Hyper-parameters	Value
sched	step
decay-epochs	2.4
decay-rate	0.97
opt	rmsproptf
b	192
epochs	450
opt-eps	0.001
j	8
warmup-lr	1e-6
weight-decay	1e-5
drop	0.3
drop-connect	0.2
model-ema	True
model-ema-decay	0.9999
aa	rand-m9-mstd0.5
remode	pixel
reprob	0.2
lr	0.06
amp	True
crop-pct	1.0

Table 5. The training hyper-parameters: we use Pytorch Image Models to train GPUNet, and here [4] further explains the usage of these hyper-parameters.

6. Supplemental Material

6.1. Training Receipts

Table. 5 shows the full details of training hyper-parameters. We used Pytorch Image Models in training, and we applied the same configurations to all GPUNet.

6.1.1 Sources of Baseline

The baseline models are from their original public release to ensure fair evaluations. We only convert their models to ONNX so that we can benchmark them in TensorRT. The conversion is invasive to the model latency and structure, and we use the Pytorch and Tensorflow native support for ONNX conversions. Here is the list that shows the source of the original implementation.

- FBNet:https : / / github . com / facebookresearch/mobile-vision
- EfficietNet-X:https : / / github . com / tensorflow / tpu / blob / master / models / official / efficientnet / tpu / efficientnet_x_builder.py
- EfficientNet:https : / / github . com / rwightman/pytorch-image-models
- RegNet:https : / / github . com / facebookresearch/pycls
- AlphaNet:https : / / github . com / facebookresearch/AlphaNet

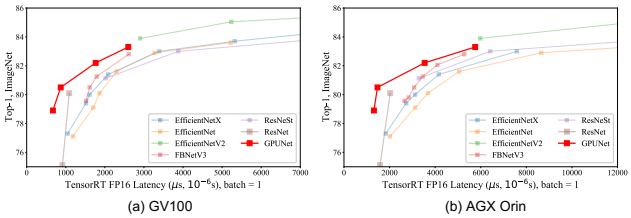

Figure 7. GPUNet performance on AGX Orin and GV100.

- ResNeSt:https : / / github . com / zhanghang1989/ResNeSt
- LaNet:https : / / github . com / facebookresearch/LaMCTS
- OFA:https://github.com/mit-han-lab/once-for-all

6.1.2 Verify the models on more devices

In Fig. 7, we have tested GPUNet optimized for GV100 on NVIDIA AGX Orin, and GPUNet consistently dominates other networks in the accuracy and latency Pareto frontier. GPUNet maintains the advantages because it replaces some memory-bound operators (e.g., high expansion ratio in SE layers) to compute bound operators, such as larger filters or deeper networks. An interesting observation is that the advantages of GPUNet-2 and GPUNet-3 (latency $>$ 3000 on Orin) decrease on Orin w.r.t on GV100. Because GV100 has more execution units than Orin, increasing filters or layers can better saturate the device. Therefore, the optimization strategies generalized from GPUNet in sec.4.2.4 are still applicable to other devices.