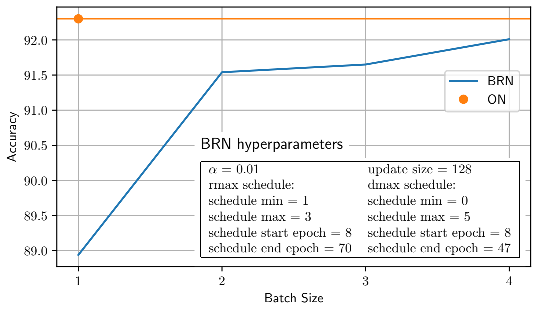

α = 0.01
rmax schedule:
schedule min = 1
schedule max = 3
schedule start epoch = 8
schedule end epoch = 70

update size = 128
dmax schedule:
schedule min = 0
schedule max = 5
schedule start epoch = 8
schedule end epoch = 47

Figure 1: BRN vs ON (CIFAR10/ResNet20).

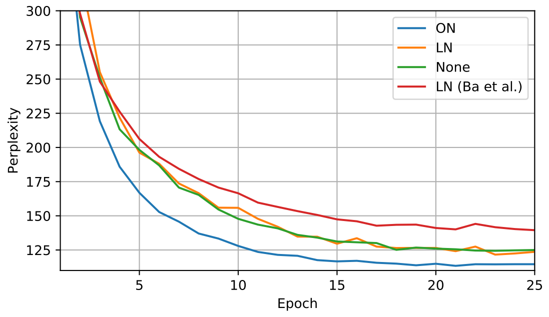

Figure 2: LSTM training.

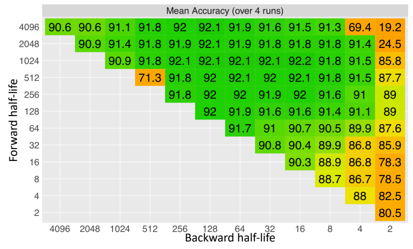

Figure 3: Hyperparameter sweep (CIFAR10/ResNet20).

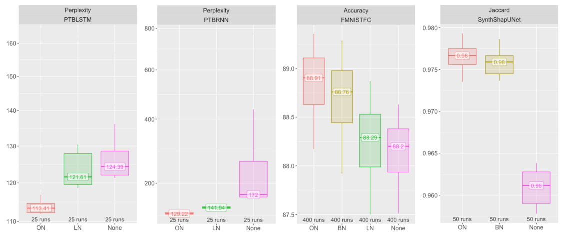

Figure 4: Run-to-run variability.

Dear Reviewers R1, R2, and R3: Thank you for your comments and suggestions to improve our paper.

Comparison with Batch Renormalization (BRN) [R1]: Online Normalization (ON) processes all data without batches, and differentiates through its estimation process. In contrast, BRN's statistical estimates are based on batches and it does not differentiate through its estimation. To counter instability, BRN adds two threshold hyperparameters to constrain its gradients to a neighborhood of the current batch's statistics. Like Instance Normalization, BRN cannot be used for dense layers at batch-1, and performs poorly on convolutions. The BRN paper reports 2% accuracy reduction for ImageNet at batch-4. We trained CIFAR-10 with BRN with even smaller batches (Figure 1), using hyperparameters and schedule based on the BRN paper's guidelines. Notice the sharp reduction in accuracy at batch-1.

Note on Batch Size [R1]: ON never requires batching. In Appendix E (supplemental materials) we present a method to run ON efficiently on GPUs. This mode preserves causality: ON still operates exactly as it did at batch-1, whereas the weights are updated at batch-N boundaries. This also allowed us to use the same hyperparameters (learning rate and momentum) of Batch Normalization for comparison.

Layer Norm (LN) LSTM [R1]: Our network is based on TensorFlow's LayerNormBasicLSTMCell. We compared it with the original version of Ba et al. After tuning its hyperparameters, we observed that it performs worse (Figure 2).

ON Hyperparameters [R1, R3]: ON removes the batch size parameter and introduces two decay rate parameters. For small scale experiments we characterized the sensitivity of ON to the decay rates expressed as half-life of averaging $h = {}^{1}/_{(1-\alpha)}$. Because the region of near-optimal performance is broad (Figure 3), we reused the same hyperparameter settings from CIFAR10 on ImageNet without any tuning. We will include this figure in the paper's appendix.

CIFAR Validation [R1]: We selected hyperparameter values in the middle of the near-optimal plateau (Figure 3, green). Note, this is not the best value observed in the sweep. We agree that using a holdout set is the best practice. Although we can not "undo" our experiment and forget the hyperparameter values, if we did run this sweep using a holdout set, the same plateau would still be discernible. Its center point will stable to the additional noise caused by the smaller dataset. The hyperparamters chosen for CIFAR also produced good results on ImageNet without any tuning.

Error Bars [R1]: Error bars reduced the plots' readability. Statistical characterization is present with the supplementary code (file experiments/ExpReproducibility.md). The run-to-run variability using ON is comparable to that of other normalizers. We will plot the full set of characterizations (example in Figure 4) and include them in the appendix.

Computation of μ and σ [R2]: We compute μ and σ per feature for each sample (line 188) and never over a batch. The layer scaling step is the only computation that looks across features within each sample (line 200).

Gradient Bias [R3]: Estimates of μ and σ can be made arbitrarily accurate with appropriate choices of the learning rate and hyperparameters that depend on the Lipschitz constant of the loss surface. In this case the expected value of the gradient will converge to the true gradient (Appendix D), therefore the gradients produced using ON are unbiased. In contrast, for batch normalization there is an inherent bias once the batch size is fixed. We provide a simple example of this in lines 126 - 130 and empirically show this effect for small batches (Paper Figure 2).