We believe our findings will open up new opportunities for on-device learning. We thank all reviewers for their comments and thank R2 and R3 for suggesting the literature. We will revise our paper weights from the super-net, all sub-nets are adapted to the target dataset while keeping the memory footprint small. We observe little loss of accuracy from BN to GN: 89.4% \rightarrow 89.0%. Streaming Training. TinyTL supports streaming training by replacing batch normalization (BN) with group normalization (GN), which supports batch 1 training. We observe little loss of accuracy from BN to GN: 89.4\% \rightarrow 89.0\% on Flowers, from 85.5\% to 88.9\% on Aircraft. We will include this feature in code release.

R5: Effects of freezing biases. Adapting biases is necessary. Without it, the accuracy drops by 1.7\% on Cars, 0.5\% on Flowers, and 4.1\% on Aircraft (Table B).

R5: Results without feature extractor adaptation. If disabling the feature extractor adaptation, the accuracy drops by 2.2\% on Cars, 0.6\% on Flowers, and 2.5\% on Aircraft (shown in Tab.1, page 7). Feature extractor adaptation is critical.

R5: Apply to conventional transfer learning. ‘Fine-tuning the full network’ can also benefit from feature extractor adaptation (FA). Compared to InceptionV3+Full, FA+Full improves the accuracy from 91.3\% to 93.2\% on Cars, from 96.3\% to 98.3\% on Flowers, from 85.5\% to 88.9\% on Aircraft. We will include this feature in code release.

R3: Streaming Training. TinyTL supports streaming training by replacing batch normalization (BN) with group normalization (GN), which supports batch 1 training. We observe little loss of accuracy from BN to GN: 89.4\% \rightarrow 89.0\% on Cars, 96.9\% \rightarrow 96.7\% on Flowers, 81.5\% \rightarrow 81.1\% on Aircraft. We will include the new results in the revision.

R3: Hardware deployment. Fig.4 used theoretical values as Pytorch does not support fine-grained memory management. We target co-designing the on-device training framework to fully exploit the theoretical benefits, which is beyond the scope of this paper. We will make this clear in the revision.

R3: Fig 3, Fig 6. In Fig.3, ‘ours’ refers to TinyTL FA from Tab.1. In Fig.6, the parameter size consists of two parts: i) frozen parameters (2.3MB,8bits); ii) trained parameters (11.3MB,32bits). We will make it more clear in the revision.