Invited Paper: BitMedViT: Ternary-Quantized Vision Transformer for Medical AI Assistants on the Edge

Mikolaj Walczak, Uttej Kallakuri, Edward Humes, Xiaomin Lin, Tinoosh Mohsenin

Department of Electrical and Computer Engineering

Johns Hopkins University

{mwalcza1, ukallak1, ehumes2, xlin52, tinoosh}@jh.edu

Abstract—Vision Transformers (ViTs) have demonstrated strong capabilities in interpreting complex medical imaging data. However, their significant computational and memory demands pose challenges for deployment in real-time, resource-constrained mobile and wearable devices used in clinical environments. We introduce, BiTMedViT, a new class of Edge ViTs serving as medical AI assistants that perform structured analysis of medical images directly on the edge. BiTMedViT utilizes ternary-quantized linear layers tailored for medical imaging and combines a training procedure with multi-query attention, preserving stability under ternary weights with low-precision activations. Furthermore, BiTMedViT employs task-aware distillation from a high-capacity teacher to recover accuracy lost due to extreme quantization. Lastly, we also present a pipeline that maps the ternarized ViTs to a custom CUDA kernel for efficient memory bandwidth utilization and latency reduction on the Jetson Orin Nano. Finally, BiTMedViT achieves 86% diagnostic accuracy (89% SOTA) on MedMNIST across 12 datasets, while reducing model size by 43$\times$, memory traffic by 39$\times$, and enabling 16.8 ms inference at an energy efficiency up to 41$\times$ that of SOTA models at 183.62 GOPs/J on the Orin Nano. Our results demonstrate a practical and scientifically grounded route for extreme-precision medical imaging ViTs deployable on the edge, narrowing the gap between algorithmic advances and deployable clinical tools. Github is available at **https://github.com/M-iki/BitMedViT**

Index Terms—Vision Transformers, Medical Imaging, Ternary Quantization, Edge Computing, Real Time.

I. INTRODUCTION

Machine learning for medical imaging [6, 14, 31, 42] and disease detection [8] are rapidly advancing fields with the potential to transform healthcare by enabling real-time, automated analysis of imaging modalities such as X-rays, MRIs and CT scans. This automation supports accurate disease identification, improves diagnostic precision and accelerates clinical decision-making. Historically, Convolutional Neural Networks (CNNs), especially those with residual architectures like ResNet[9], have demonstrated strong performance. Recent breakthroughs [19, 20, 41] have advanced medical image classification by surpassing traditional CNN accuracy through sophisticated architectures such as Vision Transformers (ViTs) [7] and Vision Mamba [2, 44].

Despite these advances, deploying ViTs in clinical environments remains challenging due to their high computational and memory demands, which restrict adoption in clinical settings with limited hardware resources or unreliable connectivity.
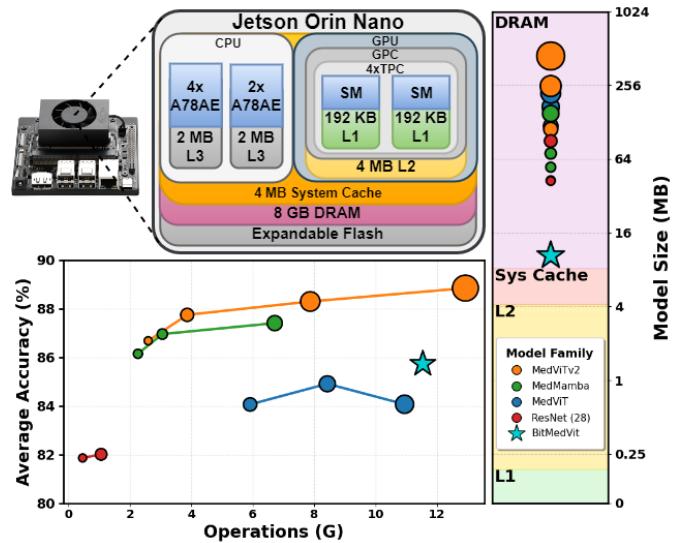

Fig. 1. Jetson Orin Nano hierarchy and SOTA medical image classification model comparisons for average accuracy on the MedMNIST dataset vs operations and model size compared to the memory available within the GPU consisting of 4 TPCs each with two SMs containing their own L1 cache. Marker size corresponds to the parameter count within each model.

Furthermore, transmitting sensitive medical data over wireless networks introduces privacy and security risks [4] - a major concern given HIPAA and other stringent medical data regulations [21]. To address these issues, we investigate deploying these models on the resource-constrained NVIDIA Jetson Orin Nano edge device. Equipped with Arm Cortex-A78AE CPU and an NVIDIA Ampere GPU, the Orin Nano is tailored for low-power, resource-constrained environments. Deploying models optimized for its core computing architecture ensures efficient resource utilization, reduced power consumption, and optimal performance within strict energy budgets. As shown in Figure 1, the state-of-the-art medical classification models, even under ideal settings, exceed the low memory hierarchy (L1, L2, L3) limits, causing high latency, low throughput, and increased energy use due to the reliance on external accesses to high-latency DRAM. To overcome these deployment constraints, recent research has emphasized model compression techniques such as pruning [10, 11, 12, 13, 18], quantization [22], and Knowledge Distillation (KD) [1, 29, 30] that reduce

model size and inference costs while maintaining accuracy. KD has proven especially effective, and when combined with quantization, it yields lightweight models retaining high performance at a fraction of the original footprint, which is ideal for edge medical AI applications. In ViTs, further reductions in parameters and memory can be achieved by replacing standard Multi-Head Self-Attention with Multi-Query Attention, significantly cutting computational overhead with minimal impact on accuracy [5, 16, 34].

In this work, we introduce BiTMedViT, which integrates KD with extreme quantization to enhance model compression. We focus on 2-bit ternary quantization of feed-forward layers using the BitNet-1.58B framework [17] and are one of the first works to explore MQA in ViTs to compress the model footprint for medical image classification. We evaluate our method for accuracy and efficiency on edge hardware, benchmarking against MedViTV2 [20] deployed on the NVIDIA Jetson Orin Nano with custom CUDA kernels optimized for performance.

Our key contributions are summarized as follows:

- We evaluate the efficacy of MQA in ViTs for medical image classification, demonstrating more efficient parameter use while preserving accuracy.
- We apply feature and logit distillation from a high-performing state-of-the-art teacher model and propose a distilled student BiTMedViT for medical applications that enables robust classification .
- We integrate BitNet linear layers within BiTMedViT, enabling the training of scalable, memory- and parameter-efficient edge-based ViTs.
- We develop a optimized CUDA kernel compatible with TensorRT and compatible with Nvidia Ampere GPU architectures, found within the Jetson Orin Nano, enabling real-time inference and efficient memory utilization for edge-based medical AI applications.

II. RELATED WORKS

A. Knowledge Distillation for Medical ViTs

Knowledge distillation (KD) is a proven strategy for compressing large models, particularly effective in data-scarce medical imaging scenarios. Feature-based KD enhances student learning by aligning internal representations with those of a powerful teacher model. Medical-specific distillation methods such as [27, 33] address limited and imbalanced data, while quantization-aware techniques [15, 43] further reduce inference cost. ***BiTMedViT*** builds on these by combining feature distillation with low-bit quantization to produce accurate, lightweight ViTs for edge devices.

B. Multi-Query Attention in Vision Models

Multi Query Attention (MQA) reduces attention complexity by sharing key-value pairs across queries, offering significant efficiency gains over Multi-Head Self Attention (MHSA). Although adopted in Large Language Models (LLMs) [5, 34], MQA has seen little application in ViTs. Ainslie et al. [3] show that MHSA architectures can be converted to MQA with minimal changes. BiTMedViT is among the first to evaluate MQA in ViTs for medical imaging.

C. Ternary ViTs and Edge Deployment

Ternary quantization reduces weights and activations to three discrete values, balancing efficiency and accuracy. Prior works like Tervit [36] and BitNet-ViT [40] demonstrate that ViTs can be effectively quantized with minimal accuracy loss. However, these methods often overlook deployment feasibility on constrained hardware. BiTMedViT builds on the BitNet-1.58B framework [17], integrating BitLinear ternary layers and demonstrating practical, low-latency deployment on edge devices like the Jetson Orin Nano. Furthermore, deploying ViTs on edge devices requires tight alignment between model design and hardware constraints. While frameworks like BitNet-Efficient [35] and FPGA-based ternary transformers [39] demonstrate efficient execution, they target general-purpose or LLM scenarios. Prior works have also explored KD, quantization, and ternary ViTs for efficient model compression, but often lacks real-world deployment on medical setting. In addition, MQA is common in LLMs but is rarely applied to ViTs. Our proposed work bridges these gaps by combining feature distillation, extreme ternary quantization, and MQA into a compact ViT pipeline, optimized and deployed with custom CUDA kernels for real-time inference on edge devices like the Jetson Orin Nano.

III. PROPOSED APPROACH

A. BiTMedViT Architecture

The BiTMedViT architecture is based on the traditional ViT [7], parameterized by the number of attention heads H, transformer layers L, and patch embedding dimension E_d. The post-attention Feed-Forward Networks (FFNs) employ an expansion factor of 4, yielding an FFN dimension of $E_{\text{ff}} = 4 \times E_d$. In this work, we fix the configuration to $L = 3$, $H = 8$, and $E_d = 512$, providing a balanced trade-off between model capacity for medical image analysis and a reduced overall parameter count. Prior to deployment on the Orin Nano GPU platform, it is essential to ensure the model is sufficiently compact to meet stringent memory constraints before hardware-specific optimizations are applied. As illustrated in Fig. 2, a parameter breakdown of the conventional MHSA architecture under this configuration shows that the FFN layers and key–value projections contribute the largest share of parameters. This motivates our focus on compressing these components to reduce overall memory footprint.

1) Attention Layer Compression: Transformer architectures rely heavily on the attention mechanism, which computes context-aware representations through learned input feature linear projections. This mechanism is formally expressed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \quad (1)$$

where the queries (Q), keys (K), and values (V) are projections of the input (X). Moreover, the total embedding dimension in Equation 1 is divided into multiple heads H, each with separate Q, K, and V projection weights. While this increases parallelism and expressivity, the total number of
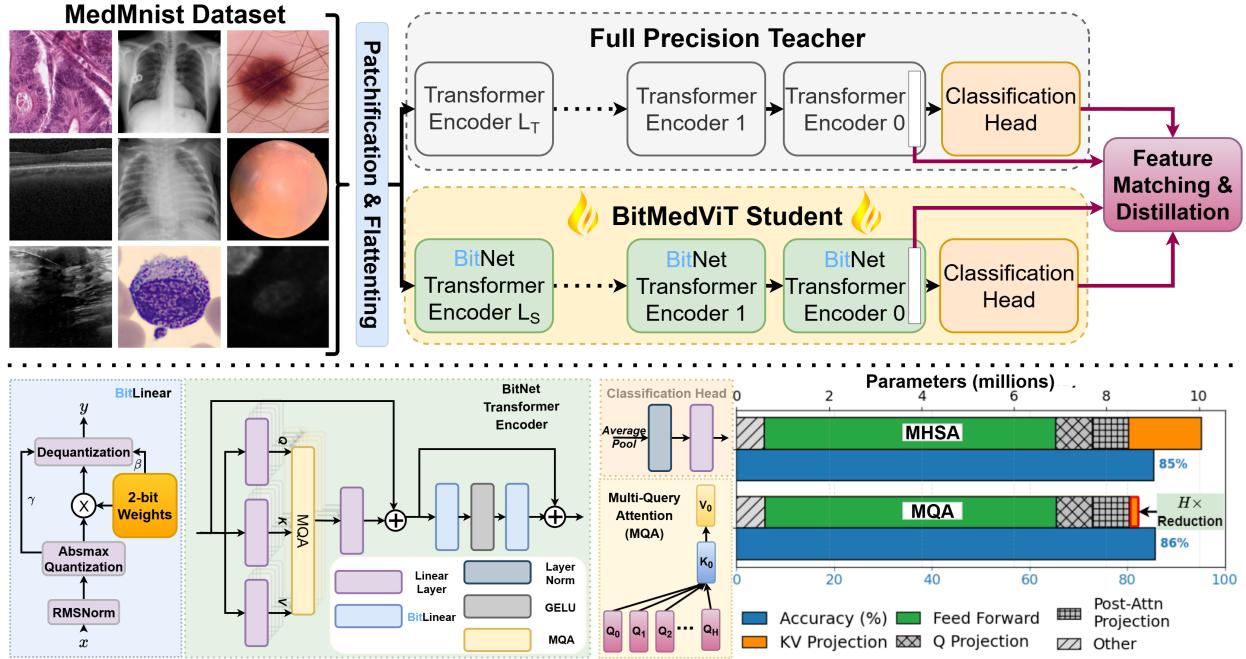
Fig. 2. Model training paradigm and BiTMedViT Architecture. Experiments illustrate the overall parameter breakdown and reduction when comparing Multi-Head Self-Attention (MHSA) and Multi-Query Attention (MQA) mechanisms within BiTMedViT. The blue plots depict the average validation accuracy achieved when training 12 student models across each of the MedMNIST2D [37, 38] datasets rounded to the nearest percentage point after 100 epochs.

parameters remains constant, as the projections are simply partitioned into smaller subspaces. MQA modifies this structure by sharing the key and value projections across all heads. Let $X \in \mathbb{R}^{N \times D}$ be the patch embeddings, with H heads of size d_h so $D = Hd_h$. The attention output for MHSA can then be computed using Equation 1 with the projection weights of Q,K and V in a single head being

$$W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{D \times d_h} \tag{2}$$

where Q_h, K_h and V_h are computed as $Q_h = XW_h^Q$, $K_h = XW_h^K$, $V_h = XW_h^V$. Within each head we further compute $Y_h = \text{Attn}(Q_h, K_h, V_h)$, then $Y = \text{Concat}_h(Y_h)W^O$. The number of parameters in this case is, $3Dd_hH$.

MQA on the other hand, maintains the per–head queries but shares the key and value projections across all the heads. Equation 2 in this case can be rewritten as

$$W_h^Q \in \mathbb{R}^{D \times d_h}, \qquad W_{\text{sh}}^K, W_{\text{sh}}^V \in \mathbb{R}^{D \times d_h}. \tag{3}$$

where, W_{sh}^K and W_{sh}^V are the shared projections. We then compute for Q_h, K_{sh} and V_{sh} as, $Q_h = XW_h^Q$ across all heads, while $K_{\text{sh}} = XW_{\text{sh}}^K$, $V_{\text{sh}} = XW_{\text{sh}}^V \in \mathbb{R}^{N \times d_h}$ is shared. The attention within each head is now $Y_h = \text{Attn}(Q_h, K_{\text{sh}}, V_{\text{sh}})$ followed by $Y = \text{Concat}_h(Y_h)W^O$.

From Equations 2 and 3 we can conclude that by replacing the MHSA layers by MQA layer the number of parameters for the self-attention reduces by,

$$\underbrace{|W^K| + |W^V| = 2Dd_hH}_{\text{MHSA}} \longrightarrow \underbrace{|W_{\text{sh}}^K| + |W_{\text{sh}}^V| = 2Dd_h}_{\text{MQA}} \tag{4}$$

$$\Rightarrow \quad \mathbf{1/H} \text{ reduction for } K, V.$$

We adopt this approach in ViTs and evaluate its effectiveness in a representative experiment shown in the lower right portion of Figure 2 using the training pipeline defined

within Section III-B. Our results align with those of [3], demonstrating that given a pretrained checkpoint or a high-capacity teacher, MQA can approach the accuracy of its MHSA counterpart while reducing overall parameter count.

2) Ternary Quantization: To aggressively compress the FFN layers found within BiTMedViT, we adapt the BitLinear layers from [17] which computes W_2A_8 output activations given 2-bit weights (W_2) and int8 activations (A_8). We modify this computation to operate over the full precision weight matrix W and patch embedding matrix A during training defined over the range $W \in \mathbb{R}^{k \times n}$, $A \in \mathbb{R}^{m \times k}$. The quantized counterparts are then computed using absmean quantization[17] for W_2 and absmax quantization[17] for A_8 formally defined as

$$W_2 = \text{RoundClip}\Big(\frac{W}{\beta + \epsilon}, -1, 1\Big), \quad \beta = \frac{1}{kn}\sum_{i=1,j=1}^{k,n}|W_{ij}| \tag{5}$$

and

$$A_8 = \text{Clip}\Big(\frac{A}{\gamma + \epsilon}, -128, 127\Big), \quad \gamma = \frac{\max(|x|)}{127} \tag{6}$$

Where ϵ represents a small floating point number. We modify $\gamma \in \mathbb{R}^m$ to be a vector of length determined by the the number of patch embeddings m, while $\beta \in \mathbb{R}$ remains as a single scalar representing the mean value over the entire weight matrix W. The scale factors γ, β are maintained within 16-bit precision during inference to ensure precise de-quantization for corresponding layers computed as $O = W_2A_8 \times \gamma \times \beta$.

B. Knowledge Distillation

Since the ternary quantized bitlinear layers require extensive training from scratch [17, 40] and MQA demonstrates effectiveness primarily when adapted from a pretrained model, we utilize MedViTv2 [20] a state-of-the-art high-performing
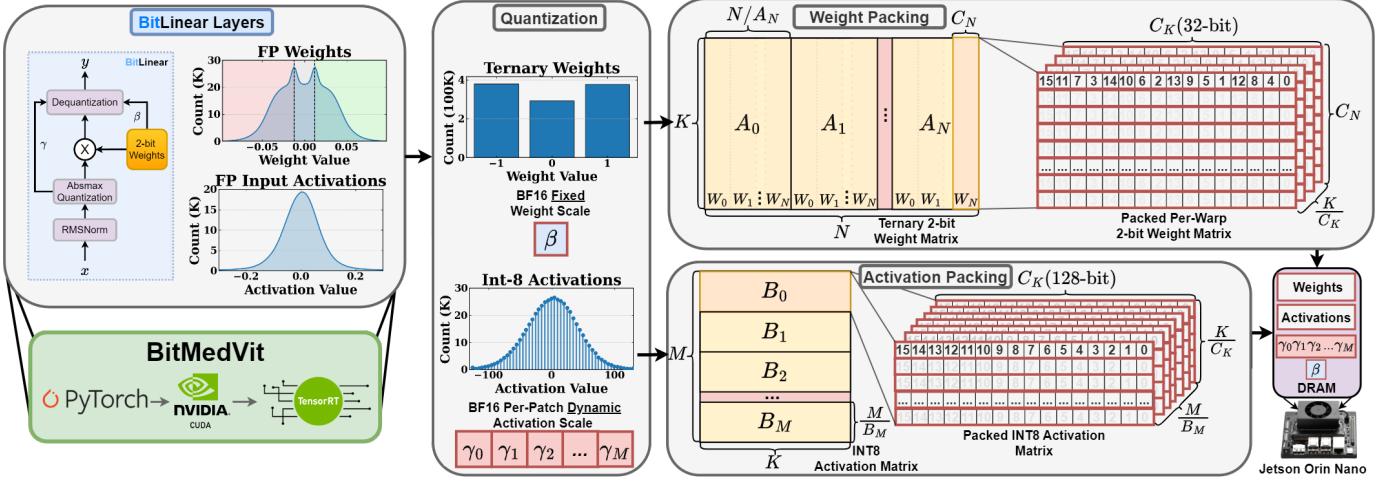
Fig. 3. Bit Packing and custom hardware deployment strategy of BiTMedViT within the GPU of the Jetson Orin Nano. Full precision weights are statically quantized to 2-bit, packed into 32-bit column major words and activations dynamically quantized to int8 rowmajor 128-bit words converting the model from its Pytorch[28] to the custom cuda implementation merged within TensorRT.

Algorithm 1 CUDA Kernel for Blocked Matrix–Matrix Multiplication with 2-bit Weight Unpacking

---

Require: Weights W (2-bit packed), patch input activations P (int8), per-patch scale factors γ, weight scale factor β, block sizes A, B

Ensure: O (dequantized, bfloat16) stored in global memory

1: **Init:**
　　 shared_mem_scale $\leftarrow (\gamma \times \beta)$
　　 $W_{\text{tile}} \leftarrow$ weight start addr (index A)
　　 $P_{\text{tile}} \leftarrow$ activation start addr (index B)
2: **for** $k = 0$ **to** K **step** tile_size **do**
3: 　　**SyncLoad:** DRAM $\rightarrow$ registers (W_{packed})
4: 　　**AsyncLoad:** DRAM $\rightarrow$ shared_mem (P_{tile})
5: 　　int8_weights $\leftarrow$ Unpack$(W_{\text{packed}}) \rightarrow$ shared_mem
6: 　　frag_B $\leftarrow$ **WMMA.Load**(int8_weights)
7: 　　**SyncThreads**
8: 　　**for** $m = 0$ **to** $M/B - 1$ **do**
9: 　　　　frag_P $\leftarrow$ **WMMA.Load**($P_{\text{tile}}[m]$)
10: 　　　　accum[m] $\leftarrow$ **WMMA.MMA**(frag_P, frag_B)
11: 　　**end for**
12: 　　**SyncThreads**
13: **end for**
14: **Finalize:**
　　 DRAM $\leftarrow$ Dequantize(accum, shared_mem_scale)

---

medical image classification ViT, achieving strong accuracy across medical imaging benchmarks. We train BiTMedViT by minimizing a composite loss function

$$\mathcal{L}_{\text{total}} = \lambda_{\text{cls}} \mathcal{L}_{\text{CE}} + \lambda_{\text{logits}} \mathcal{L}_{\text{KD}} + \lambda_{\text{feat}} \mathcal{L}_{\text{feat}} \quad (7)$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss for classification, $\mathcal{L}_{\text{KD}}$ denotes the Kullback–Leibler divergence aligning the student's logits with the teacher's, and $\mathcal{L}_{\text{feat}}$ encourages alignment between intermediate feature representations. To facilitate this process, a trainable projection layer is incorporated during training to align the student's feature dimensions with those of

the teacher, in effect defining BiTMedViT to match the same patch size as MedViTv2.

C. Model Optimization and Hardware Deployment

BiTMedViT utilizes both ternary and full precision layers, necessitating a deployment strategy that optimizes for mixed precision within the GPU of the Orin Nano consisting of a custom CUDA kernel integrated within TensorRT[26]. To design our optimized kernel, The Jetson Orin Nano GPU is organized into a Graphics Processing Cluster (GPC), which contains four Texture Processing Clusters (TPCs), each with two Streaming Multiprocessors (SMs). Each SM includes four Tensor Cores capable of performing Warp Matrix Multiply (WMMA) operations at a minimum int8 precision, of varying dimensions. To maximize memory and compute bandwidth, operations are performed at the per-warp level, where a warp consists of 32 threads, with a maximum read transaction of int4 (128 bits) per thread. Each SM communicates with the 8GB external DRAM through a hierarchical memory system: a 192KB L1 cache per SM, a 4MB L2 cache shared among all SMs, and a 4MB system cache. Since DRAM reads introduce the highest latency, designing highly optimized kernels that perform compact, coalesced data accesses is essential to reduce L2 cache misses and maximize throughput.

1) Weight Packing Strategy and WMMA Compatibility: To optimize inference of BiTMedViT, we redesign the weight packing scheme from the original BitNet [23] kernel to align with operations for maximizing onboard Tensor Core utilization. Weights are arranged into matrices of size $8 \times 32 \times 16$ (M $\times$ N $\times$ K). Within this layout, weights are packed into column major 32×16 fragments, where each 32-bit memory word encodes sixteen (C_K) 2-bit weight values with activations quantized to 8-bit integers grouped into 128-bit (int4) fragments, allowing sixteen activation values to be loaded per memory transaction. as shown in Figure 3. This compact packing reduces memory traffic and allows efficient unpacking via optimized bit masking and shifting during inference. In

addition the 32 outputs of N per input patch M enables coalesced and consecutive 16-bit (BF16) output write-backs during de-quantization.

2) CUDA Kernel Integration: As outlined within Figure 3 Our custom CUDA kernel parallelizes computation using a two-dimensional grid of size $A \times B$, where A corresponds to output channels and B to input patches divided among each of the onboard SMs. Each thread block contains multiple 32-thread warps, with threads accessing distinct output elements acting on the same activation fragment. To minimize runtime overhead, decoded weights are unpacked once per inference and stored in shared memory within each thread block. Activations, which have lower temporal reuse, are asynchronously loaded into shared memory, skipping past the L1 cache maintaining a continuous data streaming pipeline and enabling weight decoding during activation loading. Algorithm 1 outlines this functionality and the overall per-block kernel execution.
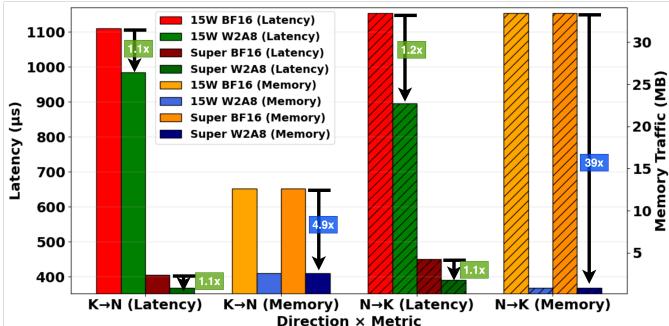

Fig. 4. Latency and memory read traffic comparing the PyTorch[28] BF16 kernel to our optimized W2A8 implementation for two varying workloads. Results on the left show the latency and memory read transactions for the K-N layer and results on the right show the N-K layer within the BiTMedViT FF network. Two power modes, 15W and Super (25W) mode are used to determine performance under varying clock speeds.

Using the Nvidia Nsight Compute [24] and Nsight Systems [25] profiling tools, we extensively evaluate the efficiency of the custom kernel implementation as well as overall inference time for the full model architecture. Profiling the linear layers in BiTMedViT as shown in Figure 4 shows a $4.9\times$ to $39\times$ reduction in weight traffic (bytes moved). This shifts most accesses to on-chip memory and minimizes DRAM transactions, aligning with the power/latency gains.

3) TensorRT Integration for Efficient Deployment: We deploy BiTMedViT as an end-to-end solution by integrating our custom CUDA kernel into NVIDIA TensorRT. Since TensorRT doesn't natively support quantized/compressed BitLinear layers, the plugin registers the layer, declares I/O shapes, and specifies supported precisions enabling building of engines with the new operation. On the Orin Nano, TensorRT had limited support for dynamic INT8 activations and would upcast them to FP16 at the plugin boundary. To accommodate this, we added kernel variants that accept BF16 activations and scales, and due to the limited BF16 support on the Orin Nano, FP16 activations and scales.
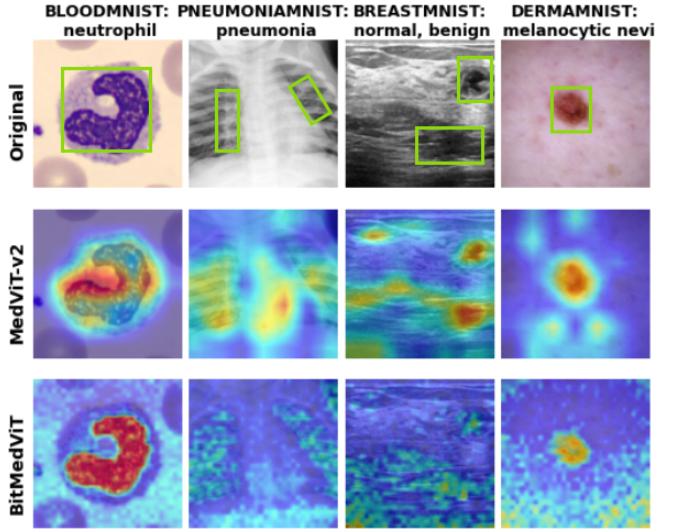

Fig. 5. Visualized GradCam[32] for the MedViT-v2-L teacher and BiTMed-ViT across four representative datasets found within MedMNIST. Green boxes represent the manually placed relevant regions for diagnosis.

IV. RESULTS

A. BiTMedViT Accuracy Benchmarking

MedMNIST2D [37, 38] serves as the dataset used to evaluate the performance of BiTMedViT for medical image classification spanning 12 2D datasets of varying modalities and complexity. Using the pretrained MedViTv2-L teacher, we apply the distillation strategy outlined in Section III-B across each dataset. Figure 5 presents attention maps generated using GradCam [32] for four representative images, which, while exhibiting increased noise, demonstrate stronger focus on diseased regions. These results highlight the ability of BiTMedViT to attend to clinically relevant regions despite aggresive reduction in parameters and weight expressivity.

Furthermore, we compare overall accuracy and ROC AUC against state-of-the-art medical image classification models, as shown in Table I. Across the MedMNIST benchmark, BiTMedViT achieves competitive accuracy relative to leading models, attaining perfect or near-perfect AUC scores on multiple datasets, including PathMNIST, BloodMNIST, and OrganCMNIST. Notably, even after applying aggressive compression strategies, BiTMedViT maintains performance close to the teacher, with only a 3% decrease in average validation accuracy (86% vs. 89%) and a similar reduction on the test set (82% vs. 85%), demonstrating effective model compression while preserving competitive performance.

B. Hardware Deployment

We deploy BiTMedViT within the Orin Nano and compare against (i) full-precision MedViTv2-L[20] baseline and (ii) MedMambaLite[2]. with hardare results summarized in Table II. BiTMedViT achieves 16.88 ms latency per inference versus 366.63 ms for the MedViTv2-L baseline ($\approx$21.7$\times$ faster). This corresponds to a 683.06 GOPs/sec throughput, 19.4$\times$ higher than the baseline. When compared to MedMambaLite[2], our implementation is 57.7$\times$ better in terms of throughput while being $\approx 42\times$ more energy efficient.

TABLE I

PERFORMANCE OF BITMEDVIT ACROSS MEDMNIST2D DATASETS IN COMPARISON WITH STATE-OF-THE-ART CLASSIFICATION MODELS. THE TEACHER MODEL MEDVITV2-L AND BITMEDVIT ARE HIGHLIGHTED. VALIDATION ACCURACY IS REPORTED FOR FAIR COMPARISON, WHILE TEST ACCURACY FOR EACH DATASET IS INDICATED WITH A ∗.

Model	PathMNIST		ChestMNIST		DermaMNIST		OCTMNIST		PneumoniaMNIST		RetinaMNIST	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
ResNet50 (224) [9]	89.2	98.9	94.8	77.3	73.1	91.2	77.6	95.8	88.4	96.2	51.1	71.6
MedMamba-S [41]	95.5	99.7	-	-	75.8	92.4	92.9	99.1	93.6	97.6	54.5	71.8
MedMamba-B [41]	96.4	99.9	-	-	75.7	92.5	92.7	99.6	92.5	97.3	55.3	71.5
MedViTv2-S [20]	96.5	99.8	96.4	80.3	79.2	94.6	94.2	99.4	96.5	99.6	56.2	78.0
MedViTv2-B [20]	97.0	99.9	96.4	81.5	80.8	94.8	94.4	99.6	96.9	99.6	57.5	78.3
MedViTv2-L [20]	97.7	99.9	96.7	82.3	81.7	95.0	95.2	99.6	97.3	99.7	57.8	78.5
	*93.0	*100.0	*93.6	*75.6	*83.0	*93.0	*94.8	*100.0	*97.0	*99.0	*50.3	*75.8
BitMedViT (ours)	**99.0**	**100.0**	**94.0**	**72.0**	**79.0**	**95.0**	**95.0**	**99.0**	**86.0**	**98.0**	**53.0**	**81.0**
	*91.1	*99.2	*93.7	*71.5	*76.2	*93.6	*85.7	*98.8	*88.0	*95.0	*51.3	*73.3
Model	BreastMNIST		BloodMNIST		TissueMNIST		OrganAMNIST		OrganCMNIST		OrganSMNIST	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
ResNet50 (224) [9]	84.2	86.6	95.0	99.7	68.0	93.2	94.7	99.8	91.1	99.3	78.5	97.5
MedMamba-S [41]	85.3	80.6	98.4	99.9	-	-	95.9	99.9	94.4	99.7	83.3	98.4
MedMamba-B [41]	89.1	84.9	98.3	99.9	-	-	96.4	99.9	94.3	99.8	83.4	98.3
MedViTv2-S [20]	89.5	94.7	98.5	99.9	70.5	93.9	96.6	99.9	95.0	99.8	83.9	98.6
MedViTv2-B [20]	90.4	94.9	98.5	99.9	71.1	94.2	96.9	99.9	95.3	99.8	84.4	98.7
MedViTv2-L [20]	91.0	95.3	98.7	99.9	71.6	94.3	97.3	99.9	96.1	99.9	85.1	98.7
	*86.5	*92.5	*98.5	*100.0	*75.7	*95.7	*84.6	*98.8	*86.7	*98.9	*82.7	*98.3
BitMedViT (ours)	**87.0**	**91.0**	**97.0**	**100.0**	**64.0**	**92.0**	**98.0**	**100.0**	**96.0**	**100.0**	**81.0**	**99.0**
	*82.1	*82.6	*97.5	*99.9	*63.8	*91.9	*90.2	*99.5	*88.4	*99.2	*74.1	*97.4

TABLE II

HARDWARE COMPARISON AGAINST STATE OF THE ART MEDICAL IMAGE CLASSIFICATION MODELS DEPLOYED WITHIN THE JETSON ORIN NANO. VALUES WITH * ARE RECOMPUTED WITH OUR METRICS

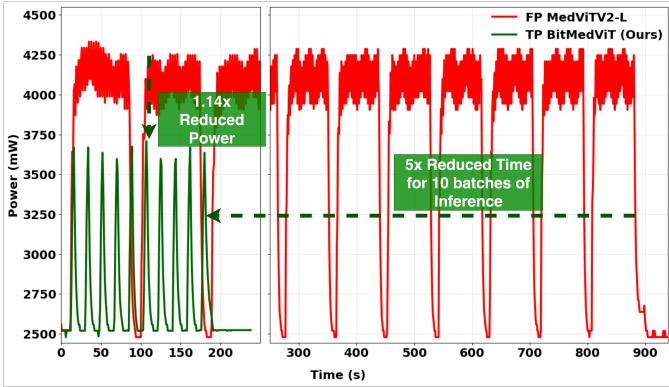| Work	Model Type	Precision	Parameters (M)	Model Size (MB)	Operations (GOPs)	Power (W)	Latency (ms)	Performance (GOPs / sec)	Energy Efficiency (GOPs / J)
MedViTv2-L (Baseline) [20]	ViT	Float32	117.26	447.71	12.9	4.25	366.63	35.3	8.31
MedMambaLite-ST [2]	Mamba	Float32	0.63	2.4	0.15	2.7	13.03	11.84	*4.39
BiTMedViT (ours)	**ViT**	**W2A8**	**8.65**	**10.5**	**11.53**	**3.72**	**16.88**	**683.06**	**183.62**

Fig. 6. Jetson Orin Nano Power Versus Time for the Ternary Precision (TP) BiTMedVit(Ours) and Full precision (FP) MedVitV2-L teacher. BiTMedViT achieves nearly a $1.14\times$ reduction in power while achieving nearly a $5\times$ reduction in latency for processing 10 batches of 50 inferences.

Figure 6 shows the power trace for executing an identical workload on the teacher and BiTMedViT on the Orin Nano platform: 50 inferences followed by 3 seconds of idle time, repeated 10 times. Compared to the teacher, BiTMedViT achieves a $5\times$ reduction in total inference time and a $1.14\times$ lower peak power consumption.

V. CONCLUSION

In this work, we presented BiTMedViT, a ternary-quantized ViT for efficient, real-time medical image classification on the edge. By integrating Multi-Query Attention, knowledge distillation, and hardware-aware CUDA & TensorRT optimization, BitMedViT achieves 86% accuracy on MedMNIST only 3% below its teacher while reducing parameters by $13.6\times$, model size by $43\times$, and memory transfers by $39\times$, at a 92% L2 cache hit rate and performing at 183.62 GOPs/J, $22\times$ that of MedViTv2-L [20] and $42\times$ that of MedMambaLite-ST [2]. These results demonstrate that extreme-precision quantization, combined with architectural and deployment co-design, enables SOTA ViT performance within the strict compute and memory constraints of clinical edge devices.

Future work will explore mixed-precision training techniques to dynamically adjust bit precisions between layers, alongside adaptive quantization based on input modality or task complexity. In addition, expanding BiTMedViT to CPU-only and specialized low-power devices will further enhance its deployability and accessibility across diverse clinical environments and devices

REFERENCES

[1] Romina Aalishah et al. Mambalitesr: Image super-resolution with low-rank mamba using knowledge distillation. In *2025 26th International Symposium on Quality Electronic Design (ISQED)*, pages 1–8. IEEE, 2025.

[2] Romina Aalishah et al. Medmambalite: Hardware-aware mamba for medical image classification. *arXiv preprint arXiv:2508.05049 [Accepted in IEEE BioCAS 2025]*, 2025.

[3] Joshua Ainslie et al. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[4] Al Ameen et al. Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of medical systems*, 36(1):93–101, 2012.

[5] Rohan Anil et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[6] Chao Chen et al. A review of convolutional neural network based methods for medical image classification. *Computers in biology and medicine*, 185:109507, 2025.

[7] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Hao Guan et al. Federated learning for medical image analysis: A survey. *Pattern recognition*, 151:110424, 2024.

[9] Kaiming He et al. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Morteza Hosseini et al. Cyclic sparsely connected architectures for compact deep convolutional neural networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(10): 1757–1770, 2021.

[11] Uttej Kallakuri et al. Resource-aware saliency-guided differentiable pruning for deep neural networks. In *Proceedings of the Great Lakes Symposium on VLSI 2024*, pages 694–699, 2024.

[12] Uttej Kallakuri et al. Enabling on-device medical ai assistants via input-driven saliency adaptation. *arXiv preprint arXiv:2506.11105 [Accepted in IEEE BioCAS 2025]*, 2025.

[13] Uttej Kallakuri et al. Magrip: Magnitude and gradient-informed pruning for task-agnostic large language models. *ACM Trans. Embed. Comput. Syst.*, September 2025. ISSN 1539-9087. doi: 10.1145/3766068. URL https://doi.org/10.1145/3766068. Just Accepted.

[14] Hee E Kim et al. Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):69, 2022.

[15] Jangho Kim et al. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019.

[16] Nikita Kitaev et al. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[17] Shuming Ma et al. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 1(4), 2024.

[18] Nitheesh Kumar Manjunath et al. An energy efficient edgeai autoencoder accelerator for reinforcement learning. *IEEE Open Journal of Circuits and Systems*, 2:182–195, 2021.

[19] Omid Nejati Manzari et al. Medvit: a robust vision transformer for generalized medical image classification. *Computers in biology and medicine*, 157:106791, 2023.

[20] Omid Nejati Manzari et al. Medical image classification with kan-integrated transformers and dilated neighborhood attention. *arXiv preprint arXiv:2502.13693*, 2025.

[21] Mason Marks and Claudia E Haupt. Ai chatbots, health privacy, and challenges to hipaa compliance. *Jama*, 330(4):309–310, 2023.

[22] Arnab Neelim Mazumder and Tinoosh Mohsenin. Reg-tunev2: A hardware-aware and multiobjective regression-based fine-tuning approach for deep neural networks on embedded platforms. *IEEE Micro*, 43(6):74–83, 2023.

[23] Microsoft. Bitnet gpu inference kernels, 2025. URL https://github.com/microsoft/BitNetu. Accessed: 2025-08-13.

[24] NVIDIA Corporation. NVIDIA Nsight Compute. https://docs.nvidia.com/nsight-compute, 2024. Accessed: 2025-07-30.

[25] NVIDIA Corporation. NVIDIA Nsight Systems. https://developer.nvidia.com/nsight-systems, 2024. Accessed: 2025-07-30.

[26] NVIDIA Corporation. NVIDIA TensorRT: High-performance deep learning inference optimizer and runtime. https://developer.nvidia.com/tensorrt, 2024. Accessed: 2025-07-30.

[27] Sangjoon Park et al. Self-evolving vision transformer for chest x-ray diagnosis through knowledge distillation. *Nature communications*, 13(1):3848, 2022.

[28] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[29] Hasib-Al Rashid et al. Tinyvqa: Compact multimodal deep neural network for visual question answering on resource-constrained devices. *CoRR*, 2024.

[30] Hasib-Al Rashid et al. Hac-m-dnn: Hardware aware compression of sustainable multimodal deep neural networks for efficient tinyml deployment. In *2025 IEEE Conference on Technologies for Sustainability (SusTech)*, pages 1–7. IEEE, 2025.

[31] Md Eshham Rayed et al. Deep learning for medical image segmentation: State-of-the-art advancements and challenges. *Informatics in medicine unlocked*, 47:101504, 2024.

[32] Ramprasaath R Selvaraju et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[33] Aynur Sevinc et al. A distillation approach to transformer-based medical image classification with limited data. *Diagnostics*, 15 (7):929, 2025.

[34] Hugo Touvron et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[35] Jinheng Wang et al. Bitnet. cpp: Efficient edge inference for ternary llms. *arXiv preprint arXiv:2502.11880*, 2025.

[36] Sheng Xu et al. Tervit: An efficient ternary vision transformer. *arXiv preprint arXiv:2201.08050*, 2022.

[37] Jiancheng Yang et al. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

[38] Jiancheng Yang et al. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[39] Chenyang Yin et al. Tereffic: Highly efficient ternary llm inference on fpga. *arXiv preprint arXiv:2502.16473*, 2025.

[40] Zhengqing Yuan et al. Vit-1.58 b: Mobile vision transformers in the 1-bit era. *arXiv preprint arXiv:2406.18051*, 2024.

[41] Yubiao Yue et al. Medmamba: Vision mamba for medical image classification. *arXiv preprint arXiv:2403.03849*, 2024.

[42] Yichi Zhang et al. Segment anything model for medical image segmentation: Current applications and future directions. *Computers in Biology and Medicine*, 171:108238, 2024.

[43] Ke Zhu et al. Quantized feature distillation for network quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11452–11460, 2023.

[44] Lianghui Zhu et al. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.