Supernet NAS for Hybrid TinyML CNN-Vision Transformers with Mixed Precision Quantization

1st Mikhael Djajapermana
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
mikhael.djajapermana@tum.de

2nd Daniel Mueller-Gritschneder
Institute of Computer Engineering
TU Wien
Vienna, Austria
daniel.mueller-gritschneder@tuwien.ac.at

3rd Ulf Schlichtmann
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
ulf.schlichtmann@tum.de

***Abstract*—TinyML aims to bring intelligence to resource-constrained edge devices, with computer vision being a key application. Conventional strategies such as manual design, pruning, or uniform quantization often yield suboptimal trade-offs or large accuracy losses under strict resource budgets. While Vision Transformers (ViTs) have shown promise in large-scale scenarios, their potential remains largely unexplored in the TinyML regime, where CNN-based networks continue to dominate. In this work, we present a Neural Architecture Search (NAS) framework that jointly optimizes architectures and Mixed-Precision Quantization (MPQ) policies for hybrid CNN–ViT models in a single supernet training run. We further incorporate feature-based knowledge distillation to reduce quantization noise during the MPQ-aware supernet training. After the supernet training, we can search for a number of smaller, quantized models without retraining. Unlike prior approaches that either treat NAS and MPQ separately or focus exclusively on either CNNs or ViTs, our framework co-designs quantized hybrid CNN–ViT models specifically for TinyML. Experiments on CIFAR10 and Tiny-ImageNet demonstrate that our method discovers CNN–ViTs with superior accuracy to state-of-the-art tiny CNNs and ViTs, achieving 63.89% on Tiny-ImageNet at 1 MB and 87.48% on CIFAR10 at 75 kB. Our results demonstrate the first sub-100kB CNN–ViT models with competitive performance and efficiency, highlighting their potential for extremely constrained edge devices.**

***Index Terms*—Neural Architecture Search, Mixed Precision Quantization, Convolutional Neural Networks, Vision Transformers, TinyML**

I. Introduction

The proliferation of IoT and edge devices has ignited interest in TinyML, which aims to embed intelligence directly into resource-constrained hardware. Computer vision is a key application, yet designing accurate vision models under severe memory and latency budgets remains a significant challenge. Traditional design methodologies, such as manual architecture design followed by pruning and post-training quantization, are too inflexible to derive optimized custom-tailored models. They rely heavily on expert intuition and frequently lead to unacceptable accuracy degradation when pushed to extreme compression. Neural Architecture Search (NAS) has emerged as a powerful paradigm for automating the design of efficient architectures, while quantization-aware training (QAT) reduces the memory footprint by lowering precision. Mixed-Precision Quantization (MPQ) is particularly attractive as it goes beyond uniform quantization, allowing different layers to adopt different bit-widths for a finer-grained accuracy-efficiency trade-off. Concurrently, Transformers [37], and especially the Vision Transformer (ViT) [10], have revolutionized visual recognition with their global self-attention mechanism, outperforming Convolutional Neural Networks (CNNs) on large-scale datasets. However, their massive parameter count and computational demand make them ill-suited for TinyML, where compact CNNs still dominate [2]. Hybrid CNN-ViT architectures offer a promising middle ground, combining the local inductive biases and efficiency of CNNs with the global contextual reasoning of ViTs. Recent works [26], [27] have demonstrated their potential on mobile devices, but their viability in the extremely constrained sub-100 kB regime remains almost entirely unexplored.

This work is motivated by two main research questions: (1) Can the global reasoning of ViTs provide measurable benefits in a hybrid CNN-ViT design even under a very tight parameter budget (< 100 kB)? (2) Can we develop a unified and automated framework that efficiently searches for the optimal joint architecture and MPQ policy specifically tailored for these tiny hybrid CNN-ViT models? We demonstrate that with careful co-design of supernet-based NAS and MPQ training and search strategies, hybrid CNN-ViT models can be realized under strict resource constraints with superior accuracy compared to the current state-of-the-art CNN and ViT models. Our contributions are as follows:

- We propose a hybrid CNN-ViT supernet search space specifically designed for generating tiny hybrid models (Section III-A). Our experiments reveal that the inclusion of self-attention provides consistent and measurable accuracy gains, even under a very limited size budget of 100 kB.
- We develop a joint search strategy based on supernet-based NAS for architectural choices with integrated Gumbel-Softmax sampling for MPQ. Our key contribution is the successful integration of these methods, enabling joint optimization of architectural parameters and

This work was partly funded by the German Federal Ministry of Research, Technology and Space (BMFTR) within MANNHEIM-FlexKI (16IS22086L) and DI-EDAI (16ME0991).

MPQ bit-widths within a single, weight-sharing supernet (Section III-B).
- To combat the severe accuracy loss from quantizing tiny models, we integrate a novel feature-based Knowledge Distillation (KD) strategy into the MPQ supernet training loop (Section III-C). Our distillation loss, combining MSE and cosine similarity, provides robustness against quantization noise by aligning the features of a quantized subnet with its full-precision counterpart.

We evaluate our method on CIFAR-10 and Tiny-ImageNet. Our method discovers CNN-ViT models that achieve superior accuracy compared to state-of-the-art small and tiny CNNs and ViTs, attaining 63.89% on Tiny-ImageNet under a 1 MB size constraint and 87.48% on CIFAR-10 under a 100 kB size constraint. To the best of our knowledge, this is the first work to demonstrate competitive-performing hybrid CNN-ViT models below 100 kB on a TinyML image classification benchmark. The code is available in https://github.com/mikhaeldj/tiny_nas/.

II. Related Works

A. Neural Architecture Search (NAS)

NAS is a powerful framework for automating the design of deep neural networks. Early work on NAS [33], [47], [48] used reinforcement learning or evolutionary algorithms, but was prohibitively expensive due to the need to train thousands of models from scratch. Differentiable NAS (DNAS) approaches [24], [38], [43] reduced this cost by relaxing the discrete search space to a continuous one, enabling gradient-based optimization. However, DNAS methods often suffer from optimization instability and a discrepancy between the continuous relaxation and the final discrete model, typically requiring the final discrete model to be retrained/fine-tuned. More recently, supernet-based NAS approaches [4], [13], [46] emerged as a more efficient alternative. In this paradigm, a single weight-sharing over-parameterized network (a *supernet*) is trained once, from which sub-networks (*subnets*) can be extracted and deployed without additional retraining/fine-tuning. BigNAS [46] introduced the sandwich sampling rule and in-place distillation to stabilize supernet training and achieve high-performing subnets for direct deployment. Building on top of BigNAS, AttentiveNAS [40] improved the sampling strategies, and AlphaNet [39] enhanced the distillation loss. However, these methods operate primarily in full precision and do not address the joint search for architecture and quantization policy, both of which are critical for TinyML. Our work addresses this gap by integrating MPQ into the supernet-based NAS paradigm to enable the discovery of high-quality TinyML vision models through joint architecture and MPQ search.

B. Mixed Precision Quantization (MPQ)

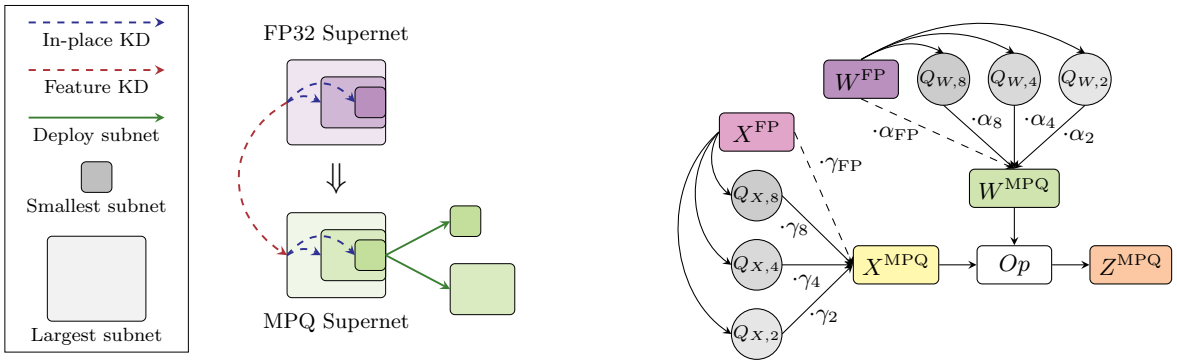
MPQ reduces model size and latency by assigning different bit-widths to different layers or tensors. Early methods, such as HAQ [41], used reinforcement learning to search for MPQ policies, while differentiable methods, such as MPQ-DNAS [42], employed gradient-based exploration of bit-width assignments. These methods, however, searched for MPQ policies on a *fixed* architecture. A more integrated line of work jointly searches the network architecture and its MPQ policy. SPOS [13] trained a MPQ-aware supernet with uniform path sampling to search for the bit-width assignment and channel size, but required the final discovered subnet to be retrained. QFA [1] removed the retraining requirement by proposing BatchQuant, a robust activation quantizer to stabilize the training of a MPQ supernet for joint architecture and MPQ search. However, most of MPQ research focuses on large or medium-scale networks, where parameter redundancy can absorb quantization noise. In contrast, tiny networks lack such redundancy and are more negatively affected by low-precision quantization [17]. Our work advances this line by enabling joint architecture and MPQ search specifically for the small- and tiny-model-size regime.

C. Knowledge Distillation (KD)

Knowledge Distillation (KD) improves the performance of a compact student model by transferring knowledge from a larger teacher network [15]. Techniques such as logit matching, feature distillation, and attention transfer have been explored [20]. In quantization, KD has been used to help low-precision students approximate their full-precision counterparts [19], [30].However, these works typically assume a static, pre-defined student architecture. In contrast, our work involves MPQ supernet training, whose dynamically changing architectures and precisions make optimization more fragile. We resolve this by incorporating feature-based KD directly into MPQ supernet training, improving robustness to quantization noise in tiny subnets.

D. Hybrid CNN-ViT

ViT [10] introduced self-attention as a powerful alternative to convolution in computer vision tasks. While highly scalable, pure ViTs often require large datasets and lack the inductive biases of CNNs, making them inefficient for small datasets and edge deployment. Subsequent efforts aimed to improve their data efficiency and reduce computational cost [25], [36]. SL-ViT [23] improved the performance of pure-ViTs on small datasets by increasing their local inductive bias through careful architectural modifications. Hybrid CNN-ViTs, such as LeViT [12] and MobileViT [26], [27], combine the local feature extraction of CNNs with the global reasoning of ViTs, achieving favorable accuracy-efficiency trade-offs on mobile devices. NAS has also been applied to design such hybrids [11], [35], [45]. Nevertheless, these works focused on mobile-scale models, with little exploration into the severely constrained TinyML regime. The work in [9] used NAS to design CNN-ViT models for TinyML vision tasks, but no prior works have considered *joint architecture and MPQ optimization of hybrid CNN-ViT models*. Our work brings MPQ CNN-ViT hybrids into the TinyML regime, showing that self-attention layers can still provide measurable benefits even under a strict



(a) MPQ-KD Supernet Training

(b) Soft Sampling for MPQ with {INT2, INT4, INT8}

Fig. 1. Overview of our proposed MPQ-KD supernet training strategy for joint architecture and MPQ search. The FP32 supernet guides MPQ subnets via feature distillation, while Gumbel-softmax soft sampling enables differentiable bit-width selection.

parameter budget when carefully integrated and designed using our unified NAS and MPQ framework.

III. CNN-ViT-MPQ Supernet Training

Our approach trains a hybrid CNN-ViT supernet using MPQ-aware training procedure based on Gumbel-Softmax sampling and feature-based KD from a full-precision teacher. Specifically, as illustrated in Fig. 1a, we train the MPQ supernet using the sandwich sampling rule [46], which involves simultaneously optimizing the largest, smallest, and random subnets with two distillation mechanisms: 1) in-place distillation [46] (blue dashed lines, Fig. 1a), where smaller subnets are trained to mimic the soft logits of the largest subnet, and 2) our proposed feature-based distillation (red dashed line, Fig. 1a), where the quantized (MPQ) largest subnet is guided by the features of the FP32 largest subnet. The MPQ optimization is implemented via a soft, differentiable sampling of quantization paths using Gumbel-Softmax (Fig. 1b). After the MPQ supernet training, joint architecture and MPQ search can be performed for any given constraint without additional subnet retraining. Our goal is to produce a well-trained CNN-ViT-MPQ supernet that enables the discovery of multiple highly accurate, quantized hybrid subnet candidates within a single training run.

A. Hybrid CNN-ViT Supernet Search Space

The supernet is an over-parameterized network that contains smaller, nested subnets. Our hybrid CNN-ViT supernet incorporates searchable CNN and ViT components within a sequential structure composed of three parts: a stem, a sequence of searchable stages, and a head. Our CNN-ViT supernet structure is illustrated in Fig. 2. The stem is a standard convolutional layer with batch normalization [16] and a ReLU activation [29]. The head consists of a global average pooling layer followed by a fully-connected linear classifier. The searchable stages are composed of CNN blocks for local feature extraction in the early stages, followed by ViT blocks for global reasoning in the later stages.

The CNN blocks follow the inverted residual structure of MobileNetV2 [31]. The block executes the sequence of 1×1 conv $\rightarrow$ $k \times k$ DW-conv $\rightarrow$ 1×1 conv. This structure is parameterized by three searchable parameters: 1) expansion ratio e controls the number of channels in the middle DW-conv by multiplying its input channels by e, 2) kernel size k controls the spatial size of the DW-conv filter, 3) output width c controls the output channel size of the last 1×1 conv. The searchable parameters can dynamically change during training. A residual connection is applied, with a 1×1 conv if input and output channels differ or if the stride is greater than 1.

The ViT blocks enhance representational capacity with Multi-Head Self-Attention (MHSA). Each ViT block comprises a 3×3 conv $\rightarrow$ 1×1 conv, followed by a MHSA layer and a Feed-Forward layer. The MHSA is performed directly on the $(B \times C \times H \times W)$[1] feature maps. For an input $X \in \mathbb{R}^{C \times H \times W}$, queries (Q), keys (K), and values (V) are generated via three separate 1×1 conv projections, W^Q, W^K, W^V, respectively:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{1}$$

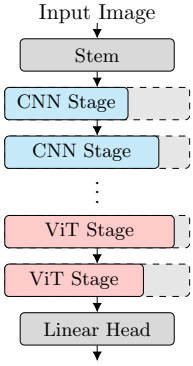
The MHSA aggregates the outputs of all attention heads,

$$\text{head}_i = \text{Attention}(Q, K, V) = \sigma\left(\frac{QK^T}{\sqrt{c_h}}\right), \tag{2}$$

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_{n_h})W^O, \tag{3}$$

where σ is the softmax function, n_h is the number of attention heads, c_h is the head dimension, and W^O is another 1×1 conv. The Feed-Forward layer consists of two 1×1 convs separated by a ReLU activation. The first convolution expands the channel count by a searchable factor e, while the second projects it to the searchable output width c. Residual connections are applied around both the MHSA and Feed-Forward layers. All normalization layers use batch normalization. Fig. 2 shows the overall structure and the searchable parameters of our CNN-ViT supernet. This hybrid design enables the CNN

[1] B = batch size, C = channel size, H = height, W = width



Stage	Type	Channel Size c			Depth d	Kernel Size k or Num;Dim Heads (n_h;c_h)	Expand Ratio e	Stride s
		min	max	step				
Stem	CNN	8	64	8	1	{3,5,7}	-	2
1	CNN	16	80	16	{1,2,3}	{3,5}	{2,4}	2
2	CNN	32	128	32	{0,1,2,3}	{3,5}	{2,4}	1
3	CNN	64	256	64	{0,1,2,3}	{3,5}	{2,4}	1
4	ViT	64	256	32	{1,2}	{4,6,8};{32,64}	{2,4}	2
5	ViT	64	512	64	{0,1,2}	{4,6,8};{32,64}	{3,6}	1
Head	Linear	# classes			1	-	-	-

Fig. 2. Hybrid CNN–ViT Supernet search space. The CNN stages handle local feature extraction, while the ViT stages capture global dependencies. Both CNN and ViT stages are dynamic and searchable.

stages to handle locality while the ViT stages capture global dependencies.

B. MPQ Supernet Training with Gumbel-Softmax Sampling

Our MPQ supernet training strategy is built upon the BigNAS sandwich rule [46], which optimizes the supernet by aggregating the losses of the largest subnet, the smallest subnet, and two randomly sampled subnets. We extend this rule to support MPQ-aware training of the supernet. Once the MPQ supernet is trained, multiple MPQ subnets can be searched and directly extracted without additional retraining/fine-tuning.

To enable differentiable optimization over discrete bit-width choices, we employ a Gumbel-Softmax [18] soft-sampling technique. Instead of hard-sampling a single precision per layer, soft-sampling calculates a weighted combination of all quantized paths (Fig. 1b). For a full-precision weight tensor W^{FP} and activation X^{FP}, their MPQ representations are:

$$W^{\text{MPQ}} = \sum_{b \in \mathcal{B}} Q_b(W^{\text{FP}}) \cdot \alpha_b, \tag{4}$$

$$X^{\text{MPQ}} = \sum_{b \in \mathcal{B}} Q_b(X^{\text{FP}}) \cdot \gamma_b, \tag{5}$$

where $\mathcal{B}$ is the set of available bit-width choices, $Q_b(\cdot)$ is the quantization function to b bits, and the probabilities α_b and γ_b are obtained via the Gumbel-softmax distribution. The output of a MPQ layer is then:

$$Z^{\text{MPQ}} = Op(X^{\text{MPQ}}, W^{\text{MPQ}}) \tag{6}$$

where Op is the layer's operation (e.g., convolution). This soft-sampling scheme creates a single, differentiable mixed-precision path per layer. While this soft-sampling scheme has been used for MPQ search on fixed architectures [5], [42], our key contribution is its application within a supernet with searchable architectural dimensions (e.g., widths, kernel sizes). We keep the Gumbel-softmax temperature high to ensure fair training across all nested architecture-MPQ paths within the supernet. The differentiable MPQ formulation of (6) and the sandwich training rule enable joint learning of both the architecture and the MPQ parameters within a single, weight-sharing supernet.

C. MPQ-aware Knowledge Distillation in Supernet Training

Quantization introduces noise that is particularly detrimental to the performance of tiny models. To mitigate this, we integrate our proposed feature-based KD into the MPQ supernet training loop, augmenting the existing in-place distillation from the sandwich training rule [46]. Our training consists of two phases: 1) In the warmup phase, the MPQ supernet is trained with the soft-sampling MPQ strategy, the sandwich rule, and in-place distillation. The available bit-width choices are {INT2, INT4, INT8, FP32}. We empirically find that including the FP32 option stabilizes early training. After this phase, the FP32 largest subnet is frozen to serve as a teacher for our feature-based distillation. 2) In the distillation phase, the available bit-widths are restricted to low-precision options {INT2, INT4, INT8}. Training continues with the soft-sampling MPQ strategy and the sandwich rule. Crucially, the MPQ largest subnet is now trained with a combined loss: a cross-entropy loss $\mathcal{L}_{\text{CE}}$ against the dataset labels and our proposed feature distillation loss $\mathcal{L}_{\text{KD-Feat}}$ that aligns its features with those of the FP32 largest subnet. The total loss for the MPQ largest subnet (S_{max}) is:

$$\mathcal{L}_{\text{total},S_{\text{max}}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{KD-Feat}}, \tag{7}$$

$$\mathcal{L}_{\text{CE}} = \text{Cross-Entropy}\left(\sigma(Z^{\text{MPQ}}_{S_{\text{max}}}), t\right), \tag{8}$$

$$\mathcal{L}_{\text{KD-Feat}} = \lambda \cdot \mathcal{L}_{\text{MSE}} + (1-\lambda) \cdot \mathcal{L}_{\cos} \tag{9}$$

The feature distillation loss $\mathcal{L}_{\text{KD-Feat}}$ is a weighted combination of an MSE loss to align feature magnitudes, and a cosine similarity loss to align feature directions:

$$\mathcal{L}_{\text{MSE}} = \text{MSE}\left(\hat{Z}^{\text{MPQ}}_{S_{\text{max}}}, \hat{Z}^{\text{FP}}_{S_{\text{max}}}\right), \tag{10}$$

$$\mathcal{L}_{\cos} = \left(1 - \cos\left(\hat{Z}^{\text{MPQ}}_{S_{\text{max}}}, \hat{Z}^{\text{FP}}_{S_{\text{max}}}\right)\right), \tag{11}$$

$$\hat{Z}^{\text{MPQ}}_{S_{\text{max}}} = \Psi\left(Z^{\text{MPQ}}_{S_{\text{max}}}\right), \quad \hat{Z}^{\text{FP}}_{S_{\text{max}}} = \Psi\left(Z^{\text{FP}}_{S_{\text{max}}}\right), \tag{12}$$

where λ is a parameter to balance the two terms, t is the ground-truth label, σ is the softmax function, $\hat{Z} = \Psi(Z)$ denotes L2-normalized logits, and $Z^{\text{FP}}_{S_{\text{max}}}$, $Z^{\text{MPQ}}_{S_{\text{max}}}$ denote the logits of the FP32 and MPQ subnets, respectively. The smaller MPQ subnets benefits from our enhanced feature-based guidance,

while they continue to be trained with in-place distillation to match the soft logits from the MPQ largest subnet. This two-phase approach ensures the MPQ subnets inherit robustness from the full-precision teacher, significantly reducing accuracy degradation. Importantly, the entire process operates within the supernet framework, such that discovered MPQ subnets require no further retraining/fine-tuning.

IV. Experiments

A. Experiment Setup

We evaluate our approach on two standard image classification benchmarks: CIFAR-10 [21] and Tiny-ImageNet [22]. These datasets represent low- and medium-scale scenarios relevant to TinyML applications. We apply data augmentation to increase the effective size and diversity of these datasets, using Auto Augment [7] and Repeated Augment [8]. We approximate the latency of a model architecture α by its number of BitOps$(\alpha) = \sum_\ell \text{MAC}_\ell(\alpha) * w_\ell(\alpha) * a_\ell(\alpha)$, where $\text{MAC}_\ell(\alpha)$, $w_\ell(\alpha)$, $a_\ell(\alpha)$ are the number of multiply-accumulates (MACs), the weight bit-width, and the activation bit-width of the l-th layer of the architecture α, respectively.

a) Training Details: We present the search space definition of our CNN-ViT supernet in Fig. 2. All architectural parameters are searchable except for stride, and a block with a chosen depth of $d = 0$ is treated as an identity function. We train the FP32 supernet for 500 epochs on CIFAR-10 and 350 epochs on Tiny-ImageNet, respectively. We use the SGD optimizer with momentum [32] and Cautious Weight Decay [6], and a cosine annealing learning rate schedule with an initial learning rate of 0.1 and 5 warmup epochs. Then, we apply our proposed MPQ-KD supernet training strategy (Section III-B), which runs for 200 epochs (50 warmup + 150 distillation) with the same optimizer and scheduler but a reduced initial learning rate of 0.01. Both weights and activations are quantized to mixed precision according to (4) and (5). We use LSQ+ [3] as the quantization function $Q(\cdot)$ and consider the bit-width choices $\{$INT2, INT4, INT8$\}$.

b) Search Details: After the MPQ-KD supernet training, we perform evolutionary searches to find subnets under different storage constraints. The search starts with 100 random candidates satisfying the constraint, preserves the top 20 in each generation, and produces 100 new generated candidates: 50 via mutation and 50 via crossover. This process is repeated for 30 iterations. Following [46], batch normalization statistics are recalibrated with 20 mini-batches, but no subnet retraining or fine-tuning is performed.

c) Baselines: We compare against models designed manually and those obtained via NAS-based approaches. For the manually designed models, we consider pure CNNs such as ResNets [2], [14], [28] and MobileNetV2 [31], pure ViTs such as SL-ViT [23], and hybrid CNN-ViTs such as MobileViT [26] and MobileViTV2 [27]. Their strides are adjusted such that the size of the final feature maps is 8×8. The FP32 version of these models is trained from scratch using the same training hyperparameters as our supernet. Then, MPQ-aware training is performed using Gumbel-Softmax sampling for 150 epochs. Since these models have fixed architecture, we only perform a MPQ search on them. We then report their INT8 and MPQ performances. For comparison against NAS-generated TinyML models, we consider Joint Pruning and Mixed-Precision (JPMP) ResNet models [28]. The JPMP-ResNet represents the current state-of-the-art mixed-precision tiny CNN-based networks. We obtain the JPMP-ResNet results using its public codebase, replicating the experiment setup described in [28].

B. State-of-the-Art Comparison

a) CIFAR-10 Results: The left-hand side of Table I summarizes the results on CIFAR-10. Under extreme compression, our best-performing CNN-ViT subnet achieves 87.48% accuracy at 73 kB size and 0.33 G BitOps, outperforming the current state-of-the-art tiny MPQ CNN, JPMP-ResNet8 with 86.32% at 75 kB and 0.35 G BitOps. Our CNN-ViT subnets continue to outperform MobileNetV2, SL-ViT, and MobileViT baselines at higher storage capacities. We demonstrate this by searching for CNN-ViT subnets under 2 MB and 1 MB, achieving 96.44% and 96.22% accuracy at 7.62 G BitOps and 6.72 G BitOps, respectively. These results indicate that the MHSA in our hybrid models provides a consistent and measurable accuracy boost, even under a size budget of less than 100 kB. This demonstrates that global reasoning capabilities are valuable not just in large models but are critical for achieving high accuracy under extreme compression. The improvements over CNN-only and ViT-only baselines validate the design of our hybrid supernet search space. Our joint CNN-ViT architecture and MPQ optimization approach achieves better trade-offs than the MPQ-trained baseline models with fixed architectures.

b) Tiny-ImageNet Results.: The right-hand side Table I summarizes the results on Tiny-ImageNet. On this more complex dataset, our approach continues to deliver competitive models under tight constraints. At 5 MB, our CNN-ViT subnet achieves 67.09% accuracy with 43.69 G BitOps, outperforming MobileViT and MobileViTv2 in terms of accuracy at a similar scale. Furthermore, our CNN-ViT substantially exceeds pure CNN and ViT baselines, such as MobileNetv2 and SL-ViT-8, which achieve only 64.38% and 61.10%, respectively. Under a 2 MB constraint, our CNN-ViT subnet maintains strong performance with 65.41% accuracy and 28.02 G BitOps, surpassing JPMP-ResNet18 in accuracy at 58.78%. While JPMP-ResNet18 consumes fewer BitOps, our CNN-ViT models consistently deliver higher accuracy with only a modest increase in BitOps, suggesting a flexible trade-off between accuracy and efficiency. Further scaling down to 1 MB size results in a CNN-ViT subnet with 63.89% accuracy and 12.29 G BitOps. These results further highlight that our hybrid CNN-ViTs generalize better than either pure CNNs or pure ViTs under resource-constrained conditions.

C. Ablation Study

To quantify the contribution of each component in our proposed MPQ-KD supernet training strategy (Section III-B

TABLE I
PERFORMANCE COMPARISON OF OUR CNN-VITS DISCOVERED BY OUR PROPOSED METHOD TO THE STATE-OF-THE-ART MODELS.

Models on CIFAR-10		Storage (MB) ↓	BitOps (G) ↓	Test Acc. (%) ↑
MobileNetV2 [31]*	INT8	2.43	9.19	94.62
	MPQ	0.98	5.09	94.56
SL-ViT-P4 [23]*	INT8	3.42	11.42	92.13
	MPQ	2.52	6.55	90.26
MobileViT-XS [26]*	INT8	1.93	27.08	92.28
	MPQ	1.27	17.45	91.58
MobileViTV2-0.75 [27]*	INT8	2.81	10.34	94.53
	MPQ	1.94	7.10	94.46
Ours-CNNViT	INT8	2.85	11.58	95.80
	MPQ	1.94	7.21	95.48
	MPQ	0.94	6.72	95.15
JPMP-ResNet8 [28]	INT8	0.081	0.81	87.05
	MPQ	0.075	0.35	86.32
Ours-CNNViT	INT8	0.089	0.73	87.72
	MPQ	0.073	0.33	87.48

Models on Tiny-ImageNet		Storage (MB) ↓	BitOps (G) ↓	Test Acc. (%) ↑
MobileNetV2 [31]*	INT8	2.91	9.88	64.38
	MPQ	2.16	6.49	62.63
SL-ViT-P8 [23]*	INT8	3.59	14.51	61.10
	MPQ	2.61	8.53	58.87
MobileViT-S [26]*	INT8	4.64	120.46	63.87
	MPQ	2.84	61.24	61.41
MobileViTv2-1.0 [27]*	INT8	4.90	73.14	65.91
	MPQ	2.94	48.22	63.04
JPMP-ResNet18 [28]	INT8	11.1	61.83	68.04
	MPQ	5.21	33.84	66.82
	MPQ	2.53	22.39	62.38
	MPQ	1.89	16.66	58.78
Ours-CNNViT	INT8	5.28	88.68	67.88
	MPQ	4.85	43.69	67.09
	MPQ	1.95	28.02	65.41
	MPQ	0.97	12.29	63.89

- The MPQ considers the bit-widths {INT2, INT4, INT8}.
- The symbol * indicates that the model has a fixed architecture.

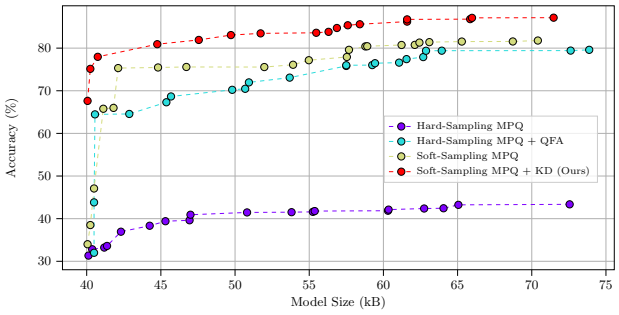


Fig. 3. Comparison between evolutionary search results on CIFAR10 from our CNN-ViT supernet trained with different MPQ-aware supernet training strategies.

and Section III-C), we conduct a series of ablation experiments on CIFAR-10 using our CNN-ViT supernet, focusing on the most challenging sub-100kB size regime. Without MPQ, the extracted subnets cannot meet these strict memory constraints. Therefore, we focus on ablating our techniques for training the MPQ supernet to maximize accuracy under extreme compression. We compare different MPQ training strategies, as shown in Fig. 3. The baseline MPQ supernet training is performed by hard-sampling a single-path quantized branch for each layer and performing the Lowest-Random-Highest Bits co-training, similar to the sandwich rule. This method was used in MultiQuant [44] and RFQuant [34], both produced great results on supernets with dynamic bit-width choices but *fixed* architectures. The hard-sampling method causes severe accuracy degradation (up to 5%) when applied to our CNN-ViT supernet with flexible architecture and MPQ configurations. This shows that the hard-sampling approach fails in this more complex search space. Incorporating the robust activation quantizer from QFA [1] into the hard-sampling method improves training stability, but yields subnets that still underperform. Then, we replace the hard-sampling strategy with our Gumbel-Softmax-based soft-sampling. The Gumbel-Softmax soft-sampling strategy recovers a significant portion of the accuracy loss, confirming that a differentiable approximation is crucial for stable joint architecture and MPQ optimization. Finally, adding our proposed feature-based KD provides an additional 1-2% accuracy gain. This demonstrates that our combined strategy of soft-sampling MPQ and feature-based KD produces the most stable and accurate quantized CNN-ViT subnets.

V. CONCLUSION

In this paper, we present a NAS framework that unifies MPQ and KD within a CNN–ViT supernet. Our method enables the joint optimization of architectures and MPQ policies in a single run, with the search specifically targeting the challenging TinyML regime. We demonstrate that hybrid CNN-ViT designs yield significant benefits even under extreme budgets (< 100 kB). On CIFAR-10 and Tiny-ImageNet, our approach discovers CNN-ViT models that achieve superior accuracy with competitive latency, outperforming state-of-the-art small and tiny CNNs and ViTs. Our findings highlight the potential of carefully co-designed hybrid CNN–ViT architectures and MPQ policies for deploying more capable vision models in edge AI and TinyML applications. Future work will explore extending this approach to additional tasks beyond classification and to more specialized hardware platforms.

REFERENCES

[1] Haoping Bai, Meng Cao, Ping Huang, and Jiulong Shan. Batchquant: Quantized-for-all architecture search with robust quantizer. *Advances in Neural Information Processing Systems*, 34:1074–1085, 2021.

[2] Colby Banbury, Vijay Janapa Reddi, Peter Torelli, Jeremy Holleman, Nat Jeffries, Csaba Kiraly, Pietro Montino, David Kanter, Sebastian Ahmed, Danilo Pau, et al. Mlperf tiny benchmark. *arXiv preprint arXiv:2106.07597*, 2021.

[3] Yash Bhalgat, Ananda Menon, Anup Bhattacharya, Sachin Bhandare, and Raghuraman Choudhury. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *CVPR Workshops*, 2020.

[4] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[5] Zhaowei Cai and Nuno Vasconcelos. Rethinking differentiable search for mixed-precision neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2020.

[6] Lizhang Chen, Jonathan Li, Kaizhao Liang, Baiyu Su, Cong Xie, Nuo Wang Pierse, Chen Liang, Ni Lao, and Qiang Liu. Cautious weight decay. *arXiv preprint arXiv:2510.12402*, 2025.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[9] Mikhael Djajapermana, Moritz Reiber, Daniel Mueller-Gritschneder, and Ulf Schlichtmann. Hybrid convolution and vision transformer nas search space for tinyml image classification. *arXiv preprint arXiv:2511.02992*, 2025.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, Qiang Liu, and Vikas Chandra. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022.

[12] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021.

[13] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part XVI 16*, pages 544–560. Springer, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[17] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[19] Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019.

[20] 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. *arXiv preprint arXiv:2211.11014*, 2022.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[23] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[26] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

[27] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022.

[28] Beatrice Alessandra Motetti, Matteo Risso, Alessio Burrello, Enrico Macii, Massimo Poncino, and Daniele Jahier Pagliari. Joint pruning and channel-wise mixed-precision quantization for efficient deep neural networks. *IEEE Transactions on Computers*, 2024.

[29] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[30] Cuong Pham, Tuan Hoang, and Thanh-Toan Do. Collaborative multi-teacher knowledge distillation for learning low bit-width deep neural networks. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 6435–6443, 2023.

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[32] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. pmlr, 2013.

[33] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2820–2828, 2019.

[34] Chen Tang, Yuan Meng, Jiacheng Jiang, Shuzhao Xie, Rongwei Lu, Xinzhu Ma, Zhi Wang, and Wenwu Zhu. Retraining-free model quantization via one-shot weight-coupling learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15855–15865, 2024.

[35] Chen Tang, Li Lyna Zhang, Huiqiang Jiang, Jiahang Xu, Ting Cao, Quanlu Zhang, Yuqing Yang, Zhi Wang, and Mao Yang. Elasticvit: Conflict-aware supernet training for deploying fast vision transformer on diverse mobile devices. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5829–5840, 2023.

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12965–12974, 2020.

[39] Dilin Wang, Chengyue Gong, Meng Li, Qiang Liu, and Vikas Chandra. Alphanet: Improved training of supernets with alpha-divergence. In *International Conference on Machine Learning*, pages 10760–10771. PMLR, 2021.

[40] Dilin Wang, Meng Li, Chengyue Gong, and Vikas Chandra. Attentivenas: Improving neural architecture search via attentive sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6418–6427, 2021.

[41] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8612–8620, 2019.

[42] Haoming Wu, Yonggan Wang, Song Han, and Yingyan Chen. Mixed precision quantization of convnets via differentiable neural architecture search. In *ICLR*, 2020.

[43] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018.

[44] Ke Xu, Qiantai Feng, Xingyi Zhang, and Dong Wang. Multiquant: Training once for multi-bit quantization of neural networks. In *IJCAI*, pages 3629–3635, 2022.

[45] Jianlei Yang, Jiacheng Liao, Fanding Lei, Meichen Liu, Junyi Chen, Lingkun Long, Han Wan, Bei Yu, and Weisheng Zhao. Tinyformer: Efficient transformer design and deployment on tiny devices. *arXiv preprint arXiv:2311.01759*, 2023.

[46] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 702–717. Springer, 2020.

[47] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[48] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.