# K-SAM: Sharpness-Aware Minimization at the Speed pf SGD (Appendix)

## A SGD with Top-K and Random Subsampling

In this section, we show the testing accuracy on CIFAR-10 and CIFAR-100 for model ResNet-18 and WideResNet-28-10 trained with top-k and random subsampled SGD, where we only backpropagate through half of the examples with the largest losses or random half of the examples. We can see in Table 6, with top-k selection strategy, we can always obtain comparable performance to SGD backpropagating through the full mini-batch. However, when random sampling is applied, for most results, we will see clear accuracy drops.

Table 6: Using a subset to update the model with SGD on CIFAR-10 and CIFAR-100. We find that using top-k subsets yields similar performance to SGD, while using random-k subsets exhibits a clear performance drop.

| Model | Accuracy(%) | |
|---|---|---|
| ResNet-18 | CIFAR-10 | CIFAR-100 |
| SGD | $96.29 \pm 0.09$ | $79.08 \pm 0.18$ |
| top-64 | $96.26 \pm 0.14$ | $79.04 \pm 0.13$ |
| random-64 | $95.92 \pm 0.06$ | $78.53 \pm 0.20$ |
| Wide-28-10 | CIFAR-10 | CIFAR-100 |
| SGD | $96.99 \pm 0.09$ | $82.05 \pm 0.15$ |
| top-64 | $97.02 \pm 0.14$ | $82.29 \pm 0.22$ |
| random-64 | $96.66 \pm 0.09$ | $82.31 \pm 0.17$ |

## B Trade-off Between the Size of Subsets and K-SAM Performance on ImageNet

We provide results for different combinations of $K_1$ and $K_2$ for K-SAM on ImageNet with model ResNet-50. We observe a clear trade-offs between the size of subsets and the testing accuracy, where when we have smaller $K$, especially $K_2$, the performance on ImageNet will drop significantly. In addition, we show that this happens to SGD as well, where the accuracy drops more than $1\%$ when we only backpropagate through half of the mini-batches.

Table 7: K-SAM has a clear trade-offs on ImageNet between the size of the subsets and testing accuracy.

| $K_1/K_2$ | 256 | 360 | 512 |
|---|---|---|---|
| 64 | 75.37 | 76.43 | 76.55 |
| 128 | 75.99 | 76.81 | 76.73 |
| 256 | 76.08 | 77.09 | 77.29 |
| 512 | 76.34 | 76.90 | 77.19 |

Table 8: Performance drops significantly on ImageNet if $K_2$ is small when applied to vanilla SGD.

| $K_2$ | 256 | 360 | 512 |
|---|---|---|---|
| SGD | 74.866 | 75.516 | 75.978 |

## C PyTorch-Like Pseudo Code for K-SAM

The PyTorch-Like pseudo code is proved in Alg 2.

**Algorithm 2** PyTorch-like implementation for K-SAM

```
# Pytorch-like training loop

for inputs, targets in data_loader:

    with torch.no_grad():

        outputs = model(inputs)

        loss = criterion(outputs, targets)

        _, idx_k1 = torch.topk(loss, K_1) # Get M_{K_1}

        _, idx_k2 = torch.topk(loss, K_2) # Get M_{K_2}

    # Get the model perturbation

    outputs_k = model(inputs[idx_k1])
    loss_k1 = criterion(outputs_k, targets[idx_k1])
    loss_k1.mean().backward()
    optimizer.first_step()
    # Get the SAM updates

    outputs_k = model(inputs[idx_k2])
    loss_k2 = criterion(outputs_k, targets[idx_k2])
    loss_k2.mean().backward()
    optimizer.second_step()
```

# D Ablation Studies

**Different Subset Sizes** Contrary to ImageNet, K-SAM consistently outperforms SGD with different $K_1, K_2$ on CIFAR-10 and CIFAR-100. In addition, the best performance is usually achieved when $K_1 = K_2 = B/2$, where $B$ is the batch size, which is almost as good as or even better than the original SAM.

Table 9: Classification accuracy and total training time for K-SAM across combinations of $K_1, K_2$. Best accuracy is usually achieved when both of $K_1, K_2$ equal to half of the batch size.

| | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|
| ResNet-18 | Accuracy(%) | Time(h) | Accuracy(%) | Time(h) |
| SGD | 96.29 ± 0.09 | 1.07 | 79.08 ± 0.18 | 1.28 |
| K-128/64 | 96.59 ± 0.03 | 1.64 | 79.83 ± 0.16 | 1.77 |
| K-64/64 | 96.64 ± 0.04 | 1.36 | 79.75 ± 0.22 | 1.26 |
| K-64/128 | 96.63 ± 0.08 | 1.74 | 79.86 ± 0.28 | 1.84 |
| K-16/64 | 96.54 ± 0.14 | 1.13 | 79.24 ± 0.29 | 1.18 |
| Wide-28-10 | Accuracy(%) | Time(h) | Accuracy(%) | Time(h) |
| SGD | 96.91 ± 0.07 | 6.19 | 82.05 ± 0.15 | 5.96 |
| K-128/64 | 97.46 ± 0.10 | 10.84 | 84.39 ± 0.20 | 10.61 |
| K-64/64 | 97.46 ± 0.04 | 8.13 | 84.52 ± 0.15 | 8.36 |
| K-64/128 | 97.45 ± 0.05 | 10.64 | 84.17 ± 0.15 | 10.79 |
| K-16/64 | 97.45 ± 0.07 | 6.28 | 84.01 ± 0.29 | 6.21 |

**Effectiveness and Efficiency under distributed learning** In Table 10, we show that in the distributed setting, where vanilla SGD and SAM will be faster, we can still achieve the same efficiency improvements by K-SAM. In addition, K-SAM will achieve comparable results to SAM as well.

Table 10: Top-k SAM in the distributed larger batch setting. In this table, we observe that results in smaller batch setting extend to larger batch sizes and distributed computation. Experiments in this table use a batch size of 512 and $\rho = 0.02$ to train a WideResNet-16-8.

| | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|
| Wide-16-8 | Accuracy(%) | Time (minutes) | Accuracy(%) | Time(minutes) |
| SGD | $96.50 \pm 0.10$ | **47** | $80.01 \pm 0.09$ | 55 |
| SAM | $96.70 \pm 0.13$ | 93 | $80.35 \pm 0.23$ | 91 |
| K-64/512 | **96.79** $\pm 0.07$ | 62 | $80.17 \pm 0.28$ | 60 |
| K-64/384 | $96.53 \pm 0.10$ | 55 | **80.45** $\pm 0.28$ | 56 |
| K-64/256 | $96.28 \pm 0.12$ | **47** | $80.30 \pm 0.08$ | **46** |

# E Broader Impact

Our work focuses on improving the efficiency of deep network training while maintaining their powerful generalization ability. SAM has become an essential tool for achieving competitive performance on a variety of tasks, but its computational costs may be prohibitive for many practitioners. We seek to make SAM available to a broader audience. One limitation of our work is that on some datasets, such as ImageNet, we cannot achieve comparable performance to SAM without expending more resources than SGD. We hope that resource-limited practitioners will test K-SAM in their own setting to verify its benefits without assuming that it will consume no more resources than SGD while achieving equal performance to SAM.