	**6.0**	**9.7**	**2.0**
EfficientViT-M2	70.8	84.9	473.1	975.2	985.6	3756.6	13.3	36.4	20.3	158.2	7.1
EdgeNeXt-XXS	71.2	64.6	59.4	105.3	120.0	390.7	4.1	8.5	5.7	22.1	2.0
MobileOne-S0	71.4	8.9	54.2	108.7	122.7	427.5	1.7	3.7	2.7	16.9	0.8
Mixer-B/16	71.7	991.6	987.8	N/A	N/A	N/A	32.8	80.0	50.2	321.1	5.8
RepVGG-A0	72.4	15.0	125.0	249.8	284.8	1205.0	2.8	6.1	3.8	33.9	0.9
EfficientViT-M3	73.4	101.6	568.1	1136.1	1223.9	4764.8	16.5	43.4	24.7	187.0	8.0
ViG-Ti	73.9	76.0	214.4	391.8	430.5	1943.5	14.0	32.7	21.1	98.4	5.3
UniForm-m	**74.1**	**15.8**	**43.7**	**81.9**	**93.3**	**346.7**	**4.3**	**9.9**	**6.7**	**13.7**	**2.0**

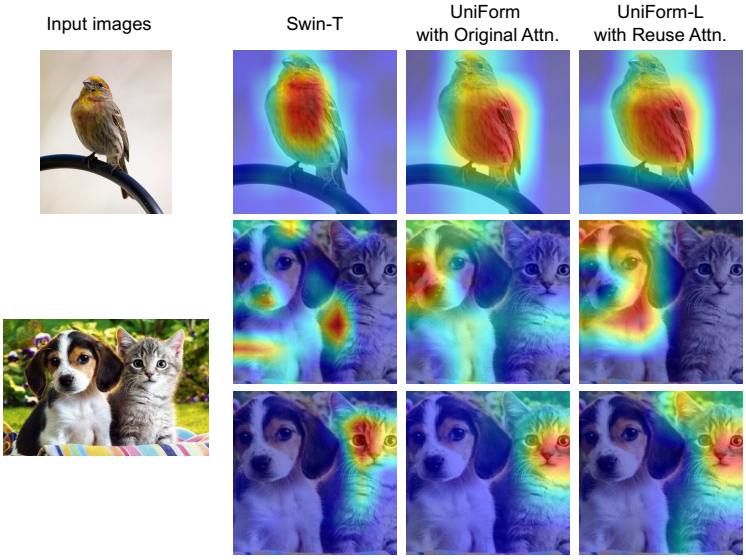

Figure 3: Visualization of feature maps from Swin-T, UniForm without and with Reuse Attention using Grad-CAM. It is obvious that our method with reuse attention can more precisely locate the objects of interest.

3.2 Impact of Reuse Attention on Interpretability and Inference Efficiency

We compare the proposed UniForm model with Reuse Attention against Swin-T and UniForm without Reuse Attention (i.e., with standard attention). As shown in Fig. 3, the CAM visualizations demonstrate that UniForm with Reuse Attention preserves strong interpretability, effectively highlighting relevant regions, similar to Swin-T and the UniForm variant without Reuse Attention. Despite the architectural simplicity of UniForm with Reuse Attention, it maintains comparable interpretability while significantly improving inference time, making it more efficient for real-time applications. This showcases the advantage of Reuse Attention, balancing between interpretability and computational efficiency.

4 Conclusions

In this paper, we introduce UniForm, an efficient Transformer architecture that enhances computational efficiency and scalability. At its core, Reuse Attention reduces redundant computations and

memory access, significantly improving inference speed while maintaining accuracy. UniForm addresses inefficiencies in conventional attention mechanisms, making Transformers more suitable for real-time deployment on resource-constrained devices. Our approach provides a scalable and practical solution for efficient AI deployment by reducing memory overhead and optimizing computation.

In future work, we plan to extend UniForm to language-centric and vision-language tasks (e.g., VQA, image captioning, etc.) to further assess the cross-modal generality of Reuse Attention. While we employ depthwise convolutions for efficient multi-scale value processing, they can incur suboptimal memory access patterns on certain hardware. To better align with the reuse-oriented philosophy of our attention mechanism, we also explore more reuse-friendly alternatives—such as point-wise group convolutions (Zhang et al., 2018b) or tensorized mixers (Novikov et al., 2015)—that offer improved memory efficiency without sacrificing performance.

5 Acknowledgments

This work was supported by the Technology Innovation Program (RS-2024-00468747, Development of AI and Lightweight Technology for Embedding Multisensory Intelligence Modules) funded By the Ministry of Trade Industry & Energy (MOTIE, Korea).

References

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Tianlong Chen, Zhenyu Zhang, Yu Cheng, Ahmed Awadallah, and Zhangyang Wang. The principle of diversity: Training stronger vision transformers calls for reducing all levels of redundancy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12010–12020, 2022a.

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-Former: Bridging mobilenet and transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5260–5269, 2022b.

MMPreTrain Contributors. OpenMMLab's pre-training toolbox and benchmark, 2023.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. RepVGG: Making vgg-style convnets great again. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13733–13742, 2021.

Peiyan Dong, Zhenglun Kong, Xin Meng, Peng Zhang, Hao Tang, Yanzhi Wang, and Chih-Hsien Chou. SpeedDETR: Speed-aware transformers for end-to-end object detection. In *International Conference on Machine Learning*, pp. 8227–8243, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. LeViT: a vision transformer in convnet's clothing for faster inference. In *IEEE/CVF International Conference on Computer Vision*, pp. 12239–12249, 2021.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision GNN: an image is worth graph of nodes. In *Advances in Neural Information Processing Systems 35*, 2022.

Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for MobileNetV3. In *IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

Ajay Jaiswal, Shiwei Liu, Tianlong Chen, and Zhangyang Wang. The emergence of essential sparsity in large pre-trained models: The weights that matter. In *Advances in Neural Information Processing Systems 36*, 2023.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, Minghai Qin, and Yanzhi Wang. SPViT: Enabling faster vision transformers via latency-aware soft token pruning. In *European Conference on Computer Vision*, pp. 620–640, 2022.

Hai Lan, Xihao Wang, Hao Shen, Peidong Liang, and Xian Wei. Couplformer: Rethinking vision transformer with coupling attention. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6464–6473, 2023.

Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. In *Advances in Neural Information Processing Systems 35*, 2022.

Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *IEEE/CVF International Conference on Computer Vision*, pp. 16843–16854, 2023.

Yinan Liang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Mcuformer: Deploying vision tranformers on microcontrollers with limited memory. In *Advances in Neural Information Processing Systems 36*, 2023.

Ji Lin, Wei-Ming Chen, Yujun Lin, John Cohn, Chuang Gan, and Song Han. Mcunet: Tiny deep learning on iot devices. In *Advances in Neural Information Processing Systems 33*, 2020.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. EfficientViT: Memory efficient vision transformer with cascaded group attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14420–14430, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pp. 9992–10002, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: practical guidelines for efficient CNN architecture design. In *Proceedings of the European Conference on Computer Vision*, pp. 122–138, 2018.

Muhammad Maaz, Abdelrahman M. Shaker, Hisham Cholakkal, Salman H. Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. EdgeNeXt: Efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *Proceedings of the European Conference on Computer Vision*, pp. 3–20, 2022.

Sachin Mehta and Mohammad Rastegari. MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022.

Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. DeLighT: Deep and light-weight transformer. In *International Conference on Learning Representations*, 2021.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32*, pp. 14014–14024, 2019.

Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry P. Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems 28*, pp. 442–450, 2015.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, 2019.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. SparQ attention: Bandwidth-efficient LLM inference. In *International Conference on Machine Learning*, 2024.

Kyuhong Shim, Jungwook Choi, and Wonyong Sung. Exploring attention map reuse for efficient transformer neural networks. *arXiv preprint arXiv:2301.12444*, 2023.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

Mingxing Tan and Quoc V. Le. MixConv: Mixed depthwise convolutional kernels. In *British Machine Vision Conference*, pp. 74, 2019.

Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems 34*, pp. 24261–24272, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357, 2021.

Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. MobileOne: An improved one millisecond mobile backbone. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7907–7917, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, 2017.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: co-designing and scaling convnets with masked autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16133–16142, 2023.

Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global vision transformer pruning with hessian-aware saliency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18547–18557, 2023a.

Yuedong Yang, Hung-Yueh Chiang, Guihong Li, Diana Marculescu, and Radu Marculescu. Efficient low-rank backpropagation for vision transformer adaptation. In *Advances in Neural Information Processing Systems 36*, 2023b.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision*, pp. 6022–6031, 2019.

Seokju Yun and Youngmin Ro. Shvit: Single-head vision transformer with memory efficient macro design. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5756–5767, 2024.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. MixUp: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018a.

Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. MiniViT: Compressing vision transformers with weight multiplexing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12135–12144, 2022.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, 2018b.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI Conference on Artificial Intelligence*, pp. 13001–13008, 2020.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

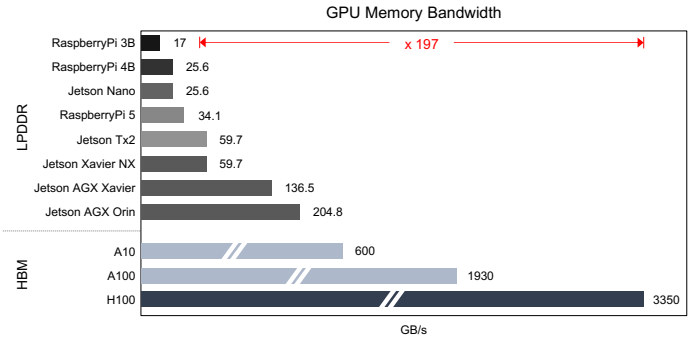

Figure A.1: Comparison of Memory Bandwidth between High-Performance GPUs and Edge Devices.

A Backgrounds

Transformer-based models, including Vision Transformers (ViTs) and Large Language Models (LLMs), have demonstrated remarkable success across various domains due to their ability to capture long-range dependencies and global relationships in input data (Vaswani et al., 2017). At the core of these models is Multi-Head Attention (MHA), which enables effective contextual learning but poses significant challenges, especially in resource-constrained environments such as edge devices. This section explores the fundamental limitations of MHA, particularly its memory and computational overhead, and the resulting performance gap between high-performance GPUs and edge devices.

A.1 Memory Bottlenecks

MHA is inherently memory-intensive due to large matrix multiplications and frequent intermediate storage, leading to significant memory traffic (Kong et al., 2022). As the length of the input sequence increases, the memory traffic worsens, severely impacting the inference speed, particularly in real-time applications (Li et al., 2023).

High-performance GPUs, equipped with High Bandwidth Memory (HBM), efficiently handle these demands. For instance, the NVIDIA H100 provides a memory bandwidth of 3.35 TB/s, enabling smooth execution of large-scale models. In contrast, edge devices like the Raspberry Pi 3B have vastly lower bandwidth (17 GB/s), nearly 197 times smaller, as illustrated in Fig.A.1. This discrepancy forces edge devices to perform frequent memory I/O operations, leading to increased cache misses and latency (Lan et al., 2023). Techniques such as FlashAttention aim to alleviate these inefficiencies by optimizing memory transfers, yet they do not fully address the fundamental memory constraints of edge hardware (Dong et al., 2023; Dao et al., 2022).

A.2 Computation and Parameter Optimization

Beyond memory constraints, MHA suffers from computational inefficiencies. Each attention head processes its attention matrix independently, introducing redundant computations, particularly as the number of heads increases (Liu et al., 2023). Studies indicate that many attention maps exhibit substantial similarity, indicating many computations are unnecessary (Michel et al., 2019). Specifically, Michel et al. found that over 60% of attention heads in BERT can be pruned with minimal accuracy loss, highlighting significant redundancy. This redundancy not only wastes computational resources but also exacerbates memory usage, making MHA less viable for deployment on resource-constrained edge devices.

Furthermore, parameter redundancy remains a major concern. ViT pruning studies show that Query and Key components have high redundancy across layers, whereas the Value component retains crucial information (Yang et al., 2023a). Selectively pruning these elements significantly reduces parameter overhead without sacrificing accuracy. Table A.1 compares memory access costs between MHA and Reuse Attention, demonstrating a notable reduction in memory I/O requirements. It shows

Table A.1: MAC comparison between Multi-Head and Reuse Attention.

	Multi-Head Attention		Reuse Attention	
	Memory Load	Memory Store	Memory Load	Memory Store
$\boldsymbol{S = QK^T}$	$2 \times (N \times d) \times h$	$N^2 \times h$	$2 \times (N \times D)$	N^2
$\boldsymbol{P = Softmax(S)}$	$N^2 \times h$	$N^2 \times h$	N^2	N^2
$\boldsymbol{O = PV}$	$(N^2 + (N \times d)) \times h$	$(N \times d) \times h$	$N^2 + (N \times d \times h)$	$(N \times d) \times h$

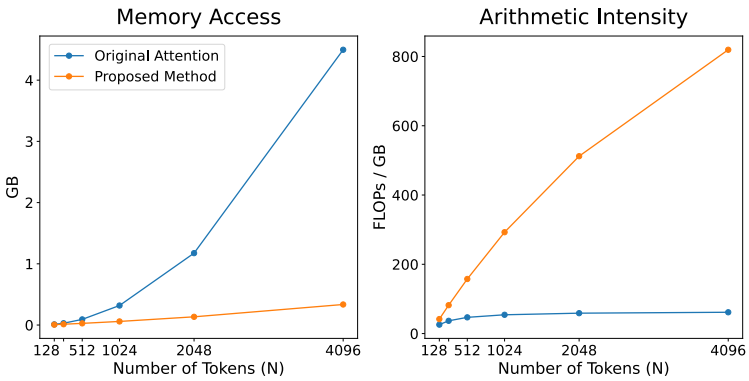

Figure A.2: Impact of the number of tokens on Memory Access and Arithmetic Intensity in Attention.

the substantial memory I/O required by MHA at every stage, as the cost scales with the number of heads, highlighting the need for more efficient attention mechanisms.

In contrast to recent single-head designs like SHViT (Yun & Ro, 2024), which reduce complexity by removing multi-head attention entirely, our method retains multi-head structure and expressiveness via efficient attention reuse and multi-scale value processing.

A.3 Performance Gap between High-performance GPUs vs. Edge Devices

The main difference in hardware capabilities between high-performance GPUs and edge devices results in a substantial performance gap for Transformer-based models. While GPUs leverage HBM for low-latency processing, edge devices like Jetson Nano (25.6 GB/s bandwidth) struggle with memory-constrained computations. As token counts increase, memory demands grow exponentially, overwhelming edge resources and causing frequent cache misses (Lan et al., 2023).

Figure A.2 illustrates how memory access and arithmetic intensity scale with token count in conventional and optimized attention mechanisms. As sequence length surpasses 1024 tokens, memory access becomes the primary bottleneck, significantly slowing down edge-device inference. In contrast, high-performance GPUs maintain efficient execution due to their ample memory bandwidth.

Table A.2: MACs between multi-head and reuse attention across various model categories, including LLM (Llama 2 7B, GPT-3, ALBERT xxlarge, BERT Large, Transformer XL, T5 Large), VLM (CLIP Text Encoder), ViT (ViT Base), and Text-to-Image Generation (Stable Diffusion). Reuse Attention demonstrates a significant reduction in MAC compared to multi-head attention.

Model	Multi-Head Attention	Reuse Attention	Reduction (%)
Llama 2 7B	141.73 GB	8.59 GB	93.94
GPT-3	328.56 GB	22.55 GB	93.14
ALBERT xxlarge	1.81 GB	226.49 MB	87.50
BERT Large	905.97 MB	150.99 MB	83.33
Transformer XL	679.48 MB	113.25 MB	83.33
T5 Large	905.97 MB	150.99 MB	83.33
CLIP Text Encoder	8.34 MB	4.35 MB	47.78
ViT Base	59.23 MB	18.25 MB	69.19
Stable Diffusion	16.68 MB	8.33 MB	50.06

To further emphasize the substantial memory access costs associated with traditional MHA across various models, Table A.2 compares the MAC between MHA and optimized attention mechanisms

Table A.3: Comparison of attention mechanisms in terms of key performance metrics: model performance, GPU efficiency, edge device efficiency, and memory scalability. Each category is rated on a scale of one to three stars (★), with more stars indicating superior performance in that category.

	Model Performance	GPU Efficiency	Edge Efficiency	Memory Scalability
Attention	★★	★	★	★
Linear Attention	★	★★★	★★	★★
Cascaded Attention	★★★	★★★	★	★
Flash Attention	★★	★★★	★	★
Ours	★★★	★★★	★★★	★★★

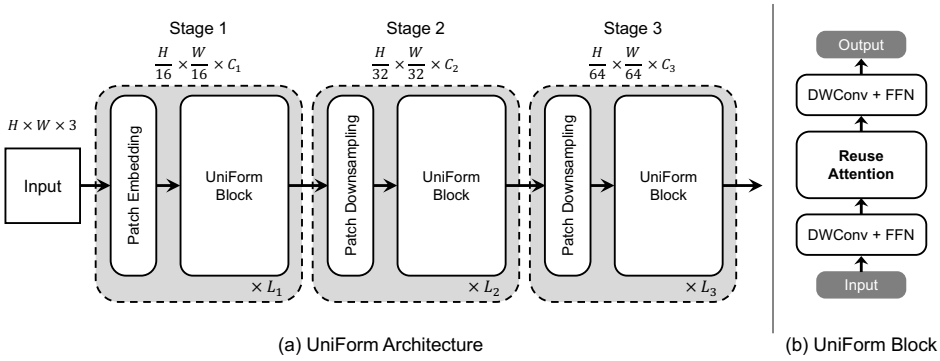

Figure A.3: (a) The architecture of UniForm; (b) UniForm Block including Reuse Attention.

across different model categories, including LLMs (e.g., Llama 2 7B, GPT-3), Vision-Language Models (e.g., CLIP Text Encoder), Vision Transformers (e.g., ViT Base), and Text-to-Image Generation models (e.g., Stable Diffusion). The results demonstrate that MHA incurs significantly higher MAC, exacerbating the performance gap between high-performance GPUs and edge devices.

In contrast, our proposed method effectively mitigates these inefficiencies by reusing the attention matrix across heads and employing multi-scale value processing to reduce memory and computational demands, making it a viable solution for edge deployment. Table A.2 highlights the significant reduction in memory access costs across various model categories (i.e. model performance, GPU efficiency, edge efficiency, and memory scalability).

B Overall Architecture

Figure A.3 provides an overview of UniForm, our hierarchical backbone incorporating Reuse Attention. Similar to other multi-stage architectures (e.g., Swin (Liu et al., 2021), EfficientViT (Liu et al., 2023), and MobileNetV3 (Howard et al., 2019)), UniForm adopts a progressive design with three stages that successively increase feature dimensionality, depth, and the number of attention heads. Each stage refines features at increasing levels of abstraction while controlling computational cost. Key components of the UniForm include:

- **Overlapping Patch Embedding**: We begin by converting 16×16 input patches into token embeddings of dimension C_1. Overlapping windows capture fine-grained local details while minimizing compute overhead.
- **UniForm Blocks**: Each stage comprises multiple UniForm Blocks that embed the proposed Reuse Attention. After each block, the spatial resolution is reduced by a factor of 4 through downsampling. This hierarchical progression maintains spatial coherence while reducing computational load, following the logic of Swin Transformers and other hierarchical ViTs (Liu et al., 2021; 2023).
- **Depthwise Convolutions and FFN Layers**: Placed before and after the Reuse Attention modules, DWConv and Feed-Forward Network (FFN) layers provide a balanced treatment of local and global information. This arrangement enhances model capacity without the

overhead of full self-attention at each step, echoing prior efficiency-driven designs (Li et al., 2023; Chen et al., 2022b).

- **Scalability**: As detailed in Table B.1, UniForm supports Tiny, Small, Medium, and Large configurations, each tailored for different computational budgets and task complexities. By adjusting the channels, depths, and heads, UniForm scales seamlessly to meet the requirements of diverse edge and server-level deployments.

Table B.1: Architecture detail of UniForm model variants.

Architecture	**Channel** $[C_1, C_2, C_3]$	**Depth** $[L_1, L_2, L_3]$	**Head** $[H_1, H_2, H_3]$
Tiny	$[64, 128, 192]$	$[1, 2, 2]$	$[4, 4, 4]$
Small	$[128, 144, 192]$	$[1, 2, 2]$	$[4, 4, 4]$
Medium	$[128, 240, 320]$	$[1, 2, 3]$	$[4, 3, 4]$
Large	$[192, 288, 384]$	$[1, 2, 3]$	$[3, 3, 4]$

UniForm leverages intermediate feature reuse within and across stages to further reduce computational overhead, all while preserving accuracy. This flexible framework accommodates varying patch sizes and resolutions, ensuring broad applicability in tasks ranging from object recognition to dense prediction.

C Implementation Details

To rigorously evaluate the effectiveness of UniForm, we conduct extensive experiments spanning both large-scale benchmarks and real-world deployment scenarios. Our evaluation aims to validate the efficiency, scalability, and generalizability of the proposed architecture, emphasizing both accuracy and inference efficiency across diverse hardware configurations. The models are implemented using PyTorch 2.3.0 (Paszke et al., 2019) and MMPreTrain 1.2.0 (Contributors, 2023) and trained from scratch on ImageNet-1K (Deng et al., 2009) for 300 epochs using two NVIDIA A100 GPUs. We adopt the AdamW optimizer (Loshchilov & Hutter, 2019) with a cosine annealing learning rate scheduler (Loshchilov & Hutter, 2017), setting the total batch size to 512. Input images are resized and randomly cropped to 224×224. The initial learning rate is set to 0.001 with a weight decay of 0.025. We incorporate advanced augmentation and regularization techniques such as Mixup (Zhang et al., 2018a), Cutmix (Yun et al., 2019), and random erasing (Zhong et al., 2020) to enhance generalization.

Additionally, we assess throughput across different hardware platforms. For GPU performance, throughput is measured on an NVIDIA A100 with a batch size of 2048 to ensure a fair comparison across models. For CPU-based inference, we evaluate runtime on an Intel Xeon Gold 6426Y @ 2.50 GHz processor using a batch size of 16 and single-thread execution following the methodology in (Graham et al., 2021).

In contrast to prior works, we conduct extensive inference evaluations on a range of edge devices. Specifically, we test on multiple versions of the *Jetson* (*Nano, Xavier, Tx2, Nx, and AGX Orin*) and *Raspberry Pi* (*2B, 3B, 3B Plus, 4B, and 5*). All models are evaluated with a batch size of 16 and executed in single-thread mode to ensure consistency. For all edge devices, we used ONNX Runtime with the default CPU Execution Provider. All inferences were performed using FP32 precision without quantization or mixed precision optimization. This evaluation underscores the practicality of UniForm for edge computing environments, where computational and memory constraints are significantly more stringent than those encountered in server-grade hardware.

Furthermore, we assess the transferability of UniForm to downstream tasks. For image classification, we fine-tune the models for 300 epochs following the methodology of (Zhang et al., 2022) with similar hyperparameter settings. For instance segmentation on the COCO dataset (Lin et al., 2014), we use Mask R-CNN and train for 12 epochs (1× schedule) with the same settings as (Liu et al., 2021) using the MMDetection framework (Chen et al., 2019).

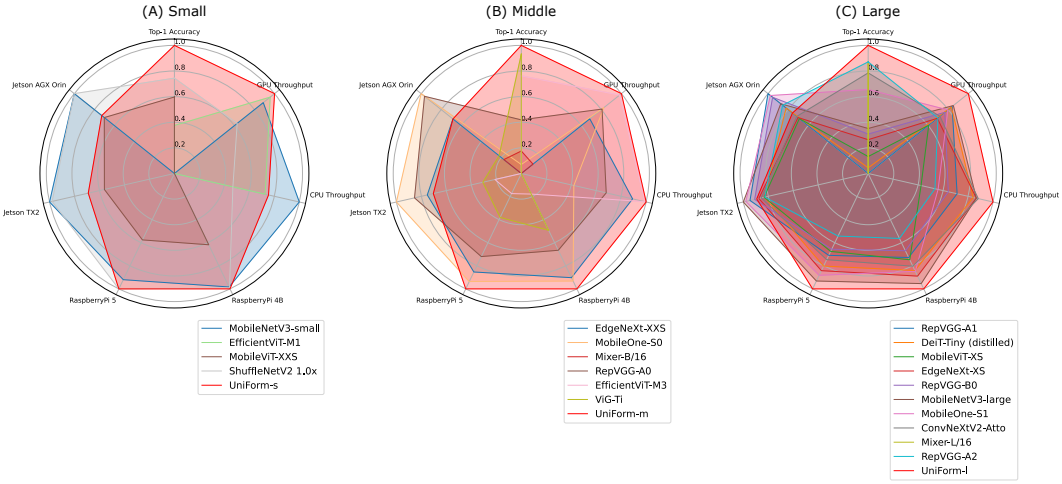

Figure C.1: Chart comparison of UniForm models with comparable state-of-the-art models across different sizes on a variety of metrics (Top-1 Accuracy and GPU/CPU/Edge-device throughput).

Table C.1: Inference speed and accuracy comparison between UniForm and state-of-the-art CNN and ViT models across various edge devices.

Model	Top-1 Accuracy	Jetson-Nano	RaspberryPi4B	RaspberryPi3B Plus	RaspberryPi3B	RaspberryPi2B	Jetson-Xavier	Jetson-Tx2	Jetson-Nx	RaspberryPi5	Jetson AGX Orin
EfficientViT-M4	74.3	108.6	634.9	1264.7	1352.8	5263.7	17.2	45.6	25.9	201.7	8.3
RepVGG-A1	74.4	22.4	212.4	425.6	478.4	1991.4	3.6	9.0	6.0	73.2	1.1
DeiT-Tiny (distilled)	74.5	39.0	134.0	237.5	260.1	1109.6	7.5	16.6	10.2	49.9	2.0
MobileViT-XS	74.7	36.4	196.1	353.7	389.0	1290.3	7.0	16.3	10.3	84.3	2.9
EdgeNeXt-XS	75.0	115.0	109.9	186.2	209.9	723.9	6.0	13.1	8.3	42.3	2.8
RepVGG-B0	75.1	264.4	269.7	546.3	608.0	2529.4	4.4	11.0	7.3	91.1	1.3
MobileNetV3-large	75.2	14.9	84.2	163.5	181.7	566.5	3.1	6.7	4.2	29.1	1.7
MobileOne-S1	75.9	119.0	130.1	228.7	265.3	973.6	3.2	7.5	4.9	35.8	1.2
ConvNeXtV2-Atto	76.2	146.4	156.1	294.5	318.2	993.1	6.6	14.5	8.9	62.8	2.7
Mixer-L/16	76.4	N/A	4113.5	N/A	N/A	N/A	105.6	N/A	N/A	1323.3	14.2
RepVGG-A2	76.4	43.6	416.5	885.7	960.8	4170.4	6.9	16.8	10.6	145.3	1.8
UniForm-l	**76.7**	**19.6**	**69.7**	**134.3**	**148.7**	**567.5**	**5.1**	**12.0**	**7.8**	**21.8**	**2.4**

D Downstream Tasks

We validated the effectiveness of UniForm models on several downstream tasks, focusing on image classification and instance segmentation to showcase the model's adaptability and competitive edge over state-of-the-art architectures.

D.1 Image Classification Downstream Tasks

To further assess its generalization capability, UniForm is evaluated on various image classification benchmarks, including CIFAR-10, CIFAR-100, Flowers-102, and Oxford-IIIT Pet. Table D.1 illustrates that UniForm-l achieves **98.2% accuracy on CIFAR-10** and **97.5% on Flowers-102**, demonstrating its adaptability across diverse dataset distributions. These findings affirm that the efficiency improvements of UniForm do not compromise its generalization ability, making it a robust solution for various vision tasks.

D.2 Instance Segmentation

Table D.2 shows the performance of UniForm on instance segmentation tasks with COCO dataset using Mask R-CNN. UniForm-l achieves $\mathbf{AP}^b$ **of 33.2** and $\mathbf{AP}^m$ **of 31.5**, outperforming EfficientViT-m4 and MobileNetV3, indicating its effectiveness in instance segmentation.

Table D.1: Results of UniForm and other state-of-the-art models on various downstream image classification datasets (CIFAR-10, CIFAR-100, Flowers-102, and Oxford-IIIT Pet).

Model	Throughput ↑ GPU	CPU	Inference Time on Edge ↓	ImageNet	CIFAR-10	CIFAR-100	Flowers-102	Oxford-IIIT Pet
DeiT-Tiny	13785	63	49.9	74.5	98.1	86.3	96.9	91.5
MobileViT-XS	6098	13	84.3	74.7	97.5	84.1	96.1	91.9
RepVGG-B0	10868	30	91.1	75.1	97.7	85.4	96.0	91.2
ConvNeXtV2-Atto	9120	73	62.8	76.2	97.2	83.4	87.7	71.4
MobileOne-S1	12150	22	35.8	75.9	97.7	85.8	97.4	92.2
RepVGG-A2	8483	20	145.3	76.4	97.9	85.5	95.9	91.9
EfficientViT-M5	21572	101	326.5	77.1	98.0	86.4	97.1	92.0
DeiT-small	5768	17	147.4	**80.6**	**98.5**	**87.2**	95.7	91.8
UniForm-l	**25356**	**113**	**21.8**	76.7	98.2	86.5	**97.5**	**92.2**

Table D.2: Performance comparison of instance segmentation on COCO2017

Model	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
MobileNetV2	29.6	48.3	31.5	27.2	45.2	28.6
MobileNetV3	29.2	48.6	30.3	27.1	45.5	28.2
FairNAS-C	31.8	51.2	33.8	29.4	48.3	31.0
EfficientNet-B0	31.9	51.0	34.5	29.4	47.9	31.2
MNASNet-A1	32.1	51.9	34.2	29.7	49.0	31.4
EfficientViT-m4	32.8	54.4	34.5	31.0	51.2	32.2
UniForm-l	**33.2**	**54.9**	**35.3**	**31.5**	**51.8**	**32.8**