FTerViT: Fully Ternary Vision Transformer

Szymon Ruciński[1,2] **Pietro Bonazzi**[2]
szymon.rucinski@csem.ch pbonazzi@ethz.ch
Engin Türetken[1] Simon Narduzzi[1] Michele Magno[2] Nadim Maamari[1]
[1]CSEM, Neuchâtel, Switzerland [2]ETH Zürich, Zürich, Switzerland

Abstract

Ternary Vision Transformers offer substantial model compression, however state-of-the-art methods only ternarize the encoder layers, leaving patch embeddings, LayerNorm parameters, and classifier heads in full precision. In compact models targeting resource-constrained processors, such as microcontrollers, these remaining full-precision components determine the total memory footprint, severely limiting deployment efficiency and on-device feasibility. In this work, we introduce a fully ternarized Vision Transformer in which *all* weight matrices and normalization parameters are ternarized, while activations remain 8-bit (FTerViT). To this end, we introduce two novel operators: TernaryBitConv2d with per-channel scaling for patch embedding and TernaryLayerNorm. FTerViT is trained using knowledge distillation, followed by a lightweight quantization-aware recovery phase. On a matched DeiT-S backbone, FTerViT reaches 77.47% ImageNet-1K top-1 at 5.81 MB, +3.27 pp over TerViT. With the stronger DeiT-III-S backbone at 384×384 resolution, our ternary W2A8 model achieves 82.43% at 6.09 MB ($\sim$15× compression, $-$2.42 pp vs. FP32). This backbone- and resolution-advantaged result exceeds prior ternary ViTs by up to 8 pp. Finally, we demonstrate the first implementation of ternary vision transformers on a dual-core Xtensa LX7 microcontroller inside the ESP32-S3 system-on-chip. By deploying FTerViT-Small (based on DeiT-III-Small at 224×224 resolution, 5.81 MB), we achieve 79.64% ImageNet-1K top-1 accuracy.

1 Introduction

Vision Transformers (ViTs) [1, 2, 3] are strong image classifiers, yet their substantial memory footprint makes them poorly suited for microcontroller-class devices. A standard compact ViT like DeiT-Small [3] requires 88.3 MB in FP32, while typical microcontroller units (MCUs) offer only a few megabytes of external RAM. This mismatch is particularly problematic for duty-cycled, event-triggered low-power vision applications where cloud offloading is undesirable or infeasible [4].

Ternary quantization offers a compelling solution [5, 6]. By constraining the weights to $\{-1, 0, +1\}$ and packing them in 2 bits, ternary models can theoretically achieve 16× weight compression relative to FP32 models. However, existing low-bit ViTs [6, 5, 7, 8, 9, 10, 11, 12, 13, 14], only ternarize the encoder layers, leaving the patch embedding, LayerNorm, and classifier head at INT8 or FP32. While at moderate bitwidths, these exceptions are tolerable; at 2-bits, they become the dominant bottleneck. In DeiT-Tiny [2], the non-ternary components account for less than 4% of parameters yet consume 38% of the model size (Fig. 1b). Our proposed method ternarizes the patch embedding, LayerNorm, and classifier head jointly (Table 3).

This design choice stems from well-known sensitivity of quantizing the first and last layers [15, 13, 14, 10, 11] predating the invention of ViTs [16, 17, 18]. Recently, TerViT [6], reports significant accuracy drops (22.4 pp) for full-ternarization of the patch embedding and final classifier. In TerViT,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

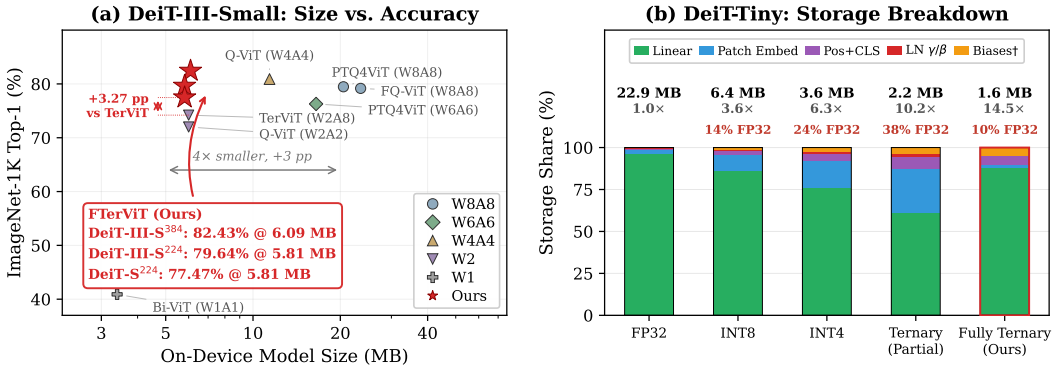


Figure 1: **(a)** DeiT-III-Small size vs. accuracy; FTerViT (W2A8): 82.43% at 6.09 MB (384^2) / 79.64% at 5.81 MB (224^2). FTerViT based on DeiT-S^{224} reaches 77.47% under ternary quantization. **(b)** DeiT-Tiny storage; partial-W2 leaves 38% of bytes at FP32, fully ternary drops the share to 10%. DeiT-III-Small follows the same trend (24% partial, 4% fully ternary; 88.3 MB $\rightarrow$ 5.81 MB).

LayerNorm parameters are similarly left un-quantized due to activation outliers and inter-channel variation [19, 20, 21, 22, 23, 24, 25]. DeiT-Small [2] showcases its patch embedding as the single most sensitive layer ($4.7 \pm 0.5\%$ of total importance, Section 3.1) under a Taylor first-order analysis [26].

The resulting model substantially narrows the gap between standard ViT accuracy and MCU-scale deployment constraints. FTerViT-Small based on DeiT-III-S^{224} occupies 5.81 MB and reaches 79.64% ImageNet-1K top-1. Based on DeiT-III-S^{384}, it reaches 82.43% at 6.09 MB, losing 2.42 pp from FP32 and reducing the TerViT-DeiT-S accuracy gap from 5.7 pp to 2.4 pp (Fig. 1a). The 224×224 model runs on a $10 ESP32-S3 with 8 MB PSRAM, while the original 88.3 MB FP32 model remains far beyond the device memory budget.

The main contributions of the paper are summarized as follows:

- We are the first to show that the most fragile components of ViT: patch embedding, LayerNorms, and classifier head can be ternarized to $\{-1, 0, +1\}$ when trained with knowledge distillation. Our work pushes the boundary of what was considered quantizable.
- Our approach maintains strong accuracy at extreme compression: 82.43% top-1 on ImageNet-1K at 6.09 MB (14.6$\times$ compression), losing 2.42 pp from FP32. Our FTerViT-DeiT-S achieves 3.27 pp higher ImageNet-1K top-1 than TerViT-DeiT-S [6].
- We propose a simple yet effective training strategy based on same-architecture knowledge distillation and lightweight recovery fine-tuning, enabling substantially faster convergence and improved final accuracy over prior ternary ViT training approaches.
- We demonstrate a standalone C implementation of a ternary Vision Transformer on the dual-core ESP32-S3, a resource-constrained commodity MCU platform with only 8MB of memory. We validate the practical feasibility of fully ternary ViTs for resource-constrained edge devices.

2 Methodology

FTerViT is designed to enable fully ternary Vision Transformers retaining competitive accuracy while satisfying the strict memory constraints (below 10MB) of microcontroller-class hardware. Achieving this goal is challenging because several components beyond the transformer encoder – most notably patch embeddings and normalization layers – are highly sensitive to quantization and are therefore typically retained in FP32 by prior work.

This section presents the FTerViT, which is a full ternary adaptation of DeiT-Tiny [2] (5.5M params, 22.9 MB FP32) and DeiT-Small [3] (22.1M params, 88.3/88.9 MB FP32 at 224/384 resolution) trained on ImageNet-1K. The main design of FTerViT is structured around two core elements: (1) a complete set of ternary primitives (Section 2.1) and (2) a two-phase knowledge distillation procedure that enables stable quantization (Section 2.2).

2.1 Ternary Primitives: TernaryBitLinear, TernaryBitConv2d, TernaryLayerNorm

We replace the three types of weight-carrying components in a ViT with dedicated ternary modules: TernaryBitLinear for fully-connected layers, TernaryBitConv2d for the patch-embedding convolution, and TernaryLayerNorm for LayerNorm affines. Together, these operators eliminate all remaining FP32 parameter storage while maintaining compatibility with standard ViT architectures. The claim is fully ternary *weights and normalization affines*, not ternary arithmetic: activations are quantized to 8 bits during training (Eq. (4)) and kept at FP32 in our MCU kernels (Section 4).

TernaryBitLinear: As defined in BitNet b1.58 [27], TernaryBitLinear quantizes weights to $\{-1, 0, +1\}$ and activations to 8-bit integers. For a weight matrix $\mathbf{W} \in \mathbb{R}^{n\times m}$, the ternary quantization is:

$$\tilde{\mathbf{W}} = s_w \cdot \mathrm{RoundClip}\left(\frac{\mathbf{W}}{s_w + \epsilon},\ -1,\ +1\right), \tag{1}$$

where the clamped rounding operator is:

$$\mathrm{RoundClip}(x, a, b) = \max\big(a,\ \min(b,\ \mathrm{round}(x))\big), \tag{2}$$

and s_w is the per-tensor *absmean* (mean absolute value) of the weight matrix:

$$s_w = \mathrm{absmean}(\mathbf{W}) = \frac{1}{nm}\sum_{i,j}\big|\mathbf{W}_{ij}\big|. \tag{3}$$

Absmean is preferred over max-based scaling because $\max|W|$ obviously pushes most ratios $|W|/s_w \ll 1$ into the zero bin of the ternarized grid. Consistent with this intuition, TWN [28] similarly derives the optimal threshold $\Delta^* \approx 0.75\,\mathbb{E}|W|$ analytically for Gaussian weights, well below $\max|W|$.

Following BitNet b1.58 [27], we RMS-normalize and quantize input activations of a layer to 8-bit integers via per-token absmax scaling,

$$s_x = \frac{127}{\max\limits_i |x_i|}, \qquad \tilde{x} = \mathrm{RoundClip}(s_x\, x,\ -128,\ 127)\,/\,s_x. \tag{4}$$

Gradients pass through both quantization operations via the straight-through estimator (STE) [29].

TernaryBitConv2d: For the patch embedding layer, we introduce TernaryBitConv2d, which applies per-channel scaling to better handle heterogeneous filter magnitudes [30, 31]. For an activation map $x^{(c)} \in \mathbb{R}^{H\times W}$, let $\Omega = \{1,\dots,H\}\times\{1,\dots,W\}$ denote its spatial grid. Each channel c has its own scales:

$$s_w^{(c)} = \frac{1}{K}\sum_k \big|\mathbf{W}_{\mathrm{conv}}^{(c)}[k]\big|, \qquad s_x^{(c)} = \frac{127}{\max\limits_{(h,w)\in\Omega}\big|x^{(c)}[h,w]\big|}, \tag{5}$$

where K is the number of kernel elements, and H, W denote spatial size. The activation scale uses one scalar per sample and channel.

TernaryLayerNorm: For LayerNorm layers, we ternarize the learnable affine parameters γ (scale) and β (shift):

$$\mathrm{TernaryLN}(\mathbf{x}) = \tilde{\gamma} \odot \frac{\mathbf{x} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \tilde{\beta}, \tag{6}$$

where $\tilde{\gamma}$ and $\tilde{\beta}$ use the same absmean scheme as weights (3). Per-token statistics (μ, σ^2), as well as biases, are kept at FP32 since their parameter count is negligible and ternarizing them would destroy positional information. The 25 LayerNorms hold $<$0.2% of parameters but account for 34–39% of Taylor-FO importance and 21–28% of Hessian-trace importance (Section 3.1).

2.2 Training Protocol

Our ternary student is initialized directly from the pretrained FP32 teacher and trained via quantization-aware distillation (QAD). Training proceeds in two phases. The first phase runs at learning rate (LR) $1\mathrm{e}{-4}$ with cosine decay until validation top-1 saturates (reaching 76.78% on DeiT-III-S^{224}). During

the second phase, we restart LR at $1e{-}5$ with cosine decay over 10 epochs at the same loss (see Table 1). The reduction in LR enables recalibration and recovery of the final scale from Phase 1 saturation, lifting top-1 by 3–4 pp across distillation settings (Table 5).

During both steps, FTerViT is trained using forward-KL distillation [32], to match its pretrained frozen full-precision ViT equivalent [9]. Given student logits z_S and teacher logits z_T, the FTerViT loss can be simply described as:

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}\big(\mathrm{softmax}(z_T) \,\|\, \mathrm{softmax}(z_S)\big), \tag{7}$$

Prior ternary and low-bit ViTs adopt a single-phase loss, ranging from label-only cross-entropy (CE) without distillation [6, 5], attention or query/key similarity matching [13, 33], soft-logit distillation [11, 14], hard-label distillation [12], multi-step knowledge-distillation (KD) across bit-precisions [34], to combined CE+KL+feature objectives [7].

In contrast, we use only the KL term since a cross-entropy term conflicts with KL in low-bit networks [35, 36, 37]. In our experiments, we set the distillation temperature to $T{=}1$ [38, 39, 40].

Table 1: Two-phase training hyperparameters. Both phases minimize only the KL loss; only the cosine learning-rate schedule changes.

	Phase 1 (Training)	Phase 2 (Fine-tuning)
Loss	$\mathcal{L}_{\mathrm{KL}}$ ($T{=}1$)	$\mathcal{L}_{\mathrm{KL}}$ ($T{=}1$)
Learning rate (cosine decay)	$1e{-}4$ (250 ep)	$1e{-}5$ (10 ep)
Optimizer	AdamW [41]	AdamW [41]
Weight decay	0.05	0.01
Batch	1024	512
Teacher	Pretrained FP32 (frozen)	Pretrained FP32 (frozen)
Student init	Ternarized from teacher	Phase 1 checkpoint

3 Experiments

3.1 Layer Sensitivity Analysis

A core obstacle to full ternarization of Vision Transformers has been the well-documented sensitivity of the patch embedding and LayerNorm layers [6, 15, 13]. Prior work therefore retained these components in higher precision.

To quantify this sensitivity, we measure per-component importance on ImageNet-1K using two established estimators: Taylor first-order (FO) importance [26] and Hessian-trace approximation (HAWQ-style [18]).

Table 2: Per-component importance share on ImageNet-1K (mean $\pm$ SEM). LayerNorm and patch embedding together dominate importance despite negligible parameter counts, explaining why prior ternary ViTs left them in higher precision. "Top quantizable layer" excludes non-ternarized positional/class embeddings.

	DeiT-Tiny (5.7 M)			**DeiT-Small (22.1 M)**		
Component	Params (%)	Taylor-FO (%)	Hess. (%)	Params (%)	Taylor-FO (%)	Hess. (%)
LayerNorm	0.17	$\mathbf{39.4 \pm 1.2}$	$\mathbf{28.0 \pm 0.1}$	0.09	$\mathbf{34.1 \pm 1.0}$	$\mathbf{20.7 \pm 0.4}$
FFN (FC1, FC2)	62.1	33.2 ± 0.9	28.7 ± 0.2	64.3	$\mathbf{25.9 \pm 0.6}$	30.2 ± 0.7
Attention (Q, K, V, Proj)	31.1	18.4 ± 0.4	15.0 ± 0.1	32.2	17.4 ± 0.3	19.3 ± 0.2
LayerScale		—		0.04	15.7 ± 1.2	9.4 ± 0.7
Patch embed	2.6	$\mathbf{3.3 \pm 0.1}$	$\mathbf{4.4 \pm 0.1}$	1.3	$\mathbf{4.6 \pm 0.4}$	$\mathbf{8.6 \pm 0.7}$
Classifier head	3.4	1.3 ± 0.1	0.2 ± 0.0	1.7	0.3 ± 0.0	0.01 ± 0.00
↓ Sorted by Taylor-FO importance (descending) on DeiT-Small						

As shown in Table 2, all LayerNorms combined account for 21–39% of total importance while occupying $<0.2\%$ of parameters. Furthermore, among all individual layers in DeiT-Small, patch embedding is the single most important one under both metrics. These findings directly explain the 22.4 pp accuracy drop reported by TerViT when attempting full ternarization [6].

3.2 Benchmark Results

We first compare FTerViT against prior quantized ViTs on ImageNet-1K. As shown in Table 3, FTerViT achieves the best reported accuracy among ternary DeiT-S models, reaching **77.47%** top-1 with 2-bit weights and 8-bit activations. A 5-seed replication of Phase 2 gives 77.45 ± 0.07 for FTerViT-DeiT-S and 79.70 ± 0.08 for FTerViT-DeiT-III-S^{224} (mean $\pm$ std), with the published single-run values inside both intervals. The seed noise is roughly $40\times$ smaller than the +3.27 pp margin over TerViT. Keeping the patch embedding, classifier head, and LayerNorm affines at INT8 under the identical two-phase recipe yields 78.28% at 6.33 MB. Full ternarization costs 0.81 pp and buys the final 9% of compression, so both operating points come from one unchanged recipe. Per-component, scaling-granularity, and distillation controls on the same DeiT-S backbone are reported in Section A.2. Unlike the methods we compare against, FTerViT applies ternary quantization not only to the transformer blocks but also to the patch embedding, normalization layers, and classifier head, resulting in a fully ternary model that achieves both the best reported accuracy and the highest compression among ternary DeiT-S models. Starting from the stronger DeiT-III-S backbone further improves performance: the 224×224 variant reaches **79.64%** top-1 (-3.44 pp from its FP32 teacher), while the 384×384 variant achieves **82.43%** top-1 (-2.42 pp).

Table 3: Comparison of quantized ViT methods on ImageNet-1K (Top-1). W/A = weight/activation bits. Rows with 1- or 2-bit activations (Bi-ViT, Q-ViT, LSQ) use a lower activation precision than FTerViT, their accuracies are listed for context. The ternary comparators (ViT-1.58b, TerViT) are W2A8, matching FTerViT. FTerViT is *fully* ternary.

Method	W	A	Model	Size (MB)	Comp.	Regime	Epochs	Top-1
Bi-ViT [10]	1	1	DeiT-S	3.4	26×	QAT	300	40.9
BiViT [9]	1	mixed	Swin-S	15.4	13×	QAT	300	75.6
PTQ4ViT [20]	8	8	DeiT-S	22	4×	PTQ	32 imgs	79.47
RepQ-ViT [22]	6	6	DeiT-S	16.7	5.3×	PTQ	32 imgs	78.90
RepQ-ViT [22]	4	4	DeiT-S	11	8×	PTQ	32 imgs	69.03
Q-ViT [13]	2	2	DeiT-S	6.0	14.7×	QAT	300	72.1
LSQ [42, 13]	2	2	DeiT-S	6.0	14.7×	QAT	300	68.0
OFQ [14]	4	4	DeiT-S	11.4	7.7×	QAT	325	81.10
TernaryViT								
ViT-1.58b [5]	2	8	ViT-L	57	20×	Scratch	500+	74.25
TerViT [6]	2	8	DeiT-S	6.0	14.7×	QAT+PT	300*	74.2
FTerViT (Ours)	2	8	DeiT-S	**5.81**	**15.2**×	QAD	**260**	**77.47**
FTerViT (Ours)	2	8	DeiT-III-S^{224}	**5.81**	**15.2**×	QAD	**260**	**79.64**
FTerViT (Ours)	2	8	DeiT-III-S^{384}	**6.09**	**14.6**×	QAD	**260**	**82.43**

FTerViT uses DeiT-III-S$^{224}/^{384}$ [3], whose FP32 baseline (83.08–84.85%) is higher than the DeiT-S baseline (79.86%). Furthermore, our method generalizes well to other image classification benchmarks. As reported in Appendix A.1, our DeiT-Tiny achieves 97.43% and 86.01% top-1 accuracy on CIFAR-10 and CIFAR-100, respectively. These numbers are nearly on par with the full-precision DeiT-Tiny (97.52% / 86.54%) while using only 1.53 MB of storage, equivalent to $15\times$ reduction in model size.

Two-Phase Training Effectiveness. To analyze the contribution of the proposed training strategy, we study Phase 1 (high learning rate training) and Phase 2 (low learning rate recovery). As shown in Table 4, while Phase 1 converges slowly toward a stable accuracy, Phase 2 achieves substantial recovery within only a few epochs.

Table 4: FTerViT results across datasets and input resolutions. Phase 1 (250 epochs) denotes saturation performance; Phase 2 (+10 epochs) denotes final fine-tuned results.

Model	Dataset	Input Resolution	FP32	Phase 1	Phase 2
DeiT-Tiny	CIFAR-10	224×224	97.52%	97.43%	**97.43%**
DeiT-Tiny	CIFAR-100	224×224	86.54%	84.42%	**86.01%**
DeiT-Tiny	ImageNet-1K	224×224	72.19%	59.54%	**63.00%**
DeiT-Small	ImageNet-1K	224×224	79.86%	75.05%	**77.47%**
DeiT-III-Small	ImageNet-1K	224×224	83.08%	76.78%	**79.64%**
DeiT-III-Small	ImageNet-1K	384×384	84.85%	78.63%	**82.43%**

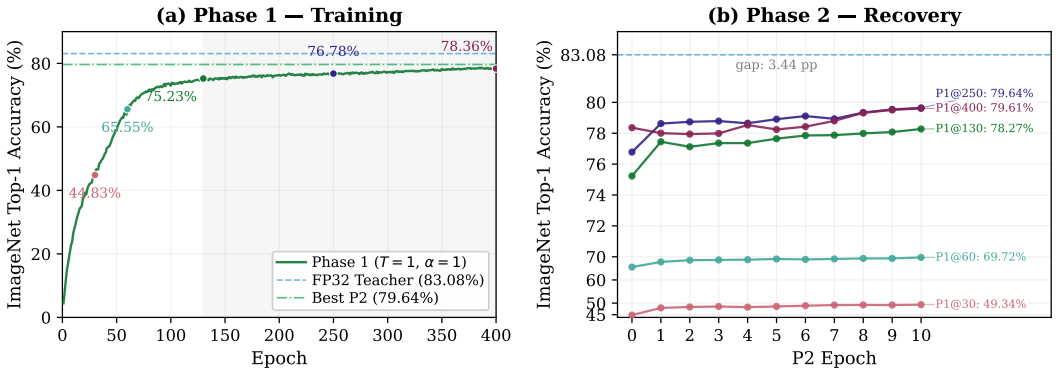


Figure 2: **(a)** Phase 1 training of DeiT-III-S^{224} on ImageNet-1K for 250 epochs. Validation top-1 saturates near 78% and never bridges the gap to FP32 (83.08%). **(b)** Phase 2 fine-tuning from five P1 checkpoints (epochs 30–400). P1@250 converges to ∼79.64% top-1 (−3.44 pp vs. FP32) in 10 epochs. P1@400 reaches 79.61%, while early checkpoints (P1@30, P1@60) recover far less.

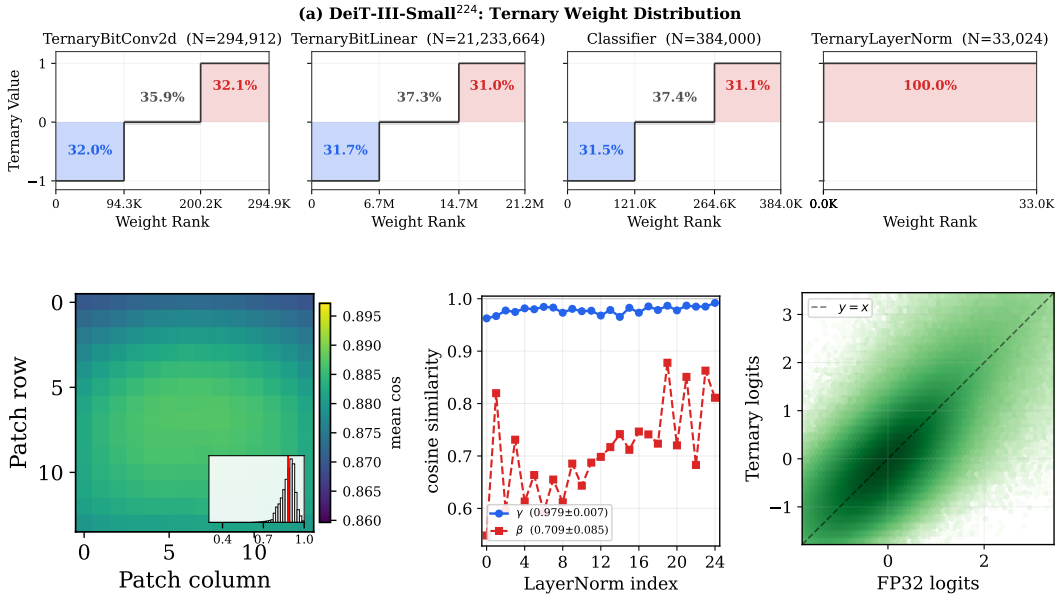
To better understand the source of performance gains in FTerViT, we analyze the optimization trajectory across both training phases on ImageNet-1K (DeiT-III-S^{224}). The key finding is that Phase 2 fine-tuning is more efficient than prolonging Phase 1: a 10-epoch low-LR restart from epoch 250 outperforms 150 additional Phase 1 epochs followed by the same restart as shown in Fig. 2. More specifically, we observe the following dynamics.

Phase 1: Saturation. Phase 1 converges to 76–78% top-1 under cosine decay. Accuracy gains slow past epoch 130, and extending training to epoch 400 improves the Phase 1 checkpoint by only 1.6 pp (76.78% → 78.36%).

Phase 2: Rapid Recovery. A low-LR restart recovers accuracy within 10 epochs regardless of Phase 1 maturity as shown in Table 5. Crucially, starting Phase 2 from epoch 250 (76.78%) reaches **79.64%**, marginally higher than starting from epoch 400 (79.61%), demonstrating that the 150 extra Phase 1 epochs yield no net benefit. We finetune five P1 checkpoints (epochs 30, 60, 130, 250, 400) to confirm this pattern.

Table 5: Phase 2 finetuning trajectory from five Phase 1 checkpoints (DeiT-III-S^{224}, ImageNet-1K). Same P2 recipe, LR cosine $1e{-}5 \rightarrow 1e{-}6$ over 10 epochs.

P2 Ep	LR	P1@30	P1@60	P1@130	P1@250	P1@400
0	—	44.83	65.55	75.23	76.78	**78.36**
1	9.78e−6	47.93	67.79	77.45	**78.63**	78.00
3	8.15e−6	48.54	68.63	77.36	**78.79**	77.99
5	5.50e−6	48.55	69.05	77.65	**78.91**	78.24
10	1.00e−6	49.34	69.72	78.27	**79.64**	79.61



(b) Patch embedding
$\overline{\cos} = 0.88 \pm 0.05$

(c) TernaryLayerNorm parameters
$\gamma = 0.979 \pm 0.007$, $\beta = 0.71 \pm 0.09$

(d) Classifier head logits
$\bar{r} = 0.81 \pm 0.06$

Figure 3: **Component-wise fidelity of fully ternary ViTs.** (a) Global distribution of weights constrained to $\{-1, 0, +1\}$. (b,c,d) FP32–ternary alignment across key components of DeiT-III-S^{224} shows strong preservation of representational structure despite ternary quantization.

3.3 Component-Level Fidelity of Ternarization

We provide a detailed breakdown of ternarization effects in Fig. 3. First, ternary weights are balanced across $\{-1, 0, +1\}$, with 37.3% zeros (8.18 M weights), inducing sparsity that directly reduces multiply-accumulate operations at inference time. This distribution is consistent across TernaryBitLinear and classifier layers, while the patch embedding exhibits a slightly reduced zero fraction. At the representation level, patch embedding features remain closely aligned with FP32, with mean cosine similarity 0.88 (std 0.05, 5th/95th percentile $0.79/0.95$), indicating stable spatial feature extraction.

TernaryLayerNorm scale parameters closely match FP32 (0.979 ± 0.007 cosine), while the shift is reproduced less precisely (0.71 ± 0.09). The ternary γ never uses the -1 level, but its per-layer absmean scale spans 0.63 to 3.86 across the 25 LayerNorms of DeiT-III-S^{224}, so it is not an identity transform. Setting γ to 1 at inference collapses the trained models ($77.51\% \rightarrow 0.17\%$ for DeiT-S, $79.62\% \rightarrow 4.44\%$ for DeiT-III-S^{224}). Training DeiT-S with γ fixed to 1 from the start reaches 77.46% vs. 77.47%, so the scaling is essential but the network can learn it elsewhere when trained without γ from the beginning.

Finally, output behavior remains consistent: classifier logits achieve mean Pearson correlation $r = 0.81$ (std 0.06; pooled $r = 0.79$, $p < 10^{-300}$), indicating that class rankings are largely preserved despite quantization.

3.4 CLS Token Analysis

To gain deeper insights into the effectiveness of knowledge distillation for low-bit ternary models, we examine how the ternary student's internal attention patterns (FTerViT-DeiT-III-S^{224}) and component representations align with those of the full-precision (FP32) teacher. Attention rollout maps [43] computed from the final CLS token provide a principled way to visualize where each model directs its focus across the image. As shown in Figure 4, the ternary student consistently attends to the same salient semantic regions as the FP32 teacher across a diverse set of ImageNet-1K classes, indicating successful transfer of high-level visual understanding despite aggressive quantization.

4 On-Device Deployment and Profiling

We deploy the FTerViT based on DeiT-III-S^{224} on a dual-core 32-bit LX7 microprocessor named ESP32-S3-EYE, shown in Fig. 5, running at 240 MHz. Our ternary model occupies 2.83 MB PSRAM at peak (FP32 activations and input image) and 5.81 MB flash (2-bit packed weights), leaving $\sim$4.5 MB PSRAM free on-device for camera and LCD buffers. In addition, we implement a standalone pure-C inference engine that executes all ternary layers without external dependencies. Ternary weights are bit-packed (4 weights per byte) into a 5.81 MB binary, and kernels perform integer multiply-accumulate with fused QKV projections.

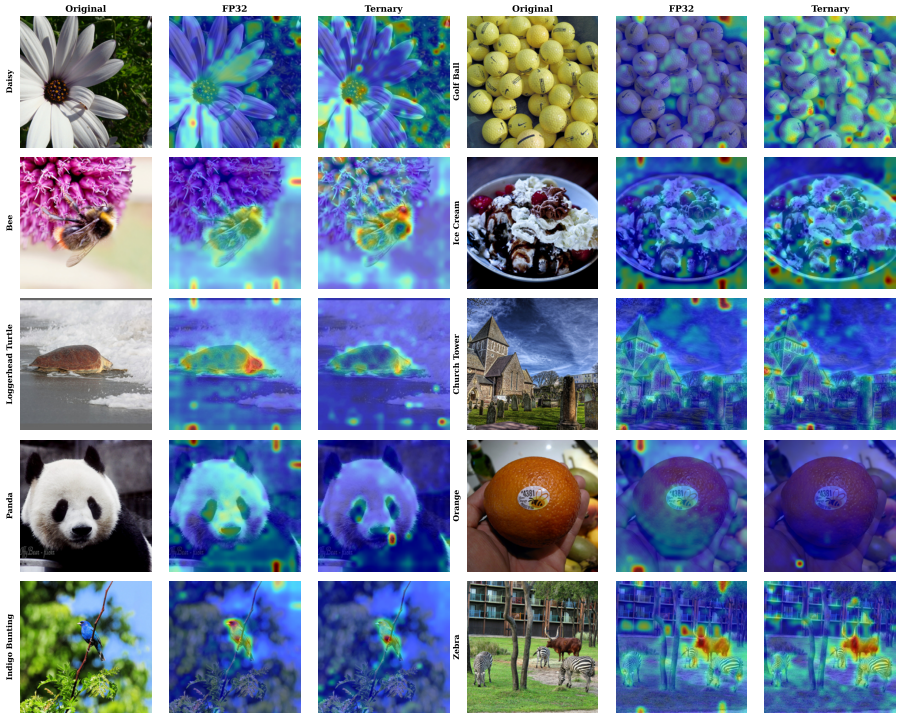


Figure 4: Attention rollout on 10 additional ImageNet-1K classes. FTerViT-DeiT-III-S^{224} consistently attends to the same semantic regions as the FP32 teacher across diverse object categories.



Figure 5: ESP32-S3-EYE board running on-device DeiT-III-S^{224} ternary inference ($10, dual-core Xtensa LX7 at 240 MHz, 8 MB PSRAM, 2 MP camera, 240$\times$240 LCD). The 5.81 MB ternary model fits in flash (79% partition utilization); the original 88.3 MB FP32 checkpoint is 15.2$\times$ larger and cannot load.

Our ternary model can be deployed and executed entirely on-device. A forward pass takes 21.06 s, with attention (Q@K^T+softmax+V) and FFN each accounting for $\approx$31% of runtime, and fused

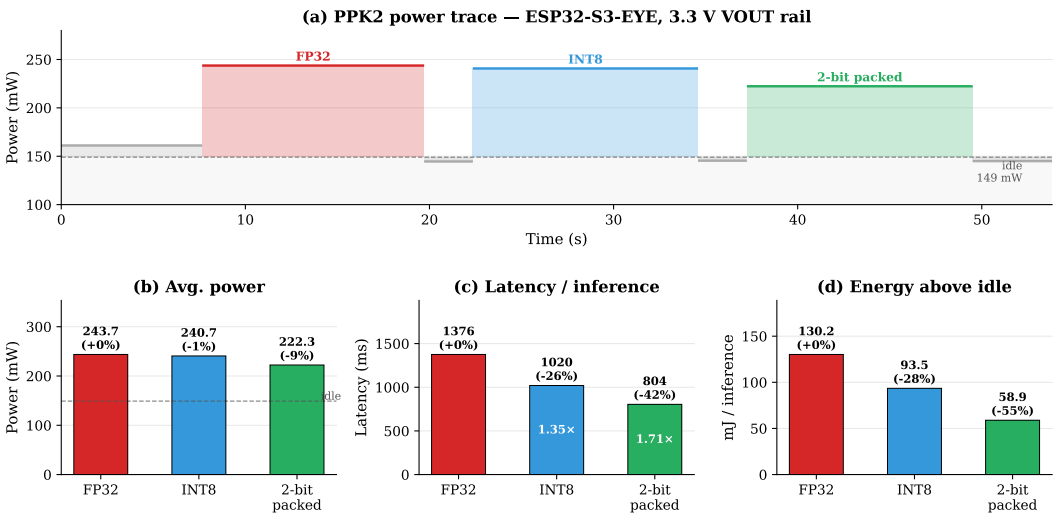


Figure 6: PPK2 power measurements on ESP32-S3-EYE (Nordic PPK2 probe, VOUT rail). **Top:** raw power trace across FP32, INT8, and 2-bit packed inference phases. Dashed line = board idle (149 mW). **Bottom:** per-format power, latency, and above-idle energy per inference. On this single FC1 kernel, packed ternary is 1.71× faster and uses 54.7% less above-idle energy than FP32 (kernel-level, not end-to-end).

QKV projections contributing a further 8.3% (see Table 8 in Section B.1). At 21 s per inference the deployment is a feasibility result rather than a real-time system. It targets duty-cycled sensing, such as a plant-disease check from one image per node per hour or periodic state monitoring of unpowered infrastructure, where the device sleeps between inferences.

Power measurements. We measure power on the ESP32-S3-EYE VOUT rail with a Nordic PPK2 probe (Fig. 6). On the DeiT-Tiny FC1 kernel (192→768), packed ternary is **1.71× faster** than FP32 (804 ms vs. 1376 ms). This is a kernel-level measurement, not end-to-end. Subtracting board idle (149 mW), above-idle energy drops **55%** (59 mJ vs. 130 mJ). Two effects compound: 4× smaller weight storage reduces memory-bandwidth pressure, explaining the lower latency; reduced data movement lowers active power (222 mW vs. 244 mW, $\approx -9\%$). Active-compute power falls **22.6%** (94.8 mW → 73.4 mW).

Comparison with prior MCU implementation of ViT. Prior MCU-scale Vision Transformers typically rely on neural architecture search combined with INT8 quantization to meet memory constraints. For example, MCUFormer [44] achieves 73.62% on ImageNet-1K at 0.90 MB, Tiny-Former [45] reaches 96.10% on CIFAR-10 at 0.91 MB, and LMaNet-Elite [46] reports 74.50% on CIFAR-100 under 1 MB. In contrast, FTerViT follows an orthogonal approach: instead of redesigning the architecture, we compress DeiT-III-S^{224} from 88.3 MB to 5.81 MB via ternarization, achieving 79.64% on ImageNet-1K (+6.02 pp over MCUFormer). INT8 MobileNets fit the same device at lower accuracy: MobileNetV3-Large reaches 73.8% at 5.4 MB and MobileNetV3-Small 64.9% at 2.5 MB [47], while MobileNetV4-Conv-S reaches 73.8% at FP32 (3.8 MB INT8) and the larger V4-Conv-M/L (9.2 and 31 MB) exceed the 8 MB budget [48].

5 Conclusion

FTerViT shows that all weight matrices and normalization parameters in a ViT can be constrained to $\{-1, 0, +1\}$ with minimal accuracy loss compared to FP32. One finding stands out: KD from a same-architecture teacher can fully ternarize ViT architecture's most sensitive components like the patch embedding, layernorm and classifier head that prior work found extremely sensitive. Our compression pipeline's primitives (like TernaryBitLinear, TernaryBitConv2d, TernaryLayerNorm) can also be used as standalone components in novel lightweight architectures. We show that the gap to FP32 scales inversely with model capacity [49]: 9.19 pp for DeiT-Tiny vs. 2.42 pp for DeiT-III-S^{384}.

In ImageNet-1K, FTerViT achieves 82.43% at 6.09 MB, surpassing previous ternary ViTs at higher compression, while 5.81 MB DeiT-III-S^{224} deploys on a $10 ESP32-S3.

Limitations. We evaluate DeiT-Tiny and DeiT-Small. Larger models exceed our MCU memory budget, so we leave scaling untested. The C inference kernel uses basic bit-unpacking without optimization. The two-stage pipeline (training + fine-tuning) could potentially be unified into a single pass.

Acknowledgments and Disclosure of Funding

This research is supported by the Swiss National Science Foundation (219943) and SwissChips, a national initiative led by ETH Zürich, EPFL, and CSEM with funding from the State Secretariat for Education, Research and Innovation (SERI) to strengthen Switzerland's semiconductor and IC-design ecosystem.

References

[1] 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. *arXiv preprint arXiv:2010.11929*, 2020. URL `https://arxiv.org/abs/2010.11929`.

[2] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML 2021*, volume 139, pages 10347–10357, 2021. URL `https://arxiv.org/abs/2012.12877`.

[3] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *ECCV 2022*, pages 516–533, 2022. doi: 10.1007/978-3-031-20053-3_30.

[4] Pietro Bonazzi, Thomas Rüegg, Sizhen Bian, Yawei Li, and Michele Magno. TinyTracker: Ultra-fast and ultra-low-power edge vision in-sensor for gaze estimation. In *IEEE SENSORS 2023*, pages 1–4, 2023. doi: 10.1109/SENSORS56945.2023.10325167.

[5] Zhengqing Yuan, Rong Zhou, Hongyi Wang, Lifang He, Yanfang Ye, and Lichao Sun. ViT-1.58b: Mobile vision transformers in the 1-bit era. *arXiv preprint arXiv:2406.18051*, 2024. URL `https://arxiv.org/abs/2406.18051`.

[6] Sheng Xu, Yanjing Li, Teli Ma, Bohan Zeng, Baochang Zhang, Peng Gao, and Jinhu Lv. TerViT: An efficient ternary vision transformer. *arXiv preprint arXiv:2201.08050*, 2022. URL `https://arxiv.org/abs/2201.08050`.

[7] Mikolaj Walczak, Uttej Kallakuri, Edward Humes, Xiaomin Lin, and Tinoosh Mohsenin. Invited paper: BitMedViT: Ternary-quantized vision transformer for medical AI assistants on the edge. In *ICCAD 2025*, pages 1–7, 2025. doi: 10.1109/ICCAD66269.2025.11240999.

[8] Shu-Hao Zhang, Wei-Cheng Tang, Chen Wu, Peng Hu, Nan Li, Liang-Jie Zhang, Qi Zhang, and Shao-Qun Zhang. TernaryCLIP: Efficiently compressing vision-language models with ternary weights and distilled knowledge. *arXiv preprint arXiv:2510.21879*, 2025. URL `https://arxiv.org/abs/2510.21879`.

[9] Yefei He, Zhenyu Lou, Luoming Zhang, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. BiViT: Extremely compressed binary vision transformers. In *ICCV 2023*, pages 5628–5640, 2023. doi: 10.1109/ICCV51070.2023.00520.

[10] Yanjing Li, Sheng Xu, Mingbao Lin, Xianbin Cao, Chuanjian Liu, Xiao Sun, and Baochang Zhang. Bi-ViT: Pushing the limit of vision transformer quantization. In *AAAI 2024*, volume 38, pages 3243–3251, 2024. doi: 10.1609/aaai.v38i4.28109.

[11] Phuoc-Hoan Charles Le and Xinlin Li. BinaryViT: Pushing binary vision transformers towards convolutional models. In *CVPR Workshops 2023*, pages 4665–4674, 2023. doi: 10.1109/CVPRW59228.2023.00492.

[12] Junrui Xiao, Zhikai Li, Jianquan Li, Lianwei Yang, and Qingyi Gu. BinaryViT: Toward efficient and accurate binary vision transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):195–206, 2025. doi: 10.1109/TCSVT.2024.3457610.

[13] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-ViT: Accurate and fully quantized low-bit vision transformer. In *NeurIPS 2022*, volume 35, pages 34451–34463, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/deb921bff461a7b0a5c344a4871e7101-Abstract-Conference.html.

[14] Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *ICML 2023*, volume 202, pages 21813–21824, 2023. URL https://arxiv.org/abs/2302.02210.

[15] Xijie Huang, Zhiqiang Shen, Pingcheng Dong, and Kwang-Ting Cheng. Quantization variation: A new perspective on training transformers with low-bit precision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2307.00331.

[16] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*, 2016. URL https://arxiv.org/abs/1602.02830.

[17] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. URL https://arxiv.org/abs/1606.06160.

[18] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *ICCV 2019*, pages 293–302, 2019. doi: 10.1109/ICCV.2019.00038.

[19] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *NeurIPS 2021*, volume 34, pages 28092–28103, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ec8956637a99787bd197eacd77acce5e-Abstract.html.

[20] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In *ECCV 2022*, pages 191–207, 2022. doi: 10.1007/978-3-031-19775-8_12.

[21] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. FQ-ViT: Post-training quantization for fully quantized vision transformer. In *IJCAI 2022*, pages 1173–1179, 2022. doi: 10.24963/ijcai.2022/164.

[22] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers. In *ICCV 2023*, pages 17181–17190, 2023. doi: 10.1109/ICCV51070.2023.01580.

[23] Lianwei Yang, Haisong Gong, Haokun Lin, Yichen Wu, Caifeng Shan, Zhenan Sun, and Qingyi Gu. DopQ-ViT: Towards distribution-friendly and outlier-aware post-training quantization for vision transformers. *arXiv preprint arXiv:2408.03291*, 2024. URL https://arxiv.org/abs/2408.03291.

[24] Navin Ranjan and Andreas Savakis. LRP-QViT: Mixed-precision vision transformer quantization via layer-wise relevance propagation. *arXiv preprint arXiv:2401.11243*, 2024. URL https://arxiv.org/abs/2401.11243.

[25] Yu-Shan Tai and An-Yeu Wu. AMP-ViT: Optimizing vision transformer efficiency with adaptive mixed-precision post-training quantization. In *WACV 2025*, pages 6828–6837, 2025. doi: 10.1109/WACV61041.2025.00664.

[26] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *CVPR 2019*, pages 11256–11264, 2019. doi: 10.1109/CVPR.2019.01152.

[27] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit LLMs: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024. URL https://arxiv.org/abs/2402.17764.

[28] Fengfu Li, Bin Liu, Xiaoxing Wang, Bo Zhang, and Junchi Yan. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016. URL https://arxiv.org/abs/1605.04711.

[29] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

[30] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *ICCV 2019*, pages 1325–1334, 2019. doi: 10.1109/ICCV.2019.00141.

[31] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. URL https://arxiv.org/abs/1806.08342.

[32] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. ReActNet: Towards precise binary neural network with generalized activation functions. In *ECCV 2020*, pages 143–159, 2020. doi: 10.1007/978-3-030-58568-6_9.

[33] Minsoo Kim, Sihwa Lee, Sukjin Hong, Du-Seong Chang, and Jungwook Choi. Understanding and improving knowledge distillation for quantization aware training of large transformer encoders. In *EMNLP 2022*, pages 6713–6725, 2022. doi: 10.18653/v1/2022.emnlp-main.450.

[34] Navin Ranjan and Andreas Savakis. Vision transformer quantization with multi-step knowledge distillation. In *Proc. SPIE 13057, Signal Processing, Sensor/Information Fusion, and Target Recognition XXXIII*, page 130570Q. SPIE, 2024. doi: 10.1117/12.3014158.

[35] Kaiqi Zhao and Ming Zhao. Self-supervised quantization-aware knowledge distillation. In *AISTATS 2024*, volume 238, pages 4375–4383, 2024. URL https://arxiv.org/abs/2403.11106.

[36] Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, Chenhan Yu, Wei-Chun Chen, Hayley Ross, Oluwatobi Olabiyi, Ashwath Aithal, Oleksii Kuchaiev, Daniel Korzekwa, Pavlo Molchanov, Mostofa Patwary, Mohammad Shoeybi, Jan Kautz, and Bryan Catanzaro. LLM pruning and distillation in practice: The Minitron approach. *arXiv preprint arXiv:2408.11796*, 2024. URL https://arxiv.org/abs/2408.11796.

[37] Meng Xin, Sweta Priyadarshi, Jingyu Xin, Bilal Kartal, Aditya Vavre, Asma Kuriparambil Thekkumpate, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Ido Shahaf, Akhiad Bercovich, Kinjal Patel, Suguna Varshini Velury, Chenjie Luo, Zhiyu Cheng, Jenny Chen, Chen-Han Yu, Wei Ping, Oleg Rybakov, Nima Tajbakhsh, Oluwatobi Olabiyi, Dusan Stosic, Di Wu, Song Han, Eric Chung, Sharath Turuvekere Sreenivas, Bryan Catanzaro, Yoshi Suhara, Tijmen Blankevoort, and Huizi Mao. Quantization-aware distillation for NVFP4 inference accuracy recovery. *arXiv preprint arXiv:2601.20088*, 2026. URL https://arxiv.org/abs/2601.20088.

[38] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. TinyViT: Fast pretraining distillation for small vision transformers. In *ECCV 2022*, pages 68–85, 2022. doi: 10.1007/978-3-031-19803-8_5.

[39] Seungwoo Son, Jegwang Ryu, Namhoon Lee, and Jaeho Lee. The role of masking for efficient supervised knowledge distillation of vision transformers. In *ECCV 2024*, pages 379–396, 2024. doi: 10.1007/978-3-031-72855-6_22.

[40] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *CVPR 2024*, pages 15731–15740, 2024. doi: 10.1109/CVPR52733.2024.01489.

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR 2019*, 2019. URL https://arxiv.org/abs/1711.05101.

[42] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *ICLR 2020*, 2020. URL https://arxiv.org/abs/1902.08153.

[43] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *ACL 2020*, pages 4190–4197, 2020. doi: 10.18653/v1/2020.acl-main.385.

[44] Yinan Liang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. MCUFormer: Deploying vision transformers on microcontrollers with limited memory. In *NeurIPS 2023*, volume 36, pages 8501–8512, 2023. URL https://arxiv.org/abs/2310.16898.

[45] Jianlei Yang, Jiacheng Liao, Fanding Lei, Meichen Liu, Lingkun Long, Junyi Chen, Han Wan, Bei Yu, and Weisheng Zhao. TinyFormer: Efficient sparse transformer design and deployment on tiny devices. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 73(5):3350–3362, 2026. doi: 10.1109/TCSI.2025.3638448.

[46] Christophe El Zeinaty, Wassim Hamidouche, Glenn Herrou, Daniel Ménard, and Mérouane Debbah. Can LLMs revolutionize the design of explainable and efficient TinyML models? In *IJCNN 2025*, pages 1–10, 2025. doi: 10.1109/IJCNN64981.2025.11229057.

[47] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *ICCV 2019*, pages 1314–1324, 2019. doi: 10.1109/ICCV.2019.00140.

[48] Danfeng Qin, Chas Leichner, Manolis Delakis, Marco Fornoni, Shixin Luo, Fan Yang, Weijun Wang, Colby Banbury, Chengxi Ye, Berkin Akin, Vaibhav Aggarwal, Tenghui Zhu, Daniele Moro, and Andrew Howard. MobileNetV4: Universal models for the mobile ecosystem. In *ECCV 2024*, pages 78–96, 2024. doi: 10.1007/978-3-031-73661-2_5.

[49] Xu Ouyang, Tao Ge, Thomas Hartvigsen, Zhisong Zhang, Haitao Mi, and Dong Yu. Low-bit quantization favors undertrained LLMs: Scaling laws for quantized LLMs with 100T training tokens. *arXiv preprint arXiv:2411.17691*, 2024. URL https://arxiv.org/abs/2411.17691.

[50] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021. URL https://arxiv.org/abs/2104.05704.

[51] Shashank Nag, Alan T. L. Bacellar, Zachary Susskind, Anshul Jha, Logan Liberty, Aishwarya Sivakumar, Eugene B. John, Krishnan Kailas, Priscila M. V. Lima, Neeraja J. Yadwadkar, Felipe M. G. França, and Lizy K. John. LL-ViT: Edge deployable vision transformers with look up table neurons. In *ICFPT 2025*, pages 19–29, 2025. doi: 10.1109/ICFPT67023.2025.00013.

[52] Zhikai Li and Qingyi Gu. I-ViT: Integer-only quantization for efficient vision transformer inference. In *ICCV 2023*, pages 17019–17029, 2023. doi: 10.1109/ICCV51070.2023.01565.

[53] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR 2018*, pages 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474.

A Experiments Appendix

A.1 Benchmark Results on CIFAR-10 and CIFAR-100

Table 6 reports DeiT-Tiny on CIFAR-10 and CIFAR-100. Our ternary model reaches 97.43% and 86.01% top-1. The FP32 DeiT-Tiny reaches 97.52% and 86.54%, so the gaps are 0.09 and 0.53 pp. Model size drops 15×, from 22.9 MB to 1.53 MB. FTerDeiT-Tiny is more accurate than every INT8 and ternary model in the table.

Table 6: Prior work comparison on CIFAR-10 and CIFAR-100.

Method	Params (M)	Size (MB)	Top-1 accuracy CIFAR-10	Top-1 accuracy CIFAR-100
Full-precision				
DeiT-Tiny (pretrained) [2]	5.5	22.9	97.52%	86.54%
CCT-7/3x1 [50]	3.76	15	96.5%	—
INT8 quantized				
TinyFormer-300K [45]	0.73	0.91	96.10%	—
LL-ViT [51]	2.5	1.93	95.5%	78.8%
I-ViT-T [52]	5.3	5.3	95.4%	79.2%
MobileNetV2 [53]	2.2	2.13	94.61%	—
LMaNet-Elite [46]	<1	<1	—	74.50%
Ternary				
ViT-1.58b [5]	307	57	72.3%	—
FTerDeiT-Tiny (Ours)	**5.5**	**1.53**	**97.43%**	**86.01%**

A.2 Isolation Controls on ImageNet-1K

Which part of the gain comes from the ternary operators, and which from distillation? We test this on FTerViT-DeiT-S^{224}. All numbers are top-1 on the ImageNet-1K validation set. Table 7 uses two protocols. A *full run* is the published two-phase recipe. An *adaptation* run starts from the released Phase 1 checkpoint, changes one factor, and fine-tunes for 20 epochs with the Phase 2 recipe. Adaptation runs are compared with a fully ternary model trained the same way (77.35 ± 0.04, 5 seeds), not with the published 77.47%.

Table 7: Isolation controls for FTerViT-DeiT-S^{224} on ImageNet-1K. Mean ± std over seeds where more than one seed was run. Δ is relative to the reference of the same protocol.

Control	Protocol	Top-1	Δ (pp)
Patch-embedding scaling granularity			
Per-channel (Ours), reference	adaptation, 5 seeds	$\mathbf{77.35 \pm 0.04\%}$	—
Per-tensor	adaptation, 5 seeds	$77.27 \pm 0.05\%$	−0.08
One component moved from ternary to INT8			
Patch embedding at INT8	adaptation, 1 seed	77.27%	−0.08
Classifier head at INT8	adaptation, 1 seed	77.53%	+0.18
LayerNorm affines at INT8	adaptation, 1 seed	76.77%	−0.58
All three at INT8 (6.33 MB)	full run	78.28%	+0.81
Fully ternary (Ours), 5.81 MB	full run	**77.47%**	—
Distillation in Phase 2			
KL to FP32 teacher (Ours)	Phase 2, 5 seeds	$\mathbf{77.45 \pm 0.07\%}$	—
Cross-entropy only, no teacher	Phase 2, 3 seeds	$76.89 \pm 0.08\%$	−0.56

Per-channel scaling helps by 0.08 pp over per-tensor scaling. This is within two seed standard deviations. No single component dominates. An INT8 patch embedding changes nothing beyond

seed noise. An INT8 classifier head gains 0.18 pp. INT8 LayerNorm affines lose 0.58 pp, the largest change. Training all three at INT8 from the start, with the same full recipe, gives 78.28% at 6.33 MB. Full ternarization therefore costs 0.81 pp and buys the last 9% of compression (Section 3.2). Removing the teacher from the 10-epoch Phase 2 restart costs 0.56 pp, about eight times the seed noise.

We ran the same ablations on DeiT-Tiny and CIFAR-100 with the full two-phase recipe. The seed spread there is σ=0.42 pp over 3 seeds. Distillation matters more in full training. Training without a teacher costs 3.17 pp. Keeping the teacher but using its hard prediction instead of its soft output costs 3.16 pp. We also isolate the cost of each component step by step. We start from an encoder-only ternary model and ternarize one more component at each step. Relative to the fully ternary model, an encoder-only ternary model is +0.63 pp. Adding a ternary classifier head gives +0.44 pp. Adding ternary LayerNorm affines gives −1.55 pp. Adding the ternary patch embedding gives the fully ternary model. Full ternarization thus costs 0.63 pp against the encoder-only model, less than a quarter of the distillation effect. The cost of ternary LayerNorm affines depends on the precision around them. Next to an INT8 patch embedding and head they are free (+0.07 pp). Next to an FP32 patch embedding they are the largest single step.

B On-Device Deployment and Profiling

B.1 Latency and Memory Results

Table 8: On-device inference profile for DeiT-III-S^{224} on ESP32-S3-EYE (dual Xtensa LX7 @ 240 MHz, 8 MB octal PSRAM, SIMD path)

Component	Time/block	12 blocks	% total
LayerNorm (pre + post)	38 ms	456 ms	2.2%
QKV projection (fused)	146 ms	1,752 ms	8.3%
Attention (Q@K^T+softmax+V)	551 ms	6,612 ms	31.4%
Output projection	70 ms	840 ms	4.0%
FFN (fc1 + fc2)	549 ms	6,588 ms	31.3%
Total (12 blocks)		**16,248 ms**	77.1%
Patch embed + head + other		4,814 ms	22.9%
End-to-end latency		**21,062 ms**	100%
Throughput		0.0475 inf/s	—
Memory & storage			
Flash (weights, ternary)		5.81 MB	—
Flash (firmware, code+rodata)		0.50 MB	—
Flash partition used / free		6.30 / 8.00 MB	79% used
Peak PSRAM (HWM, measured)		**2.83 MB**	—
of which scratch		1.89 MB	—
of which input image		0.57 MB	—
of which camera/LCD		≈ 0.37 MB	—
PSRAM heap available		7.35 MB	—
Internal SRAM min-free		93.9 KB	—
Compression vs FP32			
FP32 `.pth` reference		88.3 MB	—
Ternary flash footprint		5.81 MB	—
ImageNet-1K top-1 accuracy		79.64%	—

C Reproducibility

Each model trains in 1 to 2 days on a server with 8 NVIDIA L40S GPUs. Pretrained models and code are available at `https://huggingface.co/szymonrucinski/FTerViT` and `https://github.com/szymonrucinski/FTerViT`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state four contributions (fully ternary ViT, ternary modules for sensitive components, training efficiency via KD, deployment validation) with specific accuracy numbers and compression ratios, all supported by experiments in Sections 4–5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 (Conclusion) explicitly lists three limitations: restricted model scale (DeiT-Tiny and DeiT-III-S only), unoptimized MCU kernel and separate training stages.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results or proofs. All formulas (Eqs. 1–7) are definitions of quantization operations and loss functions, not theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 2 and 3 specify all ternary layer implementations (BitLinear, BitConv2d, TernaryLayerNorm) with equations. They list architectures, datasets, and resolutions. The loss function, hyperparameters and training recipe. The replacement ablation (Table 1) specifies all hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and model checkpoints will be released upon acceptance. All datasets used (CIFAR-10, CIFAR-100, ImageNet-1K) are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section Methodology and Experiments specifies models and datasets used for all benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main ImageNet-1K claims carry seed-replication error bars: a 5-seed Phase 2 replication gives 77.45 ± 0.07 for FTerViT-DeiT-S and 79.70 ± 0.08 for FTerViT-DeiT-III-S^{224} (mean $\pm$ std over seeds, Section 3.2), and the isolation controls in Appendix A.2 (Table 7) report mean $\pm$ std over 3–5 seeds. The factor of variability is the

random seed of the Phase 2 run. The Taylor-FO and Hessian importance shares (Table 2) report mean $\pm$ SEM over 16 mini-batches. The ESP32 latency is a deterministic single-device measurement.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports model specification,epoch counts ($\sim$260 for DeiT-III-S) and ESP32 inference time (21.06 s) and information about GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets (CIFAR-10/100, ImageNet-1K) and standard pretrained models. No human subjects, private data, or dual-use concerns are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational model compression research. The positive impact (enabling efficient ML on low-cost edge devices) is implicit. There is no direct path to negative applications beyond what any image classifier enables; ternary quantization does not introduce new risks compared to the FP32 models it compresses.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released artifacts (ternary ViT checkpoints for image classification) pose no meaningful misuse risk. They are small classifiers, not generative models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All pretrained models (DeiT-Tiny, DeiT-III-S) and datasets (CIFAR-10/100, ImageNet-1K) are explicitly mentioned. Section 5.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We open source our ternary models and code. Code and checkpoints will be shared with the community upon acceptance.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research is involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core methodology. LLM usage was limited to writing assistance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.