BHViT: Binarized Hybrid Vision Transformer

Tian Gao[1,2] Yu Zhang[3] Zhiyuan Zhang[4] Huajun Liu[1] Kaijie Yin[2] Chengzhong Xu[2] Hui Kong[2, *]

[1]Nanjing University of Science and Technology   [2]University of Macau
[3]Shanghai Jiaotong University   [4]Singapore Management University

Abstract

Model binarization has made significant progress in enabling real-time and energy-efficient computation for convolutional neural networks (CNN), offering a potential solution to the deployment challenges faced by Vision Transformers (ViTs) on edge devices. However, due to the structural differences between CNN and Transformer architectures, simply applying binary CNN strategies to the ViT models will lead to a significant performance drop. To tackle this challenge, we propose BHViT, a binarization-friendly hybrid ViT architecture and its full binarization model with the guidance of three important observations. Initially, BHViT utilizes the local information interaction and hierarchical feature aggregation technique from coarse to fine levels to address redundant computations stemming from excessive tokens. Then, a novel module based on shift operations is proposed to enhance the performance of the binary Multi-Layer Perceptron (MLP) module without significantly increasing computational overhead. In addition, an innovative attention matrix binarization method based on quantization decomposition is proposed to evaluate the token's importance in the binarized attention matrix. Finally, we propose a regularization loss to address the inadequate optimization caused by the incompatibility between the weight oscillation in the binary layers and the Adam Optimizer. Extensive experimental results demonstrate that our proposed algorithm achieves SOTA performance among binary ViT methods. The source code is released at: https://github.com/IMRL/BHViT.

1. Introduction

In recent years, ViTs have made significant progress in many computer vision fields [8, 35]. Nevertheless, due to the substantial model size and high computational complexity, deploying ViT in real-time application scenarios with limited computing resources is challenging. To deal with this issue, model quantization methods [14, 17] have been proposed, among which binary ViT techniques can
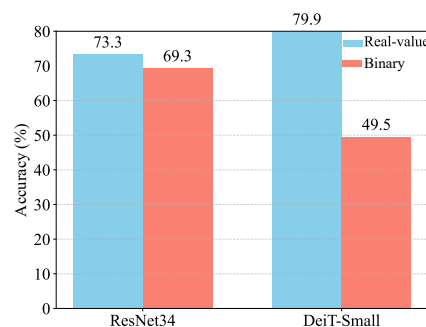
*Corresponding author

Figure 1. Classification accuracy on ImageNet-1K dataset for the binary method ReActNet [23], when applied in CNN and ViT architectures, respectively.

be the most efficient ones. Especially, the very recent Large Language Models (LLMs) [1] and Visual Language Models (VLM) [31] all adopt the transformer architecture. Therefore, the exploration of binary ViT with high performance holds practical significance. Although many binary CNN models have been preceding binary ViT, as shown in Fig. 1, directly applying existing binary CNN techniques (e.g., RSign and RPReLU [23]) to the ViT framework can not solve the significant performance degradation.

The primary reasons for the performance degradation lie in two aspects. First, the back-propagation of the attention module can be easily corrupted by the multiple clip functions and the non-differentiability of the sign operator, resulting in vanishing gradients for most elements in the activation. Second, the binary attention matrix cannot accurately represent the differences among the similarities of different tokens. The binarized attention can introduce a lot of noise and reduce the signal-to-noise ratio of the attention matrix [40], leading to a performance drop.

To tackle these challenges, we propose a hybrid ViT framework in this paper, which is better suited for binarization than the ViT and its full binarization version. Our contributions can be summarized as follows:

• We investigate the reasons for the severe performance degradation in the current binarized ViT models.

• Based on our research, we propose three novel modules to construct a high-performing Binarization-friendly
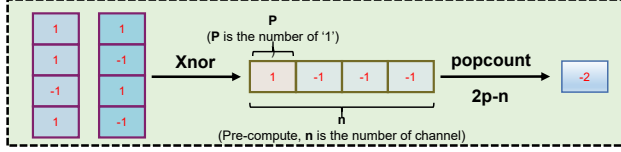
Figure 2. Multiplication of binary vectors implemented using $Xnor$ and $popcount$ operations, yielding the result $2p - n$

Hybrid ViT (BHViT) framework. Additionally, we propose a regularization loss to address incomplete optimization caused by the incompatibility between weight oscillation and the Adam Optimizer.

• To enhance the performance of the binary attention module, we propose a binarization scheme called Quantization Decomposition (QD) for attention matrices.

• We have applied the BHViT to both classification and segmentation tasks and achieved SOTA performance.

2. Related work

Binary Neural Networks (BNN). Model binarization initially begins with the CNN framework. Courbariaux et al. [2] proposed the BinaryConnect model with binary weights and full-precision activations. To address the issue of non-differentiable sign operators, they introduced the $clip$ function to approximate the gradient during the back-propagation process. Based on BinaryConnect, Hubara et al. [13] further extended the binarization to the model's activation, allowing the BNN to utilize $Xnor$ and $popcount$ operations to approximate matrix multiplication. Rastegari et al. [32] introduced the scaling factor for both weight and activation to enhance BNN performance on large datasets [4]. Lin et al. [20] introduced multiple binarization bases for weights and activations to approximate the corresponding full-precision values, but it introduces additional computations. Liu et al. [22] introduced residual links for each binary convolution layer to improve the representation capacity of the features. Furthermore, Liu et al. [23] incorporated an RPReLU activation function and the RSign operator within a MobileNet-based [34] architecture to reduce the performance gap between BNN and the corresponding full precision network.

Binary Vision Transformers (ViTs). While binary operations (e.g., RSign [23]) can be directly applied to ViTs, performance often degrades significantly. To address this, Li et al. [18] introduced the head-wise scaling operation to the attention module and utilized a ranking-aware distillation loss to train the binary ViT. Wang et al. [37] proposed channel-wise scale factors based on the Hamming distance between the binary Q and K tensors. Gao et al. [7] utilized distinct thresholds to transform the full-precision attention matrix and V matrix into a superposition form consisting of multiple binary bases. Other approaches relax bit-level constraints for specific network components. He

et al. [11] proposed a mixed-precision vision transformer, binarizing the attention module and MLP weights while keeping MLP activations in full precision. Similarly, Lu et al. [26] maintained full-precision attention matrices in their mixed-precision ViT.

In contrast to the above binary ViT methods, our work aims to mitigate the incompatibility between the ViT architecture and existing binarization techniques. Compared with BiReaL-Net [22], we first elucidate the significance of dense residual connections for binarizing the ViT architecture from an optimization perspective. Additionally, we slightly modify the original dense residual connections to accommodate the unique structural modules in Binary ViT.

3. Background

In the binary model, the weights and activations are constrained to two states, 1 and -1 (or 0 and 1). The multiplication of binary vectors is efficiently implemented using $Xnor$ and $popcount$ operations (Fig. 2). The linear layer is a fundamental component of the ViT that accomplishes the channel-wise aggregation of feature information. Its binarization process is described by:

$$\mathbf{Y} = \mathrm{B}_{\mathrm{Lin}}\left(\mathbf{X}, \mathbf{W}\right) = \alpha_{\mathbf{W}}\left(\hat{\mathbf{W}} \circledast \hat{\mathbf{X}}\right), \quad (1)$$

where $\mathbf{Y}$, $\mathbf{X}$, $\mathbf{W}$, $\hat{\mathbf{W}}$, and $\hat{\mathbf{X}}$ represent the output, input, weight, binarized weight, and binarized activation of the linear layer, respectively. $\alpha_{\mathbf{W}}$ is a scaling factor. The symbol $\circledast$ denotes binary matrix multiplication implemented by $Xnor$ and $popcount$ operations. To obtain $\hat{\mathbf{X}}$, as shown in Eq. 2, the $sign$ operator and the piecewise polynomial function [22] are applied to the forward and backward processes of the activation binarization, respectively.

$$\textbf{Forward } \hat{\mathbf{X}} = \mathrm{B}_a\left(\mathbf{X}, a, b\right) = sign\left(\frac{\mathbf{X} - b}{a}\right),$$

$$\textbf{Backward } \frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \hat{\mathbf{X}}}\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{X}} = \quad (2)$$

$$\begin{cases} \frac{\partial L}{\partial \hat{\mathbf{X}}} \cdot \left(2 + 2\left(\frac{\mathbf{X}-b}{a}\right)\right) & b - a \leqslant \mathbf{X} < b \\ \frac{\partial L}{\partial \hat{\mathbf{X}}} \cdot \left(2 - 2\left(\frac{\mathbf{X}-b}{a}\right)\right) & b \leqslant \mathbf{X} < b + a \text{ ,} \\ 0 & otherwise \end{cases}$$

where b and a represent the corresponding learnable bias and scale factor, respectively, and L is the loss function.

Due to the attention values (ranging from 0 to 1) significantly differ from the other activations, the attention is specifically binarized, as shown in Eq. 3.

$$\textbf{Forward } \hat{\mathbf{A}}_{tt} = \mathrm{B}_{att}\left(\mathbf{A}_{tt}, a, b\right) =$$

$$a \cdot clip\left(round\left(\frac{\mathbf{A}_{tt} - b}{a}\right), 0, 1\right), \quad (3)$$

$$\textbf{Backward } \frac{\partial L}{\partial \mathbf{A}_{tt}} = \begin{cases} a\frac{\partial L}{\partial \hat{\mathbf{A}}_{tt}} & b \leqslant \mathbf{A}_{tt} < a + b \\ 0 & otherwise \end{cases},$$

where B_{att} represents the binary function for the full precision attention matrix $\mathbf{A}_{tt}$, and $\hat{\mathbf{A}}_{tt}$ denotes the corresponding binary attention matrix. $clip\,(x, 0, 1)$ truncates values that fall below 0 to 0 and those above 1 to 1, effectively ensuring that the output remains within the range [0, 1]. $round$ operation maps the input to the nearest integer.

To binarize weights, we adopt the commonly used operations in Eq. 4.

$$\textbf{Forward } B_w\left(\mathbf{W}_{[:,k]}\right) = G\left(abs\left(\mathbf{W}_{[:,k]}\right)\right)$$
$$\cdot\, sign\left(\mathbf{W}_{[:,k]}\right),$$
$$\textbf{Backward } \frac{\partial L}{\partial \mathbf{W}_{[:,k]}} = G\left(abs\left(\mathbf{W}_{[:,k]}\right)\right) \quad (4)$$
$$\cdot\, \frac{\partial L}{\partial \hat{\mathbf{W}}_{[:,k]}} \cdot \mathbf{1}_{-1 < \mathbf{W}_{[:,k]} < 1},$$

where $\mathbf{W}_{[:,k]}$ represents the data in the k-th output channel of $\mathbf{W}$. $G\,()$ represents the average function to calculate the scaling factor. $\mathbf{1}_{-1 < \mathbf{W}_{[:,k]} < 1}$ denotes a mask tensor with the same size as $\mathbf{W}_{[:,k]}$. The element of the mask tensor is marked as one if the corresponding element in $\mathbf{W}_{[:,k]}$ falls in the closed interval [-1,1].

4. Method

4.1. Binarized Hybrid Vision Transformer

The architecture of our approach is depicted in Fig. 3, which comprises four stages in a feature pyramid with distinct feature sizes in spatial and channel dimensions in each stage. Given the input image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$, the patch embedding layer based on the convolution, as shown in Eq. 5, is applied to split and project image $\mathbf{I}$ to the feature sequence $X_0 \in \mathbb{R}^{64 \times \frac{H}{4} \times \frac{W}{4}}$, where H and W indicate horizontal and vertical size of image, respectively.

$$\mathbf{H}_0 = \text{GELU}\left(bn\left(Cov\left(\mathbf{I}\right)\right)\right), \mathbf{X}_0 = \mathbf{H}_0 + \mathbf{P}_e, \quad (5)$$

where bn means a batch-norm layer. GELU is a $gelu$ activation function. cov is a 4×4 convolution layer with a stride of 4. $\mathbf{P}_e$ means a learnable position embedding.

Then the embedding of $\mathbf{X}_{l-1}$ in each block as follow,

$$\mathbf{H}_{l-1} = \begin{cases} \text{MSGDC}\left(\mathbf{X}_{l-1}\right) + \mathbf{X}_{l-1}, if\ Stage \in [1,2] \\ \text{MSMHA}\left(\mathbf{X}_{l-1}\right) + \mathbf{X}_{l-1}, if\ Stage \in [3,4] \end{cases},$$
$$\mathbf{X}_l = \text{B\_MLP}\left(\mathbf{H}_{l-1}\right) + \mathbf{H}_{l-1}, \quad (6)$$

For simplification, we use "MSGDC" and "MSMHA" to represent the Binary Multi-Scale Grouped Dilated Convolution module and the Binary Multi-Scale Multi-Head Attention module, respectively.

The pyramid structure can extract multi-scale features and effectively enhance the representation ability of binary

features by increasing the channel dimension of the output feature. However, it also results in a large spatial resolution of features in the early stages of the model. It leads to a significant increase in the complexity of the attention module. To address this issue, we introduce MSGDC consisting of three 3×3 grouped atrous convolution layers (each layer employing a different dilated ratio) as the token mixer in the first two stages of the model. Meanwhile, we introduce an MSMHA module to accomplish token-wise feature fusion in the last two stages, ensuring effective information integration at different scales. A convolution layer with a stride of 2 and a 2×2 kernel size is applied in each down-sampling layer, which doubles the number of activation channels.

4.2. Token mixer

Multi-scale grouped dilated convolution (MSGDC) We apply three grouped convolutions with different dilation rates, enabling multi-scale feature fusion to enhance the representational capability of binary activation. Compared with ordinary convolutions and self-attention modules, grouped convolutions significantly reduce the model's parameters and computational complexity.

For the input feature $\mathbf{X}_{l-1} \in \mathbb{R}^{H \times W \times C}$, the embedding process of the MSGDC module is defined as

$$\mathbf{H}_{l-1}^n = \text{RPReLU}\left(B\_Cov_{3\times3,g}^{dil=2n-1}\left(B_a\left(\mathbf{X}_{l-1}\right)\right) + \mathbf{X}_{l-1}\right),$$
$$\mathbf{H}_{l-1} = bn\left(\mathbf{H}_{l-1}^1 + \mathbf{H}_{l-1}^2 + \mathbf{H}_{l-1}^3\right), \quad (7)$$

where $B_a\,()$ is the binary function defined in Eq. 2. $B\_Cov_{3\times3,g}^{dil=2n-1}$ is the binarized 3×3 grouped atrous convolution layer with dilated ratio of $2n-1$, $n \in (1, 2, 3)$. $\mathbf{H}_{l-1}^1$, $\mathbf{H}_{l-1}^2$, and $\mathbf{H}_{l-1}^3$ are the outputs of each grouped convolution layers, respectively. RPReLU is the activation function proposed by Liu et al. [23].

Multi-Scale Multi-Head Attention module (MSMHA) The original self-attention mechanism requires calculating the similarity between all tokens. However, when Q and K are binarized, the distribution of attention values exhibits a long-tail distribution with nearly 99% of attention values approaching 0 [7, 11], rendering most similarity calculations useless. To solve this problem, we introduce MSMHA, a variant of the window attention mechanism that maintains global information interaction and decreases computation costs. The motivation for applying the window attention mechanism is based on our observation that excessive numbers of tokens can degrade the performance of binary ViT, which is described in detail in **Observation** 1 (The detailed illustration is shown in the Appendix.).

Observation 1. *Avoiding excessive numbers of tokens is beneficial for Binary ViT.*
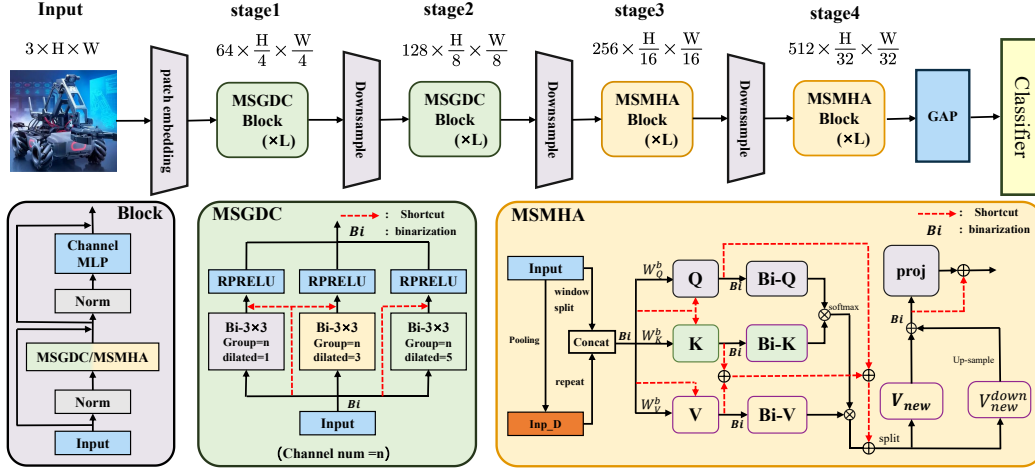
Figure 3. The architecture of the proposed binary hybrid ViT. "MSGDC" and "MSMHA" refer to the Binary Multi-Scale Grouped Dilated Convolution module and the Binary Multi-Scale Multi-Head Attention module, respectively. "GAP" stands for global average pooling. "Input_D" denotes the input tensor after downsampling. "RPRELU" is the activation layer proposed in [23].

In MSMHA, for the input feature $\mathbf{X}_{l-1} \in \mathbb{R}^{H \times W \times C}$(Fig. 4), we first apply average pooling with a kernel size of 7×7 to obtain high-scale feature $\mathbf{X}_{l-1}^{high} \in \mathbb{R}^{\frac{H}{7} \times \frac{W}{7} \times C}$. Meanwhile, we split the spatial resolution of the input feature to 7×7 and get the window version of the input feature $\mathbf{X}_{l-1}^{win} \in \mathbb{R}^{\frac{HW}{49} \times 7 \times 7 \times C}$. Subsequently, the hidden state feature $\mathbf{H} \in \mathbb{R}^{\frac{HW}{49} \times \left(49 + \frac{HW}{49}\right) \times C}$is obtained by concatenating $\mathbf{X}_{l-1}^{win}$(flattened to 1D vector) and $\mathbf{X}_{l-1}^{high}$(repeated and flattened). As shown in Eq. 8, $\mathbf{Q}_{l-1}$, $\mathbf{K}_{l-1}$, and $\mathbf{V}_{l-1}$ tensors are obtained by applying three binary linear layers to the hidden state feature $\mathbf{H}$.

$$\mathbf{Q}_{l-1} = \mathbf{B}_q\left(\mathbf{H}\right), \mathbf{K}_{l-1} = \mathbf{B}_k\left(\mathbf{H}\right), \mathbf{V}_{l-1} = \mathbf{B}_v\left(\mathbf{H}\right),$$
$$\mathbf{B}_q\left(\mathbf{X}\right) = \text{RPReLU}\left(bn\left(\text{B}_{\text{Lin}}\left(\mathbf{X}, W_q\right)\right)\right) + \mathbf{X},$$
(8)

where $\mathbf{B}_q\left(\right)$, $\mathbf{B}_k\left(\right)$, and $\mathbf{B}_v\left(\right)$ are the same operation with three different tensors of binary weight. B_{Lin} is a binary linear layer and W_q is the binary weight of linear layer in $\mathbf{B}_q$. Then the attention matrix $\mathbf{A}_{tt}$ is obtained by Eq. 9.

$$\mathbf{A}_{tt} = softmax\left(\frac{\mathbf{B}_a\left(\mathbf{Q}_{l-1}\right) * \mathbf{B}_a\left(\mathbf{K}_{l-1}\right)}{\sqrt{d}}\right). \quad (9)$$

Note that the binary attention matrix employs 0 and 1 instead of -1 and 1 as the binary states. Unlike the full-precision attention matrix, the binary version cannot assign different weights to each token based on their similarity, which is one of the primary factors resulting in the significant performance drop in binary ViT. To address this issue, we propose an effective binarization method called Quantization Decomposition (QD).

4.3. Quantization Decomposition

The softmax function ensures that each element of the attention matrix is less than 1. Therefore, we introduce a global
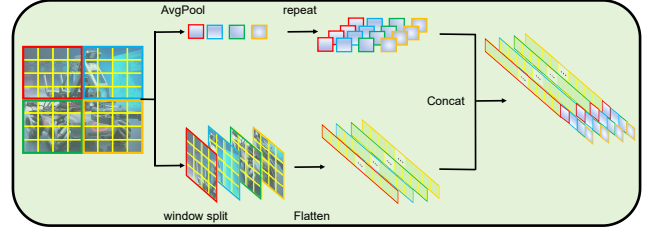
scaling constant $s = 2^n - 1$ (with $n = 2$). The binarization of the attention matrix is then achieved through:

$$\hat{\mathbf{A}}_{tt}^{\sigma} = \varphi\left(round\left(s\mathbf{A}_{tt}\right) \geqslant \sigma - 0.5\right), \sigma = (1, 2 \cdots, s),$$
(10)

where $\varphi\left(\right)$ is a boolean function. In this way, we obtain s binary attention matrices. If an element of $s\mathbf{A}_{tt}$ is larger than or equal to the constant $\sigma - 0.5$, the corresponding element of $\hat{\mathbf{A}}_{tt}^{\sigma}$ is set to 1. In contrast to the original binarization function, QD involves a decomposition process consisting of logical operations with relatively low computational complexity.

Meanwhile, we observe that applying additional shortcut in each MSMHA can improve the optimization of the binary linear layer for $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ tensors in the attention module for better performance, which is detailed in **Observation** 2 (The detailed illustration is shown in Appendix.). Different from the previous method [16, 22] aiming to enhance the representational capacity of binary activations through residual connections, our focus is on alleviating optimization issues caused by gradient mismatch and gradient vanishing. Therefore, we add shortcuts from full-precision $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ tensors to the output of the attention module directly, which improves the optimization process by decreas-

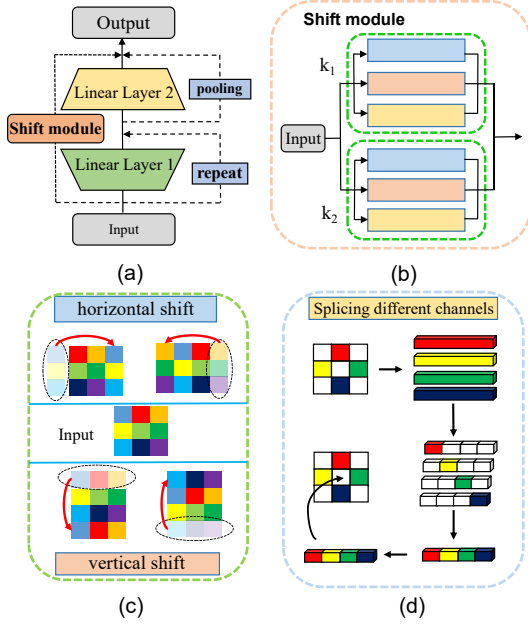Figure 4. Building hidden state feature for MSMHA.

Figure 5. Architecture of the binary MLP module. (a) Overall architecture. (b) The shift module, where colored rectangles represent horizontal, vertical, and mixed shift operations, with k1 and k2 denoting different translations. (c) and (d) illustrate specific shift operations.

ing the gradient mismatch between each activation in the attention module and the module's output.

Observation 2. *Adding a residual connection in each binary layer is beneficial for Binary ViT.*

After obtaining multiple binarized attention matrices, $\mathbf{V}_l$ can be obtained through Eq. 11.

$$\mathbf{V}_l = \left(\sum_{\sigma=1}^{s} \hat{\mathbf{A}}_{tt}^{\sigma} \circledast \mathbf{B}_a (\mathbf{V}_{l-1}) \right) + \mathbf{Q}_{l-1} + \mathbf{K}_{l-1} + \mathbf{V}_{l-1}, \tag{11}$$

According to the recorded splicing dimension information (the 'concatenation' process shown in Fig. 4) in obtaining the hidden state feature $\mathbf{H}$, $\mathbf{V}_l$ is split to the window-version feature $\mathbf{V}_l^{win}$ and the high-scale feature $\mathbf{V}_l^{high}$. Finally, the output feature $\mathbf{X}_l$ is obtained by Eq. 12.

$$\mathbf{X}_l = \vartheta \left(\mathbf{V}_l^{win} \right) + \Uparrow \left(mean \left(\mathbf{V}_l^{down} \right) \right), \tag{12}$$

where ϑ refers to reshaping the window-version feature to the original feature. $\Uparrow$ is the nearest neighbor interpolation to make the shape of $\mathbf{V}_l^{down}$ the same as $\mathbf{X}_l$. $mean(\mathbf{X})$ is a global average function along the dimension corresponding to the number of windows.

4.4. Binary MLP

The architecture of our proposed binary MLP module is depicted in Fig. 5 (a). To mitigate information loss and gradient errors, we introduce the shift operation, as shown in Fig. 5 (b), (c), and (d).

Given the input feature $\mathbf{H}_{l-1} \in \mathbb{R}^{H \times W \times C}$, the processing of the binary MLP is defined as

$$\mathbf{H}_{l-1}^{10} = bn \left(\mathrm{B}_{\mathrm{L}1} \left(\mathrm{B}_a \left(\mathbf{H}_{l-1} \right) \right) \right),$$
$$\mathbf{H}_{l-1}^{1} = \mathrm{RPReLU} \left(\mathbf{H}_{l-1}^{10} + repeat \left(\mathbf{H}_{l-1} \right) \right),$$
$$\mathbf{H}_{l-1}^{20} = bn \left(\mathrm{B}_{\mathrm{L}2} \left(\mathrm{B}_a \left(\mathbf{H}_{l-1}^{1} \right) \right) \right),$$
$$\mathbf{H}_{l-1}^{2} = \mathrm{RPReLU} \left(\mathbf{H}_{l-1}^{20} + pool \left(\mathbf{H}_{l-1}^{1} \right) \right), \tag{13}$$
$$\mathbf{S}_k = L_s \left(\mathbf{S}_k^{hor} \left(\mathbf{H}_{l-1} \right) \right) + L_s \left(\mathbf{S}_k^{ver} \left(\mathbf{H}_{l-1} \right) \right)$$
$$\quad + L_s \left(\mathbf{S}_k^{mix} \left(\mathbf{H}_{l-1} \right) \right), k \in (1, 2),$$
$$\mathbf{X}_l = \mathbf{H}_{l-1}^{2} + \mathbf{S}_1 + \mathbf{S}_2,$$

where $\mathrm{B}_{\mathrm{L}1}$ and $\mathrm{B}_{\mathrm{L}2}$ are two binary linear layers with a ratio of 4 and 0.25 between the number of input channels and output channels, respectively. L_s denotes a learnable channel-wise scaling transformation. The $repeat$ operation involves repeating the input four times and concatenating them along the channel dimension. The $pool$ function denotes 1D average pooling with a stride of 4. $\mathbf{S}_1$ and $\mathbf{S}_2$ represent two groups of shift operations with different strides. $\mathbf{S}^{hor}$, $\mathbf{S}^{ver}$, and $\mathbf{S}^{mix}$ are three fundamental shift operations defined in Fig. 5 (c) and (d), respectively.

For the horizontal shift operation shown in Fig. 5 (c), shifting the entire input feature map one pixel to the right is equivalent to moving the first column of the input feature map to the last column. Vertical shift operations can be implemented similarly. For the mix shift operation shown in Fig. 5 (d), we first identify four neighboring tokens of the current token. Then, in sequential order, we extract features from each adjacent token by taking one-fourth of the features along the channel dimension and concatenating them to replace the features of the current token.

4.5. Training Settings

Distillation Compared to the one hot label, the prediction provided by the teacher model for each class contains more information. Therefore, we utilize the DeiT-small [35] model as the teacher model during the training process to enhance the performance of the proposed binary model.

Compared with real-valued networks, weight oscillation occurs more frequently in the binary model [21, 28, 38]. In addition, we find that the commonly used Adam optimizer enlarges the weight oscillation of binary networks and results in some weights eventually ceasing updating, as described in Observation 3 (The detailed illustration is shown in Appendix.).

Observation 3. *The Adam optimizer enlarges the weight oscillation of binary networks in the later stages of the training process, failing to update numerous parameters effectively.*

To tackle this issue, we integrate L_1-regularization into the model's latent weights during the later phases of the model

training, thereby compelling some latent weights around 0 close to ±1. Then, the total loss function is shown in Eq. 14.

$$L = (1 - \lambda - \beta) L_{cls} + \lambda L_{dis} + \beta L_{re},$$

$$L_{re} = \frac{1}{n} \sum_{i=1}^{n} ||w_i| - 1|, \qquad (14)$$

$$\beta = \begin{cases} 0.1 & if \ T_{now} \geqslant 0.9 \times T_{max} \\ 0 & others \end{cases},$$

where L_{cls} and L_{dis} represent the cross-entropy loss between the class prediction and the corresponding label and the one between the output of the teacher network and the student networks' output, respectively. L_{re} refers to the L_1-regularization loss. To balance each component of L, we set the coefficient $\beta = 0.1$ (L_{re} is about 10 times of L_{cls} and L_{dis} in scale). Additionally, the model achieves the best performance when the hyperparameter λ is set to 0.8. T_{now} and T_{max} mean the current epoch index and the number of total epochs, respectively.

5. Experiments

5.1. Datasets and Implementation Details

Datasets. We evaluate the proposed method on four datasets, CIFAR-10 [15], ImageNet-1K [4], ADE20K [43] and the RS-LVF Dataset [41]. CIFAR-10 consists of 60,000 images with ten classes, where 50,000 images are used for training and 10,000 for testing. ImageNet-1K has 1,000 classes, with a training set of 1.2 million images and a test set of 50,000. ADE20K is a challenging dataset including more than 20000 images with 150 categories with a limited amount of training data per class. RS-LVF comprises 1000 aerial images with corresponding road labels in a Bird's-Eye View (BEV).

Implementation Details. We apply "RandomResized-Crop" and "RandomHorizontalFlip" operations to the ImageNet-1K dataset for data augmentation. For the CIFAR-10 dataset, we follow the data augmentation scheme proposed in DeiT [35]. The AdamW optimizer with cosine annealing learning-rate decay and an initial learning rate of 5×10^{-4} are applied to train the proposed method. The training process for the ImageNet-1K dataset is conducted using 4 NVIDIA A100 GPUs with a batch size of 512. The total epoch number for the ImagNet-1k dataset and the CIFAR-10 dataset are 150 and 300, respectively. For ADE20K and RS-LVF, the training batch sizes are 18 and 4, and the epochs are 50 and 100, respectively.

5.2. Classification

Results On CIFAR-10. We compare the classification accuracy of our method on the CIFAR-10 dataset with other BNN and binary ViT models. The result is shown in

Table 1. Classification results on CIFAR-10. W-A refers to the bit number of weights and activations for the corresponding binary method. NP denotes the number of network parameters ($\times 10^6$). The ‡ indicates that the attention matrix is fully precise.

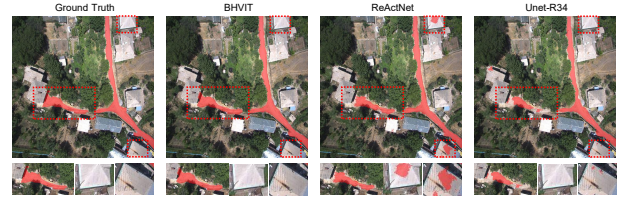| Architecture	Methods	W-A	NP	Top-1(%)
ResNet-18	Bi-RealNet [22]	1-1	11.2	89.12
	IR-Net [29]	1-1	11.2	91.20
	RBNN [19]	1-1	11.2	92.20
	XNOR-Net [32]	1-1	11.2	90.21
	RAD [6]	1-1	11.2	90.05
	Proxy-BNN [10]	1-1	11.2	91.80
	ReActNet [23]	1-1	11.2	92.31
	ReCU [39]	1-1	11.2	92.80
Mobile-Net	ReActNet-A [23]	1-1	28.3	82.95
DeiT-Small	ProxC++‡ [26]	1-1	21.6	86.19
	GSB [7]	1-1	21.6	91.20
BHViT	Ours-Tiny	1-1	13.2	93.30
	Ours-Small	1-1	22.1	**95.00**

Figure 6. Visualization on the RS-LVF dataset.

Tab. 1. Since the ViT model requires a large amount of training data [7] and lacks the inductive bias specific to visual tasks, the performance of binary ViT methods on small datasets is typically weaker than the BNN methods. Nevertheless, compared to BNN methods, our proposed BHViT-Tiny method achieves higher classification accuracy (0.5%) than the SOTA method ReCU [39]. For the methods based on ViT architecture, with the same level of the parameter's number, our proposed BHViT-Small outperforms ProxConnect++‡ [26] by nearly 8.81% and GSB [7] by 4.8% in terms of accuracy.

Results On ImageNet-1K. On the ImageNet-1K dataset, the classification performance of binary models based on different architectures is shown in Tab. 2. We find the methods with full-precision downsampling layers generally achieve good performance. Compared with ResNet-18 based methods, the performance of BHViT-tiny[†] is higher than the best method, ReActNet, by about 0.5% with similar OPs and model size. From Tab. 2, we can see the model's performance based on pure ViT architecture is not good enough. BHViT-Small[†] is 20.6 % higher than the current SOTA (ReActNet) with a similar OPs level. Compared with the Swin transformer architecture, BHViT-Small[†] exceeds the best method (BiViT) by 11.5 %. For the BinaryViT architecture, our method BHViT-Small still exceeds it by 0.7

% with the same setting. Compared with ReActNet-B∗, based on the MobileNet architecture, our BHViT-Small† achieves the same performance with a smaller model size and computational cost. In addition, the performance of ReActNet-A method on the datasets with relatively limited training data (e.g. CIFAR-10) is far inferior to the proposed algorithm. The proposed algorithm solves the problem shown in Fig. 1 and greatly surpasses current transformer-based algorithms in terms of performance, enabling the binary ViT architecture to outperform binary CNNs.

5.3. Road Segmentation and Image Segmentation

Road Segmentation. We evaluate the performance of the proposed model in road segmentation with the BEV perspective in aerial images. Road segmentation can be considered as a pixel-wise binary classification task. To assess the method's performance, we utilize the Mean Intersection over Union (mIOU) metric and mean accuracy metrics. The result is shown in Tab. 3, and some visual results for each method are demonstrated in Fig. 6. Each method is based on an encoder-decoder architecture, similar to U-Net [33]. The decoder module for each method is a ResNet structure with increasing resolution. In particular, different from ReAct-Net [23] and our method, the weight and activation of the ResNet-34 method are kept in full precision. From Tab. 3, we can observe that our proposed BHViT model exhibits higher segmentation accuracy than the CNN-based ReAct-Net, surpassing even full-precision methods.

Image Segmentation. We evaluate the performance of the proposed model in image segmentation with ADE20K [43] dataset. We apply two evaluation metrics, including pixel accuracy (pixAcc) and mean Intersection-over-Union (mIoU) to test each method. The result is shown in Tab 4, which demonstrates that our BHViT achieves SOTA performance among current binary segmentation algorithms.

5.4. Ablation Study

The impact of each proposed module. As shown in Tab. 5, the result displays the performance changes of the BHViT-Small model on the CIFAR-10 dataset by sequentially dropping the proposed modules, validating each proposed module's effectiveness. In the Tab. 5, 'FDL' denotes the down-sampling layer with full precision. 'RL' means the proposed regularization loss. 'Shift' is the proposed shift module shown in Fig. 5. In particular, the ablation of 'MSGDC' and 'MSMHA' represents whether it adds multi-scale information interaction in the corresponding module.

The impact of coefficient λ. Based on the CIFAR-10 dataset, we examine the influence introduced by the balance parameters λ on the classification performance of the BHViT-Small model. The result is shown in Tab. 6, from which we can conclude that the model achieves the highest accuracy when the hyperparameter λ is set to 0.8.

Table 2. Results On ImageNet-1K. † represents keeping the down-sampling layer at full precision and ∗ means applying the two stages training scheme proposed by [27]. OPs is defined as $OPs = \frac{BOPs}{64} + FLOPs$ [23]. BOPs and FLOPs mean binary and float operations, respectively.

Network	Methods	Size (MB)	OPs (×10⁸)	Top-1 (%)
ResNet-18 [9]	Real-valued	46.8	18.2	69.6
	BNNs† [3]			42.2
	XNOR-Net† [32]			51.2
	ABC-Net† [20]			42.7
	Bi-Real Net† [22]	2.1	1.6	56.4
	IR-Net† [29]			58.1
	RBNN† [19]			59.6
	ReCU† [39]			61.0
	ReActNet*† [23]			65.5
DeiT-Tiny [35]	Real-valued	22.8	12.3	72.2
	BiT [25]			21.7
	Bi-ViT [18]	1.0	0.6	28.7
	BiBERT [30]			5.9
	BVT-IMA [37]			30.03
Ours-Tiny	Real-valued	54.8	22.4	78.5
	BHViT-Tiny	2.5	0.5	64.0
	BHViT-Tiny†		1.1	66.0
ResNet-34 [9]	Real-valued	87.2	36.8	73.3
	Bi-Real Net†			62.2
	RBNN†	3.3	1.9	63.1
	ReCU†			65.1
MobileNet [12]	ReActNet-A*	3.7	0.87	69.4
	ReActNet-B*†		1.63	70.1
DeiT-Small [35]	Real-valued	88.4	45.5	79.9
	BiT			30.74
	BVT-IMA			47.98
	Bi-ViT	3.4	1.5	40.9
	BiBERT			17.4
	ReActNet			49.5
	Si-BiViT [42]	4.6	2.2	55.67
BinaryViT [16]	Real-valued	90.4	46.0	79.9
	BinaryViT	3.5	0.79	67.7
Swin-Tiny [24]	Real-valued	114.2	44.9	81.2
	BiBERT			34.0
	Bi-ViT	4.2	1.5	55.5
	BiViT [11]			58.6
	Si-BiViT	9.87	4.5	63.8
Ours-Small	Real-valued	90.4	45.0	79.3
	BHViT-Small	3.5	0.8	68.4
	BHViT-Small†		1.5	**70.1**

The impact of Regularization Loss (RL). We validate the effectiveness of the proposed regularization loss function from the perspective of weight distribution. The ablation of the RL on the CIFAR-10 dataset are depicted in Fig. 7 (The

Table 3. Comparison results of road segmentation in aerial view.

Encoder	Architecture	W-A	OPs(G)	mAcc	mIou
ResNet-34	U-Net	32-32	41.82	85.4	77.8
ReActNet	U-Net	1-1	4.87	76.5	63.6
Ours	U-Net	1-1	4.82	**92.2**	**85.1**

Table 4. Image segmentation results on ADE20K.

Method	Bit	OPs (G)	pixAcc (%)	mIoU (%)
BNN	1	4.84	61.69	8.68
ReActNet	1	4.98	62.77	9.22
AdaBin [36]	1	5.24	59.47	7.16
BiSRNet [5]	1	5.07	62.85	9.74
Ours	1	4.95	**65.63**	**14.87**

Table 5. Ablation study for BHViT-Small on the CIFAR-10.

Shift	MSGDC	MSMHA	QD	RL	FDL	Top1 (%)
✓	✓	✓	✓	✓	✓	95.0
✓	✓	✓	✓	✓		92.1
✓	✓	✓	✓			90.7
✓	✓	✓				88.9
✓	✓					86.7
✓						85.6
						83.2

Table 6. Ablation study for BHViT-Small on the CIFAR-10 dataset with different hyperparameters λ.

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Top1 (%)	93.6	93.8	94.1	94.2	93.9	94.5	94.8	95.0	94.7

Table 7. Classification results of BHViT with different token mixer settings. † means the downsampling layer is maintained at full precision.

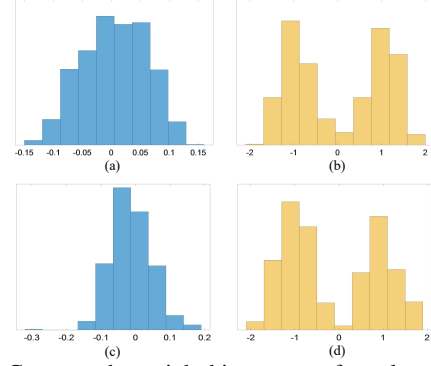| Model	Token mixer	Top1(%)
BHViT-Small†	Hybrid	70.1
BHViT-Small†	MSMHA	68.8
BHViT-Small†	MSGDC	67.2

Figure 7. Compare the weight histograms of two layers in different blocks. (a) and (c) denote the weight distribution of two layers without regularization loss. (b) and (d) denote the weight distribution of two layers with regularization loss.
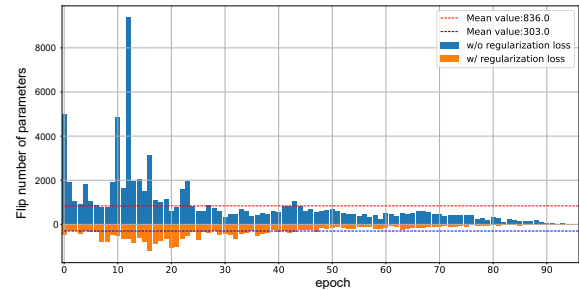

Figure 8. The number of flipped weights w/wo RL.

architecture. We validate the three different types of token mixers on ImageNet-1K, shown in Tab. 7. The pure ViT version achieves 1.3% lower classification accuracy than the hybrid version, while the pure CNN version performs the worst. The proposed hybrid architecture is a better choice for binarization.

6. Conclusions

In this paper, we propose a high-performance hybrid ViT framework and its binarized version, significantly reducing computational complexity while maintaining exceptional accuracy. Our work is based on three important observations, which guide us in introducing MSMHA, MSGDC, and MLP enhancement modules to improve the performance of binary ViT. Additionally, we propose the QD binarization method for the attention matrix and a regularization loss function to address weight oscillation issues in binary models when using the Adam optimizer. Experimental results show that the proposed method achieves SOTA performance among binarized models on benchmark datasets.

first convolution layer of block 1 and the Q linear layer of block 8, respectively), from which we observe that the RL change the distribution of latent weights, closer to +1 or -1, effectively mitigating weight oscillation. Meanwhile, at each training epoch, Fig. 8 shows the change in the number of flipped parameters with or without RL (the first convolution layer of block 1), validating the effectiveness of RL in solving weight oscillation.

The impact of different token mixer. We further studied the impact of different token mixer settings on model performance. The proposed MSMHA module can also reduce the number of tokens in the first two stages, i.e., applying MSMHA as the token mixer to all stages will turn the model into a pure ViT architecture. Alternatively, using MSGDC as the token mixer for each stage results in a pure CNN

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ah-mad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, 2015. 2

[3] 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. 7

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 6

[5] Lei Ding, Haitao Guo, Sicong Liu, Lichao Mou, Jing Zhang, and Lorenzo Bruzzone. Bi-temporal semantic reasoning for the semantic change detection in hr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–14, 2022. 8

[6] Ruizhou Ding, Ting-Wu Chin, Zeye Liu, and Diana Mar-culescu. Regularizing activation distribution for training bi-narized deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11408–11417, 2019. 6

[7] Tian Gao, Cheng-Zhong Xu, Le Zhang, and Hui Kong. Gsb: Group superposition binarization for vision transformer with limited training samples. *Neural Networks*, page 106133, 2024. 2, 3, 6

[8] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 1

[9] 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. 7

[10] Xiangyu He, Zitao Mo, Ke Cheng, Weixiang Xu, Qinghao Hu, Peisong Wang, Qingshan Liu, and Jian Cheng. Prox-ybnn: Learning binarized neural networks via proxy matri-ces. In *Computer Vision–ECCV 2020: 16th European Con-ference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 223–241. Springer, 2020. 6

[11] Yefei He, Zhenyu Lou, Luoming Zhang, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Bivit: Extremely com-pressed binary vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5651–5663, 2023. 2, 3, 7

[12] Andrew G Howard. Mobilenets: Efficient convolutional neu-ral networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 7

[13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Ad-vances in neural information processing systems*, 29, 2016. 2

[14] Youngbeom Jung, Hyeonuk Kim, Yeongjae Choi, and Lee-Sup Kim. Quantization-error-robust deep neural network for embedded accelerators. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(2):609–613, 2021. 1

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 6

[16] Phuoc-Hoan Charles Le and Xinlin Li. Binaryvit: Pushing binary vision transformers towards convolutional models. In *Proceedings of the IEEE/CVF Conference on Computer Vi-sion and Pattern Recognition workshop*, pages 4664–4673, 2023. 4, 7

[17] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. In *Advances in Neural Informa-tion Processing Systems*, 2022. 1

[18] Yanjing Li, Sheng Xu, Mingbao Lin, Xianbin Cao, Chuan-jian Liu, Xiao Sun, and Baochang Zhang. Bi-vit: Pushing the limit of vision transformer quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3243–3251, 2024. 2, 7

[19] Mingbao Lin, Rongrong Ji, Zihan Xu, Baochang Zhang, Yan Wang, Yongjian Wu, Feiyue Huang, and Chia-Wen Lin. Ro-tated binary neural network. In *Advances in Neural Informa-tion Processing Systems*, pages 7474–7485, 2020. 6, 7

[20] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. *Advances in neural in-formation processing systems*, 30, 2017. 2, 7

[21] Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *International Conference on Machine Learning*, pages 21813–21824. PMLR, 2023. 5

[22] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the perfor-mance of 1-bit cnns with improved representational capabil-ity and advanced training algorithm. In *Proceedings of the European Conference on Computer Vision*, pages 722–737, 2018. 2, 4, 6, 7

[23] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural net-work with generalized activation functions. In *Proceedings of the European Conference on Computer Vision*, pages 143–159. Springer, 2020. 1, 2, 3, 4, 6, 7

[24] 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. 7

[25] Zechun Liu, Barlas Oguz, Aasish Pappu, Lin Xiao, Scott Yih, Meng Li, Raghuraman Krishnamoorthi, and Yashar Mehdad. Bit: Robustly binarized multi-distilled trans-former. In *Advances in Neural Information Processing Sys-tems*, 2022. 7

[26] Yiwei Lu, Yaoliang Yu, Xinlin Li, and Vahid Partovi Nia. Understanding neural network binarization with forward and backward proximal quantizers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 6

[27] Brais Martinez, Jing Yang, Adrian Bulat, and Georgios Tzimiropoulos. Training binary neural networks with real-to-binary convolutions. In *International Conference on Learning Representations*, 2020. 7

[28] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330. PMLR, 2022. 5

[29] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2250–2259, 2020. 6, 7

[30] Haotong Qin, Yifu Ding, Mingyuan Zhang, Qinghua Yan, Aishan Liu, Qingqing Dang, Ziwei Liu, and Xianglong Liu. Bibert: Accurate fully binarized bert. In *International Conference on Learning Representations*, 2022. 7

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[32] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, pages 525–542. Springer, 2016. 2, 6, 7

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International of Medical image computing and computer-assisted intervention*, pages 234–241, 2015. 7

[34] 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. 2

[35] 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. 1, 5, 6, 7

[36] Zhijun Tu, Xinghao Chen, Pengju Ren, and Yunhe Wang. Adabin: Improving binary neural networks with adaptive binary sets. In *European conference on computer vision*, pages 379–395. Springer, 2022. 8

[37] Zhenyu Wang, Hao Luo, Xuemei Xie, Fan Wang, and Guangming Shi. Bvt-ima: Binary vision transformer with information-modified attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15761–15769, 2024. 2, 7

[38] Sheng Xu, Yanjing Li, Teli Ma, Mingbao Lin, Hao Dong, Baochang Zhang, Peng Gao, and Jinhu Lu. Resilient binary neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10620–10628, 2023. 5

[39] Zihan Xu, Mingbao Lin, Jianzhuang Liu, Jie Chen, Ling Shao, Yue Gao, Yonghong Tian, and Rongrong Ji. Recu: Reviving the dead weights in binary neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5198–5208, 2021. 6, 7

[40] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024. 1

[41] Kaijie Yin, Tian Gao, and Hui Kong. Pathfinder for low-altitude aircraft with binary neural network. *arXiv preprint arXiv:2409.08824*, 2024. 6

[42] Peng Yin, Xiaosu Zhu, Jingkuan Song, Lianli Gao, and Heng Tao Shen. Si-bivit: Binarizing vision transformers with spatial interaction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8169–8178, 2024. 7

[43] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 6, 7