LEARN APPROPRIATE PRECISE DISTRIBUTIONS FOR BINARY NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

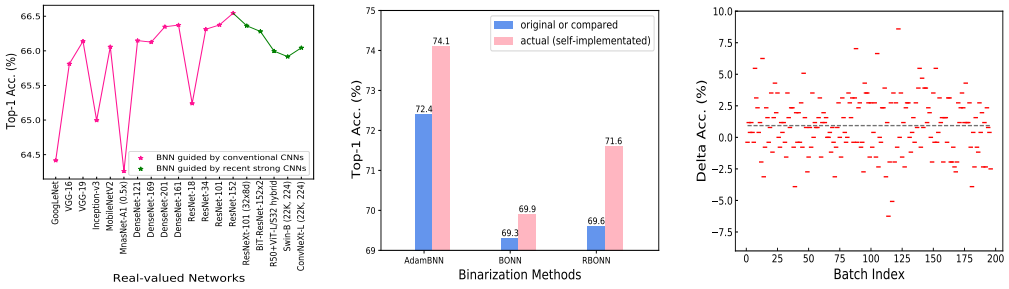
Binary Neural Networks (BNNs) have shown great promise for real-world embedded devices. However, BNNs always suffer from obtaining unsatisfactory accuracy performance on a large dataset such as ImageNet, which could hinder their further widespread applications in practice. Nevertheless, enhancing BNN's performance is extremely challenging owing to its limited capacity. Several distillation approaches in which the knowledge of a real-valued teacher model is distilled to a binary student network have been proposed to boost one BNN's accuracy. However, directly employing previous distillation solutions yields inferior results due to an unsuitable match between the representational capacity of the adopted real-valued teacher model and target binary student network. In this work, we re-examine the design of knowledge distillation framework specially for BNNs and test the limits of what a pure BNN can achieve. We firstly define one group which consists of multi real-valued networks owning particular properties, and then introduce a distribution-specific loss to enforce the binary network to mimic the distribution of one real-valued network fetched from this group in a certain order. In addition, we propose one distance-aware combinational model to provide one binary network with more comprehensive guidance, and present related suitable training strategies. The BNN in this built knowledge distillation framework can be facilitated to learn appropriate precise distributions, dubbed APD-BNN. As a result, APD-BNN can reach its performance limit while incurring no additional computational cost. Compared with the state-of-the-art BNNs, APD-BNN can obtain up to 1.4% higher accuracy on the ImageNet dataset with using the same architecture. Specifically, APD-BNN is capable of gaining 72.0% top-1 accuracy on ImageNet with only 87M OPs. Thus, it achieves the same accuracy of existing official real-valued MobileNetV2 at 71% fewer OPs, demonstrating the huge potential to apply BNNs in practice. Our code and models will be available.

1 INTRODUCTION

Binary neural networks (BNNs), in which both the weights and activations are restricted to 1-bit values, have shown enormous promise for real-world embedded devices (Xu et al., 2022). However, BNNs have always been criticized as they eventually gain unsatisfactory accuracy performance on large image classification dataset such as ImageNet (Russakovsky et al., 2015). The accuracies of BNNs to be applied in practice should perhaps be capable of at least achieving the level of MobileNet (Howard et al., 2017), because although some large-scale vision models can obtain impressive performance, practitioners typically use much smaller models in practice (Beyer et al., 2022), such as ResNet-50 (Kolesnikov et al., 2020) or MobileNet (Howard et al., 2017). However, the accuracies of pure BNNs[1] could always be much lower than MobileNetV2 (Sandler et al., 2018). Therefore, their deployment in practical scenarios is uncommon.

Recently, Real-to-Binary Net (Martinez et al., 2020) insisted that building a new baseline network probably should be the first step for gaining high accuracy for one BNN. Moreover, instead of adopting the ResNet-based architecture in Real-to-Binary Net, ReActNet (Liu et al., 2020) proposed

[1]Several previous works used real-valued convolution to enhance BNNs' accuracies. However, the operations of these BNNs could be increased by multiple times. Thus, in this paper, we only focus on the hardware-friendly BNNs which have pure 1-bit convolutions except the first and the last layers, dubbed pure BNNs.



(a) (b) (c)

Figure 1: (a) The top-1 accuracy of one BNN (ReCU-ResNet-34(Xu et al., 2021)) trained by various real-valued networks on ImageNet. (b) The potential accuracy improvements of the compared baseline real-valued networks ignored in latest BNN studies. Their adopted training strategies specially for BNNs can significantly boost the accuracies of corresponding real-valued networks based on our self-implementations (in which we totally follow the settings in their published papers). (c) The top-1 accuracy differences (Delta Acc) of ResNet-152 and ResNet-101 on mini-batches of ImageNet.

that the real-valued network design used as the starting point for binarization should be compact. Hence, these studies proved that the first step for binarization perhaps should be to design one appropriate baseline binary network, rather than directly implementing the binarization process.

Then, with a given BNN structure, various methods have been proposed for maximizing the potential in this BNN structure for better performance. Real-to-Binary Net and ReActNet proposed different kinds of distributional losses for enforcing one binary network to learn similar output distributions as those of a real-valued network. Based on the loss function in ReActNet, AdamBNN (Liu et al., 2021b) further focused on the investigation of optimizers and training strategy. These losses were all inspired by knowledge distillation (Hinton et al., 2015), in which the target binary network was as the student, while the adopted real-valued network served as one teacher. However, the single real-valued network respectively used in these works is randomly selected, such as ResNet-34 in both ReActNet and AdamBNN, and ResNet-18 in Real-to-Binary Net. Actually, the authors in Real-to-Binary Net suggested that utilizing a stronger teacher did not further improve the accuracy of one binary network, therefore they took a real-valued ResNet-18 model as a teacher.

However, contrary to Real-to-Binary Net (Martinez et al., 2020), we observe that the types and amounts of real-valued networks adopted for one binary network could be crucial for exploring the limits of what this binary network can achieve. The inappropriate type of the adopted real-valued network could lead to the suboptimality of the distribution similarity, thereby resulting in gaining inferior accuracy performance. For example, as depicted in Fig. 1(a), when training one particular binary network on ImageNet with the totally same settings except for employing different real-valued networks, the ultimate accuracy gap could be up to 2.7%.

Meanwhile, the training strategies proposed in AdamBNN particularly designed for one BNN could also improve the accuracy of its full-precision counterpart by 1.7%, as shown in Fig. 1(b). However, this improvement is neglected in AdamBNN. Besides, in BONN (Zhao et al., 2022), it was proved that their method can equally promote the performance of original real-valued ResNet-18 by 0.6% top-1 accuracy. However, the top-1 accuracy of their compared real-valued ResNet-18 baseline is the original existing official one 69.3%, thereby ignoring their gained 0.6% accuracy improvement. In addition, the new state-of-the-art RBONN (Xu et al., 2022) obtained 66.7% top-1 accuracy with using the 1-bit ReActNet-based ResNet-18. However, the possible 2.0% top-1 accuracy improvement for the original real-valued ResNet-18 baseline incurred by applying the structure improvements and the distillation loss in ReActNet (Liu et al., 2020) are also ignored, as described in Fig. 1(b). Hence, these studies including AdamBNN (Liu et al., 2021b), BONN (Zhao et al., 2022), and RBONN (Xu et al., 2022) just specially focus on boosting the accuracy of one BNN with a given structure, regardless of the possible accuracy enhancement of its original existing full-precision counterpart.

Thus, in this paper, we follow the previous studies (Liu et al., 2021b; Zhao et al., 2022; Xu et al., 2022) to specially maximize the potential in a given BNN structure for better performance. Cru-

cially, we strive to make BNN powerful enough to be able to achieve the accuracy level of MobileNetV2, with holding out hope that the practitioners can be convinced that utilizing BNNs in real-world applications is indeed feasible and promising. Specifically, unlike the simple and random distillation schemes in prior works, we aim for constructing one strong knowledge distillation system for enhancing BNNs. By utilizing this system, the limit of what a pure BNN can achieve is effectively explored. Our contributions can be summarized as:

- We firstly define one group which is comprised of multi real-valued networks owning particular properties, and then introduce a distribution-specific loss to enforce one binary network to learn the distribution of one real-valued network extracted from this group in a certain order.
- we further propose one distance-aware combinational model to provide one binary network with more comprehensive guidance, and present connected suitable training strategies.
- We are the first attempt to build one strong and effective knowledge distillation system for BNNs, which facilitates them to learn appropriate precise distributions (APD-BNN), with inducing no extra computational cost. Compared to state-of-the-art BNNs, APD-BNN can gain 1.4% higher accuracy on the ImageNet dataset with using the same architecture. Specifically, APD-BNN can reach 72.0% top-1 accuracy on ImageNet with only 87M OPs[2]. To the best of our knowledge, this is the first time that one pure BNN can achieve the same accuracy level of existing official real-valued MobileNetV2 (72.0%), demonstrating the huge potential of applying BNNs in practice.

2 Related Works

Although most previous studies can improve BNNs by adjusting network structures, their incurred additional computational cost offset the BNN's high compression advantage (Liu et al., 2021b). Thus, in this study, we are motivated to investigate the design of knowledge distillation framework specially for BNNs, which is orthogonal to the structure adjustment.

Knowledge distillation (Hinton et al., 2015) is a technique for transferring knowledge from one model (teacher) to another (student). The efficiency of distillation has been showed in several works including Real-to-Binary Net (Martinez et al., 2020) and ReActNet (Liu et al., 2020). Notably, the state-of-the-art RBONN (Xu et al., 2022) also uses the distillation loss to assist them in achieving the best performance on ImageNet when using the ReActNet-based networks. The main difference of our work to the similar works on knowledge distillation for BNNs, is that our observation is contrary to Real-to-Binary Net (Martinez et al., 2020) and we develop the first one strong and effective knowledge distillation system specially for BNNs. This system is capable of facilitating BNNs to achieve their performance limits, thus attaining new state-of-the-art results.

3 Method

This section introduces our approach. We build one strong knowledge distillation system for exploring the performance limit of one pure BNN with the aim of ultimately making it powerful enough even to be capable of achieving the accuracy level of MobileNetV2, in hope of demonstrating the tremendous potential to employ BNNs in practical applications.

3.1 Distribution-specific Loss

It has been proved that if the binary networks can learn similar distributions as real-valued networks, the performance can be enhanced (Martinez et al., 2020; Liu et al., 2020). Meanwhile, the authors in Real-to-Binary Net (Martinez et al., 2020) suggested that utilizing a stronger teacher did not further improve the accuracy of one binary network, therefore they took a real-valued ResNet-18 model as a teacher. Followingly, ReActNet (Liu et al., 2020) and AdamBNN (Liu et al., 2021b) both just simply utilized a real-valued ResNet-34 teacher model. However, as shown in Fig. 1(a), if one real-valued network has pretty low top-1 accuracy indicating that this network itself owns weak representational capacity such as AlexNet (Krizhevsky et al., 2012) and GoogleNet (Szegedy et al., 2015), then the binary network could learn a little valuable information from these real-valued

[2]OPs is a sum of binary OPs and floating-point OPs, i.e., OPs = BOPs/64 + FLOPs.

networks, thus achieving poor performance. On the other hand, the capacity of binary network is extremely limited resulting in a huge gap to the ones of outstanding real-valued networks, for example Swin-transformer (Liu et al., 2021a), so that the information offered by these real-valued networks could be too massive for the binary network to learn, thereby equally yielding inferior results. Accordingly, the suitable type of one real-valued network which is selected as the teacher model for one binary network could be critical.

There are plenty of networks, which can be simply separated based on distinguished families, such as VGG, ResNet, MobileNet, Vision Transformer, and etc. Meanwhile, it has been proved that a modified ResNet block must be used to obtain optimal results for binary networks (Martinez et al., 2020). Based on this fact, we further assume that the appropriate real-valued networks which will be utilized as the teacher models for one binary network should own ResNet blocks too. Therefore, the range of the candidate families of real-valued networks substantially shrinks. For efficiency, we merely consider two families of networks, including ResNet and DenseNet. As demonstrated in Fig. 1(a), one binary network can indeed gain higher accuracy when utilizing these two kinds of real-valued networks. Therefore, we followingly pack these networks into one group $\mathbb{G}$.

Besides, one network with larger number of layers in the same family could exhibit more excellent performance when targeting one image classification task. For example, ResNet-101 performs well than ResNet-50 in the ResNet family. Since the families of real-valued teacher models suitable for binary networks have already been identified as ResNet or DenseNet, we assume that one network with higher accuracy in its corresponding family could provide more useful guidance for one particular binary network. As described in Fig. 1(a), the binarized ReCU-ResNet-34 network (Xu et al., 2021) can gain 0.2% accuracy improvement when utilizing DenseNet-201 to replace DenseNet-121 as the teacher model, or 0.9% accuracy enhancement by replacing ResNet-18 with ResNet-34.

Hence, instead of randomly choosing one real-valued network as the teacher model for testing the performance limit of one binary network in prior studies (Liu et al., 2021b; Xu et al., 2022), one real-valued network Φ, which has the highest accuracy in the group $\mathbb{G}$, will be picked out as one teacher model in our method. Based on this fixed rule, we propose the distribution-specific loss $\mathcal{L}_\Phi$ for this fetched network Φ. It is defined as the KL divergence between the softmax output p_o of Φ and one target binary network t_b.

$$\mathcal{L}_\Phi = -\frac{1}{s}\sum_{k}\sum_{m=1}^{s} p_k^\Phi(x_m) log(\frac{p_k^{t_b}(x_m)}{p_k^\Phi(x_m)}) \tag{1}$$

where the subscript k represents the classes, and s is the batch size.

3.2 Distance-aware Combination Model

After identifying the group $\mathbb{G}$ consisting of multi candidate real-valued networks, we observe that even though two networks in $\mathbb{G}$ show similar accuracies on ImageNet, their performance on mini-batches is significantly different. As in Fig. 1(c), although ResNet-152 and ResNet-101 just have a accuracy gap of 0.934% on the whole ImageNet dataset, their accuracy differences on the same mini-batches range from -6.25% to 8.59%. Meanwhile, about 67.86% of these values are larger than 1.0% or less than -1.0%. These phenomena indicate that one network with higher accuracy on the whole dataset actually does not always perform excellently on every mini-batches. Accordingly, if ResNet-152 is firstly adopted as the real-valued model for training one binary network, and then ResNet-101 is utilized to replace ResNet-152 as the teacher model in another new training process, the guidance respectively provided by ResNet-152 and ResNet-101 in these two independent training processes may be complementary to each other on the majority of whole mini-batches.

Therefore, rather than merely utilizing one single real-valued network as the teacher model for exploring the performance limit of one binary network in previous studies (Liu et al., 2021b; Xu et al., 2022), we propose to combine multi real-valued networks in the group $\mathbb{G}$ to provide more comprehensive guidance for one binary network.

Based on our proposed distribution-specific loss in the section 3.1, one network with the highest accuracy, denoted as Φ_1, will firstly be taken out of the group $\mathbb{G}$. In this case, we will then pick out another one with highest accuracy in the remaining networks of the group $\mathbb{G}$, represented as Φ_2. In this way, there will be totally m networks, including Φ_1, Φ_2 and so on, to be extracted from the

group $\mathbb{G}$ and combined together as the unified teacher models for the target binary network t_b. Thus, we introduce the combinational loss $\mathcal{L}_c$ for providing t_b with comprehensive guidance:

$$\mathcal{L}_c = \sum_{i=1}^{m} \beta_i * \mathcal{L}_\Phi(\Phi_i) \tag{2}$$

where β_i is the weight to measure the importance of each network Φ_i.

Moreover, based on the fact that the first input convolutional layer and the output fully-connected layer in BNNs are not binarized in all previous binarization methods, we further seek for the opportunities of providing extra potential guidance for the target binary network t_b, for the sake of facilitating this binary network to reach its performance limit as far as possible.

Firstly, let us recall the goal of our distribution-specific loss $\mathcal{L}_\Phi$. It is to enforce the target binary network t_b to learn similar final output distribution of one real-valued network Φ_i. Based on this insight, there is one essential fact. These final output distributions out_d are not only the ones of the networks t_b and Φ_i, but also the outputs of the last layers Θ individually in t_b and Φ_i. Accordingly, if the final output distributions out_d of t_b and Φ_i are successfully enforced to be close to each other at last, then the input distributions in_d of Θ respectively in t_b and Φ_i should equally be similar to each other, since out_d are obtained just by directly processing in_d in Θ of t_b and Φ_i individually.

Hence, we present one distance model to measure the differences between in_d of Θ in t_b and Φ_i. By minimizing the output Dis of this distance model, extra potential enhanced guidance could be offered for the target binary network t_b. In this case, a desirable solution would be to calculate the KL divergence between two distributions in_d of Θ in Φ_i and t_b after a training iteration it, which are denoted as d_Φ^{it} and d_t^{it} respectively. We first calculate the per-input channel KL divergence between d_Φ^{it} and d_t^{it}. Then, the per-input channel KL divergence across all the input channels of the last layers Θ in the current training batch are averaged.

$$Dis = E^{it}[D_{KL}(d_\Phi^{it} \parallel d_t^{it})] \tag{3}$$

Followingly, since the computation of the KL divergence is expensive, we also simplify the computation of the KL divergence by adopting one effective second-order model in PROFIT (Park & Yoo, 2020) which considers the mean and variance of per-channel input distribution.

$$Y_{Dis} = \frac{1}{s} * \left(\log\left(st_\Phi / st_t\right) + \left(\left(st_t\right)^2 - \left(st_\Phi\right)^2 + \left(m_t - m_\Phi\right)^2\right) / \left(2 * \left(st_\Phi\right)^2\right)\right) \tag{4}$$

where st_Φ and m_Φ are the standard deviation and mean of per-channel input distribution of the last layer Θ in Φ_i, st_t and m_t are the ones in t_b, s is the batch size, and Y_{Dis} is the final output of our distance model.

Consequently, instead of utilizing the cross-entropy classification loss, we introduce the distance-aware combinational loss $\mathcal{L}_{dc}$ for one target binary network t_b.

$$\mathcal{L}_{dc} = \mathcal{L}_c + \alpha * Y_{Dis} \tag{5}$$

where α is a balancing parameter. Here, Y_{Dis} could be considered as some regularization on $\mathcal{L}_c$. In different situations, the binary networks may require varying degrees of the guidance supplied by Y_{Dis}, which is controlled by α. In some cases, if one binary network has already learned sufficiently from the real-valued network Φ_i via $\mathcal{L}_c$, then Y_{Dis} indeed makes no contributions and is unnecessary.

3.3 The Training Strategy

Actually, for the same one binary network, different studies tended to adopt distinguished training strategies. For example, when optimizing the same binary ReActNet-A network, ReActNet (Liu et al., 2020) trained it for 600K iterations with batch size being 256, while AdamBNN (Liu et al., 2021b) trained it for 600K iterations with batch size set to 512. Meanwhile, AdamBNN utilized another suitable weight decay value to maximize the potential in this given structure for better performance. Motivated by these studies, we equally design suitable training strategies for one binary network to be equipped with our distance-aware combinational model $\mathcal{L}_{dc}$.

Firstly, different from the small batch sizes including 256 in ReActNet and 512 in AdamBNN, we utilize the large batch size b_l in order to make full use of the system's computational power (You et al., 2017), such as 1696 for ReActNet-A network (Liu et al., 2020) on 8 NVIDIA A40 GPUs.

Meanwhile, FunMatch (Beyer et al., 2022) is the first work that presents the explicit identification of the certain implicit design choices of employing the knowledge distillation solution to obtain small-scale full-precision models with outstanding performance. They ultimately derive a state-of-the-art full-precision ResNet-50 model with 82.8% top-1 accuracy on ImageNet. And they also demonstrate that very long training schedule plays a key role in their proposed knowledge distillation scheme for obtaining this full-precision ResNet-50 model. In fact, it takes extremely expensive 3000K iterations to reach its gained 82.8% top-1 accuracy. Enlightened by FunMatch (Beyer et al., 2022), we assume that the suitable number of training iterations for one binary network is vital for approaching its performance limit. Besides, as originally reported in AdamBNN (Liu et al., 2021b), their binary network is trained for 600K iterations. Thus, to strike the balance between the training cost and final accuracy performance, the suitable total iterations for the binary network in AdamBNN (Liu et al., 2021b) to be trained by our distance-aware combinational model $\mathcal{L}_{dc}$ could be set to 416K.

In addition, the previous works (Liu et al., 2020; 2021b) both used the Adam optimizer (Kingma & Ba, 2014) with a linear learning rate decay scheduler. Directly combining the linear learning rate decay scheduler with large batch size b_l might not be optimal. If we just completely follow the training strategies in AdamBNN (Liu et al., 2021b) except for utilizing the large batch size b_l to replace the original small one, the final accuracy could suffer from a 0.5% degradation. Instead, we adopt the cosine annealing scheduler (Loshchilov & Hutter, 2016), and make further optimizations. Firstly, the initial learning rates i_a is slightly adjusted. The reason is that Adam adopts the adaptive method to update the gradients, which will amplify the actual learning rate values during training, so it requires a minor increment adjustment to avoid update values being too large. Meanwhile, the minimum learning rate m_f is fixed, which is set to 1e-9. Then, i_a and m_f together determine the range of the values of learning rate, and the specific value of leaning rate l_{it} in each iteration during one training process.

$$l_{it} = m_f + \frac{1}{2}(i_a - m_f)\big(1 + cos(\frac{it_{cur}}{it_{sum}^{b_l}}\pi)\big) \tag{6}$$

Where $it_{sum}^{b_l}$ is the sum of iterations based on large batch size b_l in one complete training process for one dataset, and it_{cur} accounts for how many iterations have been performed.

By integrating the distance-aware combinational model $\mathcal{L}_{dc}$ based on m particular real-valued networks fetched from our built group $\mathbb{G}$ and the above presented training strategies, one pure BNN can be effectively facilitated to learn appropriate precise distributions, dubbed as APD-BNN, while no additional computational cost is incurred.

4 Experiments

4.1 Dataset and Implementation Details

All our experiments are conducted on the ImageNet 2012 classification dataset (Russakovsky et al., 2015). We utilize the same data augmentation and pre-processing in AdaBin(Tu et al., 2022). All of the APD-BNN models follow the rule in previous methods (Liu et al., 2020; 2021b; Xu et al., 2022; Tu et al., 2022) that all layers, except the first input convolutional layer and the output fully-connected layer, are binarized. Meanwhile, all experiments are implemented using PyTorch (Paszke et al., 2019) with four or eight NVIDIA A40 GPUs, or eight NVIDIA A100-SXM4-80GB GPUs.

4.2 Comparison with State-of-the-Arts

Our solution brings constant improvements to various structures. As presented in Table 1, with the same network architecture, we achieve 1.7% higher accuracy than ReCU (Xu et al., 2021). In addition, based on the same ReActNet ResNet-based structure (Liu et al., 2020), we can gain 1.5% accuracy improvement. When applying our approach to AdamBNN (Liu et al., 2021b), it further brings 1.5% enhancement and obtains 72.0% top-1 accuracy, substantially surpassing all previous BNN models.

Meanwhile, our method will not increase the OPs since we utilize identical structures as the baselines: ReCU(Xu et al., 2021), ReActNet(Liu et al., 2020), and AdamBNN(Liu et al., 2021b). Table 2 describes the computational costs of the networks we used in experiments. Among all binary networks, ReActNet-A is the most promising one as it contains small overall OPs than other binary

Table 1: Comparison with state-of-the-art methods that binarize both weights and activations. ‡ means the ReCU-ResNet-34 (Xu et al., 2021) architecture, * denotes the ReActNet ResNet-based (Liu et al., 2020) structure, and † represents the ReActNet-A (Liu et al., 2020) structure.

Networks	Top1 Acc %	Top5 Acc %
BNNs (Courbariaux et al., 2016)	42.2	67.1
ABC-Net (Lin et al., 2017)	42.7	67.6
DoReFa-Net (Zhou et al., 2016)	43.6	-
XNOR-ResNet-18 (Rastegari et al., 2016)	51.2	69.3
Bi-RealNet-18 (Liu et al., 2018)	56.4	79.5
CI-BCNN-18 (Wang et al., 2019)	59.9	84.2
MoBiNet (Phan et al., 2020a)	54.4	77.5
BinarizeMobileNet (Phan et al., 2020b)	51.1	74.2
PCNN (Gu et al., 2019)	57.3	80.0
StrongBaseline (Martinez et al., 2020)	60.9	83.0
Real-to-Binary Net (Martinez et al., 2020)	65.4	86.2
MeliusNet29 (Bethge et al., 2020)	65.8	-
ReCU-ResNet-34‡ (Xu et al., 2021)	65.1	85.8
ReActNet ResNet-based* (Liu et al., 2020)	65.5	86.1
BONN* (Zhao et al., 2022)	66.2	86.4
RBONN* (Xu et al., 2022)	66.7	87.0
ReActNet-A† (Liu et al., 2020)	69.4	88.6
AdamBNN† (Liu et al., 2021b)	70.5	89.1
AdaBin† (Tu et al., 2022)	70.4	-
RBONN† (Xu et al., 2022)	70.6	89.0
APD-BNN‡ (ours)	**66.8**	**86.8**
APD-BNN* (ours)	**67.0**	**87.1**
APD-BNN† (ours)	**72.0**	**89.9**

networks. Based on this structure, AdamBNN (Liu et al., 2021b) proposed new training strategies to enhance its accuracy. Furthermore, the state-of-the-art works including RBONN (Xu et al., 2022) and AdaBin (Tu et al., 2022) both can equally boost the ReActNet-A network's top-1 accuracy. However, AdaBin (Tu et al., 2022) suffers from inducing extra OPs. In contrast, our APD-BNN can obtain the highest top-1 accuracy with incurring no additional OPs.

Thus, considering the tremendous challenges in previous attempts to enhance 1-bit CNNs' performance, the accuracy leap achieved by APD-BNN is significant. For example, our APD-BNN is capable of gaining 72.0% top-1 accuracy at 87M OPs. To the best of our knowledge, this is the first one pure BNN that can achieve the same accuracy level of MobileNetV2, demonstrating the enormous potential of employing BNNs in practical applications.

4.3 Ablation Study

4.3.1 The Number of Adopted Real-valued Networks

In Table 3, we illustrate the time consumption of one training iteration and accuracy performance of ReCU-ResNet-34 (Xu et al., 2021) on ImageNet with different numbers m of real-valued networks taken from our built group $\mathbb{G}$. Notably, when m is larger than three, the top-1 accuracy even decreases. In this case, the guidance supplied by extra real-valued networks could already exceed the limit of this binary network can grasp and in turn lead to an adverse effect. Overall, when m is equal to two, it strikes a balance between the accuracy performance and training cost.

4.3.2 The Families of Real-valued Networks

To reassure the credibility of building the group $\mathbb{G}$ for one binary network, we make a controlled comparison between different selections of other outstanding families of real-valued networks, in-

Table 2: Comparison of the computational cost between the state-of-the-art methods and our method. † represents the ReActNet-A (Liu et al., 2020) structure.

Networks	OPs $\times 10^9$	FLOPs $\times 10^8$	OPs $\times 10^8$	Top1 Acc %
XNOR-ResNet-18 (Rastegari et al., 2016)	1.70	1.41	1.67	51.2
Bi-RealNet-18 (Liu et al., 2018)	1.68	1.39	1.63	56.4
CI-BCNN-18 (Wang et al., 2019)	-	-	1.63	59.9
MeliusNet29 (Bethge et al., 2020)	5.47	1.29	2.14	65.8
StrongBaseline (Martinez et al., 2020)	1.68	1.54	1.80	60.9
Real-to-Binary Net (Martinez et al., 2020)	1.68	1.56	1.83	65.4
ReActNet-A† (Liu et al., 2020)	4.82	0.12	0.87	69.4
AdamBNN† (Liu et al., 2021b)	4.82	0.12	0.87	70.5
RBONN† (Xu et al., 2022)	-	-	0.87	70.6
AdaBin† (Tu et al., 2022)	-	-	0.88	70.4
APD-BNN† (ours)	4.82	0.12	**0.87**	**72.0**

Table 3: Comparison of different number of fetched real-valued networks from the group $\mathbb{G}$.

m	Time of one iteration (ms)	Top1 Acc %	Top5 Acc %
1	747	66.5	86.7
2	956	66.8	86.8
3	1279	**66.9**	86.8
4	1497	66.8	86.9
5	1674	66.7	**87.0**

cluding WRN (Zagoruyko & Komodakis, 2016), ResNeXt (Xie et al., 2017), BiT (Kolesnikov et al., 2020), Vision Transformer (ViT) (Dosovitskiy et al., 2020), Swin Transformer (ST) (Liu et al., 2021a), and ConvNeXt (Liu et al., 2022). As in Table 4, our suggested choice which utilizes ResNet-152 and ResNet-101 obtains a 0.7% better accuracy over the ConvNeXt, while the original accuracies of ConvNeXts are at least 8.5% higher than ResNet152 and ResNet-101. Among the whole validated real-valued network families, our suggested one itself has the lowest original accuracy, while its guided binary network gains the highest accuracy. It indicates a suitable match between their representational capacity, which could enable the binary network to learn sufficient valuable information from connected real-valued networks thereby achieving superior accuracy performance.

4.3.3 The Training Strategy

In this experiment, following AdamBNN (Liu et al., 2021b), we train the ReActNet-A network (Liu et al., 2020) by utilizing the two-step strategy. The activations are binarized first, and then the weights are binarized. As listed in Table 5, for fair comparison, we use the same model with 73.7% top-1 accuracy obtained in the first step as the initialization for the second step. The accuracy of ReActNet-A can be significantly boosted by adopting the appropriate batch size b_l and initial learning rate i_a. Crucially, as depicted in Fig. 2(a), there is no overfitting during training.

4.3.4 Comparisons to BNNs Containing Real-valued Convolutions

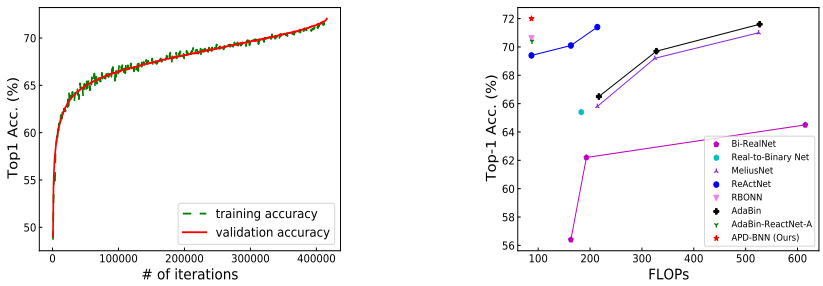
Some prior approaches utilize real-valued convolution to boost BNN's accuracy. However, the computation cost of one BNN is significantly increased. Hence, by adopting this strategy, the most state-of-the-art AdaBin (Tu et al., 2022) can obtain the 71.6% top-1 accuracy at 527M OPs, as in Fig. 2(b). In contrast, our APD-BNN has pure 1-bit convolutions except the first and the last layers, which is more hardware-friendly. Crucially, our APD-BNN is capable of achieving 72.0% top-1 accuracy on ImageNet with only 87M OPs, outperforming AdaBin (Tu et al., 2022) by 0.4% greater accuracy at 83% fewer OPs. These results demonstrate the effectiveness of our APD-BNN design.

Table 4: Comparison between our suggested choice ($i.e.$ ResNet) and other outstanding families of real-valued networks mostly built in timm (Wightman, 2019). We adopt the same data augmentation in ReCU (Xu et al., 2021) for fair comparisons. The target binary network t_b is ReCU-ResNet-34. For brevity, Top1 Acc (real) denotes the top-1 accuracies of related real-valued networks for t_b.

Real-valued networks		t_b
Model (family)	Top1 Acc (real)	Top1 Acc
resnet152 + resnet101 (ResNet)	78.2 + 77.3	**66.8**
wide_resnet101_2 + wide_resnet50_2 (WRN)	78.9 + 78.5	66.2
resnext101_32x8d + resnext50_32x4d (ResNeXt)	79.2 + 77.6	66.5
BiT-M-R152x4 + BiT-M-R101x3 (BiT)	83.8 + 83.0	66.0
vit_large_patch16_224 + vit_large_r50_s32_224 (ViT)	84.4 + 84.0	66.3
ig_resnext101_32x48d + ig_resnext101_32x32d (ResNeXt)	85.5 + 85.1	66.1
swin_large_patch4_window7_224 + swin_base_patch4_window7_224 (ST)	85.9 + 84.8	66.0
convnext_xlarge_in22ft1k + convnext_large_in22ft1k (ConvNeXt)	**86.7 + 86.3**	66.1

Table 5: Comparison of different hyper-parameter settings in the second step of two-step training on the ReActNet-A (Liu et al., 2020) structure.

Step1	Step2					
Top1 Acc %	m	α	Batch size	Initial learning rate	Top1 Acc %	Top5 Acc %
73.7	2	0.1	2816	0.00150	71.6	89.9
	2	0.1	1696	0.00125	71.6	89.8
	2	0.1	1696	0.00150	**72.0**	**89.9**
	2	0.1	1696	0.00175	71.7	89.9



(a) (b)

Figure 2: (a) The top-1 accuracy curves of our APD-BNN based on ReActNet-A (Liu et al., 2020) structure trained on ImageNet. (b) Computational Cost vs. ImageNet Accuracy.

5 Conclusion

Binary neural networks always suffer from obtaining unsatisfactory accuracy performance on the large scale image classification tasks, which could limit their widespread applications in practice. To tackle this issue, we build the first one strong and effective knowledge distillation system specially for one binary network with enforcing it to learn appropriate precise distributions (APD-BNN). APD-BNN can reach its performance limit while inducing no additional computational cost. We attain very strong empirical results. In particular, APD-BNN is capable of reaching 72.0% top-1 accuracy on ImageNet with only 87M OPs, which achieves the same accuracy level of existing official real-valued MobileNetV2 at 71% fewer OPs, demonstrating the huge potential of BNNs. We believe that they are very useful from a practical point of view and are a very strong baseline for future research on developing high-performance BNNs.

REFERENCES

Joseph Bethge, Christian Bartz, Haojin Yang, Ying Chen, and Christoph Meinel. Meliusnet: Can binary neural networks achieve mobilenet-level accuracy? *arXiv preprint arXiv:2001.05936*, 2020.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10925–10934, 2022.

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.

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.

Jiaxin Gu, Ce Li, Baochang Zhang, Jungong Han, Xianbin Cao, Jianzhuang Liu, and David Doermann. Projection convolutional neural networks for 1-bit cnns via discrete back propagation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8344–8351, 2019.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pp. 491–507. Springer, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. volume 25, 2012.

Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. *Advances in neural information processing systems*, 30, 2017.

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*, pp. 10012–10022, 2021a.

Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 722–737, 2018.

Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *European conference on computer vision*, pp. 143–159. Springer, 2020.

Zechun Liu, Zhiqiang Shen, Shichao Li, Koen Helwegen, Dong Huang, and Kwang-Ting Cheng. How do adam and training strategies help bnns optimization. In *International Conference on Machine Learning*, pp. 6936–6946. PMLR, 2021b.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

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.

Eunhyeok Park and Sungjoo Yoo. Profit: A novel training method for sub-4-bit mobilenet models. In *European Conference on Computer Vision*, pp. 430–446. Springer, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Hai Phan, Yihui He, Marios Savvides, Zhiqiang Shen, et al. Mobinet: A mobile binary network for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3453–3462, 2020a.

Hai Phan, Zechun Liu, Dang Huynh, Marios Savvides, Kwang-Ting Cheng, and Zhiqiang Shen. Binarizing mobilenet via evolution-based searching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13420–13429, 2020b.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pp. 525–542. Springer, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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*, pp. 4510–4520, 2018.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Zhijun Tu, Xinghao Chen, Pengju Ren, and Yunhe Wang. Adabin: Improving binary neural networks with adaptive binary sets. *arXiv preprint arXiv:2208.08084*, 2022.

Ziwei Wang, Jiwen Lu, Chenxin Tao, Jie Zhou, and Qi Tian. Learning channel-wise interactions for binary convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 568–577, 2019.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Sheng Xu, Yanjing Li, Tiancheng Wang, Teli Ma, Baochang Zhang, Peng Gao, Yu Qiao, Jinhu Lv, and Guodong Guo. Recurrent bilinear optimization for binary neural networks. *arXiv preprint arXiv:2209.01542*, 2022.

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*, pp. 5198–5208, 2021.

Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 6(12):6, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Junhe Zhao, Sheng Xu, Baochang Zhang, Jiaxin Gu, David Doermann, and Guodong Guo. Towards compact 1-bit cnns via bayesian learning. *International Journal of Computer Vision*, 130(2): 201–225, 2022.

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.