We sincerely thank four reviewers for the valuable comments. The training epoch issue is concerned by Reviewer #2, #3. The ablation study issue is concerned by Reviewer #2, #4. We answer these issues first.

Epochs. The LQ-Net trains 400 epochs on CIFAR-10 in total [r1] (the max_epoch parameter). We used 500 epochs to ensure the maximum accuracy is achieved. We found 400 (same with LQ-Net) epochs on the CIFAR-10 dataset could derive the same accuracies.

Ablation study. We have already conducted the ablation study on the temperature in Sec. 4.1 (Fig. 1) and the ablation study on SBN in Sec. 4.2.1 (Tab. 3). For the ablation of SLB w/o SBN on CIFAR-10, the accuracies of SLB-VGG-Small (W/A = {1/1, 2/2, 4/4}) w/o SBN on CIFAR-10 are 91.9% (-0.1%), 93.2% (-0.3%), and 93.6% (-0.2%). The temperature scheduler brings 4-6% accuracy improvement on CIFAR-10 . The temperature scheduler contributes more to the final performance compared to the SBN.

For Reviewer #1

Typos. Thanks. We fill fix the typos in the updated version and proofread the paper to make it more readible.

Distribution. As shown in Supp Fig. 1, the distribution becomes sharper during training. The distribution is not irreversible, and the gap between the soft distribution and hard representation becomes small.

Related works. The LWDNE (NeurIPS2019) introduces a novel optimizer for optimizing binary networks. Our SLB uses the SGD and Adam optimizer to optimize networks with several bit-widths. We will cite and discuss with LWDNE.

For Reviewer #2

Temperature. Thanks for the nice suggestions. We adopt these three schedulers in our experiments, because of their proven performance in a number of tasks, e.g., NAS (*e.g.*, exp scheduler in FBNet, CVPR2019), network optimization (*e.g.*, cos scheduler in DARTS, ICLR2019) and curriculum learning (Fig. 1 in [r2]). This paper aims for a new framework of searching low-bit weight. The designing and theoretical analysis of the schedulers are interesting but out of our scope. This will be nice follow ups to our current studies.

SGD-R. Thanks for this insightful comment. We tried the SGD-R scheduler to train VGG-Small (1/1) on the CIFAR-10 dataset. We warmup the temperature for three times. The accuracy is 92.3% which is 0.3% higher than the result trained w/o SGD-R. We will discuss this in the updated version.

The BN calibration. This step accelerates training by tuning BN for a few epochs. Calibrating the BN statistics after training could be efficient but would hurt the accuracy. However, the network at the last epoch is always not the best one. The second reason is that we could not observe the accuracies in the early/middle epochs because of the large quantization gap.

For Reviewer #3

GPU memory consumption. The memory consumption of VGG-Small trained on CIFAR-10 dataset is detailed in Tab. 1. The batch size is 100. In the case of 4-bit weight, only the weights require 16 times larger. During training, the GPU consumption on feature maps is much more than the GPU consumption on weights. The SLB-VGG-Small (4/4) uses around 30% extra memory compared to STE-VGG-Small. More GPU memory consumption is only required during training and our method do not increase the inference cost.

Table 1: The GPU consumption and time cost of VGG-Small on CIFAR-10.

Metric	Network	Dataset	STE	SLB(1/1)	SLB(2/2)	SLB(4/4)
GPU (G)	VGG-Small	CIFAR-10	3.1	3.1	3.2	4.0
Time (h)	VGG-Small	CIFAR-10	4.7	6.4	7.5	9.2
Time (h)	ResNet18	ImageNet	32	33	38	59

Low-bit set range. We follow DoReFa and set the value range to $[-1, 1]$. When the bit-wise is higher than 1, the accuracy might be higher because the information loss is small. For example, PACT [r3] makes the clipping value learnable and achieves higher accuracies.

Training time. We use 1 V100 to train networks on CIFAR-10 and 8 V100 to train networks on ImageNet. The time cost of VGG-Small on CIFAR-10 and ResNet18 on ImageNet is shown in Tab. 1.

Experimental details. We use Adam optimizer to train the architectures on the ImageNet dataset for 120 epochs in total with a batch size of 256, learning rate starts from 1e-3 and decays by a factor 10 at epoch 70, 90 and 110. Weight decay is 0. The temperature starts from 0.01, ends at 10.0, and changes by the exp scheduler.

CIFAR-10 error bar. We ran SLB-VGG-Small (W/A = {1/1, 2/2, 4/4}) (Tab. 2) for five times and the results are (mean$\pm$std format) 92.1$\pm$0.07, 93.5$\pm$0.08, and 93.8$\pm$0.05, respectively. The results are relatively steady.

Comparison with STE. The main problem of STE is that the gradients calculated relative to the quantized values are used to update the latent variables (Eq. 2). Our proposed method performs significantly better than STE-based methods on the condition that the differences between W_q and W_q^{lat} is large.

For Reviewer #4

Related works. Thanks for pointing out this issue. We will cite [r4] and discuss the differences between them. Admittedly, [r4] is an excellent work to explore differentiable quantization. However, the conventional sign function (Eq. 2 in [r4]) is still used to binarize weight parameters, which is still non-differentiable and Eq. 4 in [r4] can be seen as a kind of regularizer on the weight matrix W_l. In contrast, we utilize Eq. 6 in our main body to represent quantized weights during the training and do not use the sign function. For the binary case, each weight is represented by {+1, -1} at the same time, and both the two values and the probability P will be used to calculate the output features. By exploiting the proposed method, we can obtain state-of-the-art performance on benchmark network architectures on ImageNet using low-bit weights. Moreover, [r4] can only work for binary codes (see Eq. (4) in [r4]) and focus on hashing comparison experiments. Instead, ours is applicable for multiple bits and we mostly consider quantization comparison methods. This is exactly because we have a complete different algorithmic formulation.

Eq. 6. Eq. 6 is the expectation of the quantized values during training, while W_q in Eq. 7 is the finalized quantized values for inference. Given the example by #R4, we will take 2.2 for training but 1 for inference. Notably in fact, this gap will not be so large, as we will sharpen the distribution by adjusting the temperature during training (see our Theorem 1).

[1] https://github.com/microsoft/LQ-Nets/blob/master/cifar10-vgg-small.py#L151

[2] Dynamic Curriculum Learning for Imbalanced Data Classification. ICCV2019

[3] PACT: Parameterized Clipping Activation for Quantized Neural Networks. Arxiv

[4] Binarized Neural Networks for Resource-Efficient Hashing with Minimizing Quantization Loss. IJCAI2019