A ADDITIONAL PROOFS

Lemma A.1. *If a network performs distributed Cooling, then, compared to no Cooling,*

1. *the output of the i^{th} layer is $\mathbf{x}'_i = \beta^i \mathbf{x}_i$, for $1 \leq i \leq L-1$;*
2. *the input to each of the non-linearities is left unchanged;*
3. *the output logits are scaled by a factor of τ: $\mathbf{z}' = \tau \mathbf{z}$.*

Proof. We prove 1. and 2. by induction on i. For $i = 1$,

$$\begin{aligned}\mathbf{x}'_1 &= \beta\rho(\beta^{-1}(\mathbf{W}'_1\mathbf{x}_0 + \mathbf{b}'_1)) \\ &= \beta\rho(\beta^{-1}(\beta\mathbf{W}_1)\mathbf{x}_0 + \beta^{-1}(\beta\mathbf{b}_1)) \\ &= \beta\rho(\mathbf{W}_1\mathbf{x}_0 + \mathbf{b}_1) = \beta\mathbf{x}_1.\end{aligned}$$

From the last line, it is also clear that the input to the non-linearity is the same as without scaling. For $i > 1$,

$$\begin{aligned}\mathbf{x}'_i &= \beta^i\rho(\beta^{-i}(\mathbf{W}'_i\mathbf{x}'_{i-1} + \mathbf{b}'_i)) \\ &= \beta^i\rho(\beta^{-i}(\beta\mathbf{W}_i)(\beta^{i-1}\mathbf{x}_{i-1}) + \beta^{-i}(\beta^i\mathbf{b}_i)) \\ &= \beta^i\rho(\mathbf{W}_i\mathbf{x}_{i-1} + \mathbf{b}_i) = \beta^i\mathbf{x}_i,\end{aligned}$$

which proves both result 1. and 2. Regarding 3., the new output logits are given by

$$\begin{aligned}\mathbf{z}' &= \mathbf{W}'_L\mathbf{x}'_{L-1} + \mathbf{b}'_L \\ &= (\beta\mathbf{W}_L)(\beta^{L-1}\mathbf{x}_{L-1}) + \beta^L\mathbf{b}_L \\ &= \beta^L(\mathbf{W}_L\mathbf{x}_{L-1} + \mathbf{b}_L) = \tau\mathbf{z},\end{aligned}$$

where the second equality follows from 1. □

B ADDITIONAL RESULTS

Here we present additional results on CIFAR10. The training setup is the same as described in section 4.2. In Figure 3 we show the evolution of gradient norms during the training of two VGG networks: one with last layer Cooling and one with no Cooling. The learning rate is kept constant during training. We observe that the gradient norms of our proposed Cooling method greatly decay during the course of training, with two noticeable different decay patterns, reminiscent of learning rate schedules usually employed in the literature. The network trained with no Cooling shows less variation in the evolution of the gradient norms which actually increase over the course of training.

In Tables 3 and 4 we show different ablations on the VGG architecture with the CReLU and the ReLU activation functions, respectively. We report results for different training settings. We observe that last layer Cooling is almost always better or on par with other Cooling modes and it achieves the best performance across all setting when no learning rate schedule is used (i.e. the learning rate is kept constant during training).

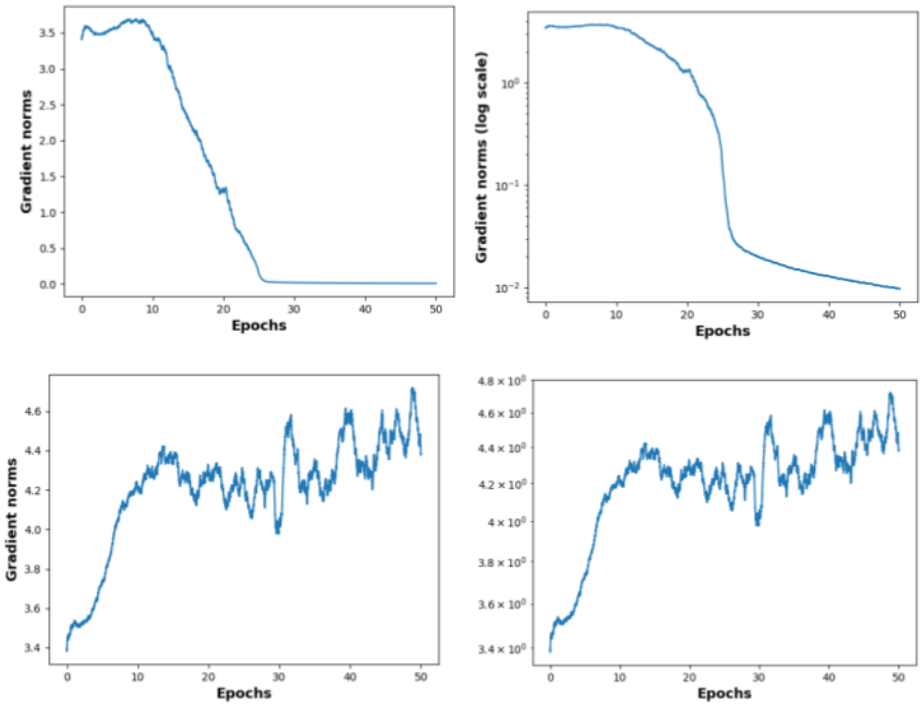

Figure 3: The evolution of gradient norms during training for a network trained using last layer Cooling (top) vs the baseline with no Cooling (bottom). Left and right columns correspond to plots without and with log scale, respectively. The two VGG networks were trained on CIFAR10 with no learning rate schedule. Compared to the baseline, we observe much more variation in the gradient norms with last layer Cooling, which exhibits two different decay patterns during training.

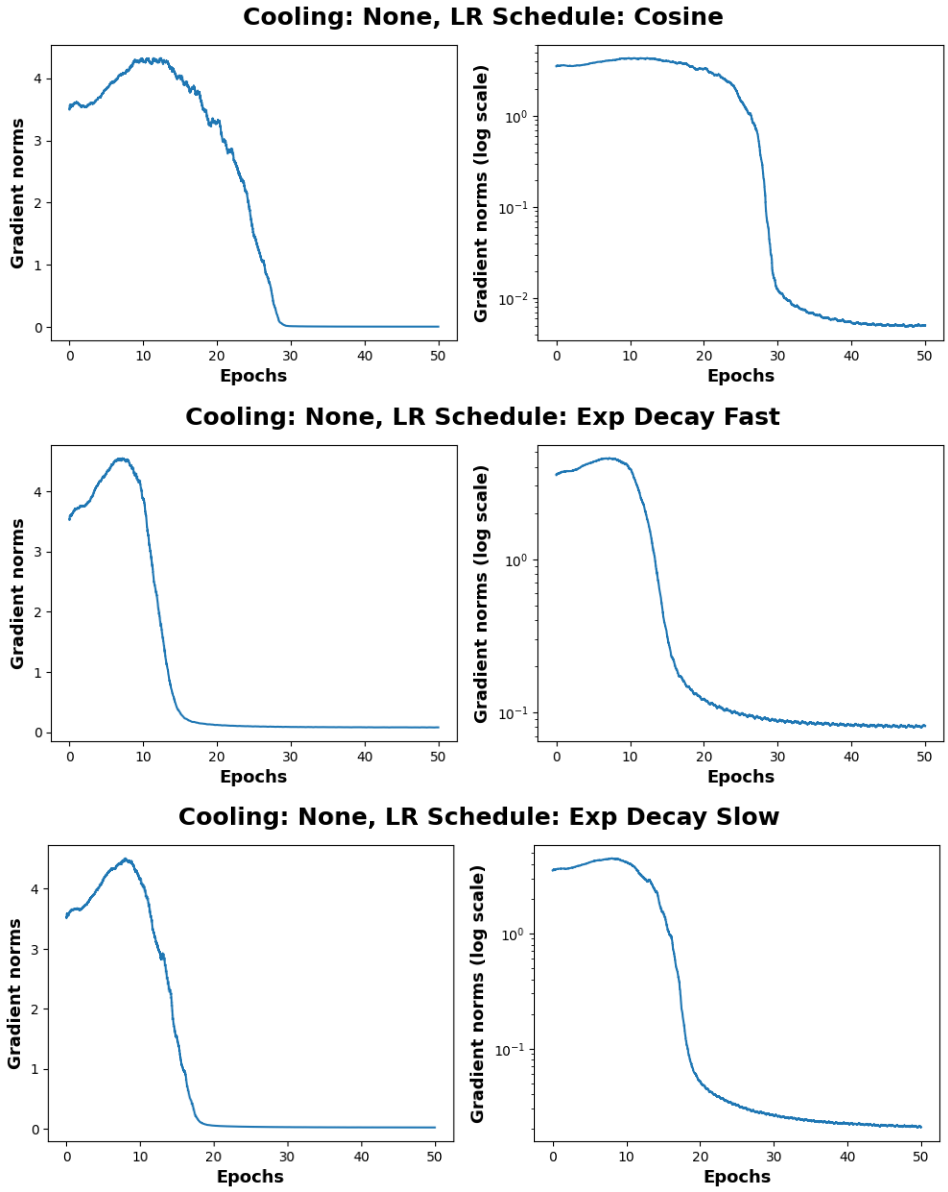

Figure 4: The evolution of gradient norms during training for a network trained without Cooling and using various learning rate schedules. Left and right columns correspond to plots without and with log scale, respectively.

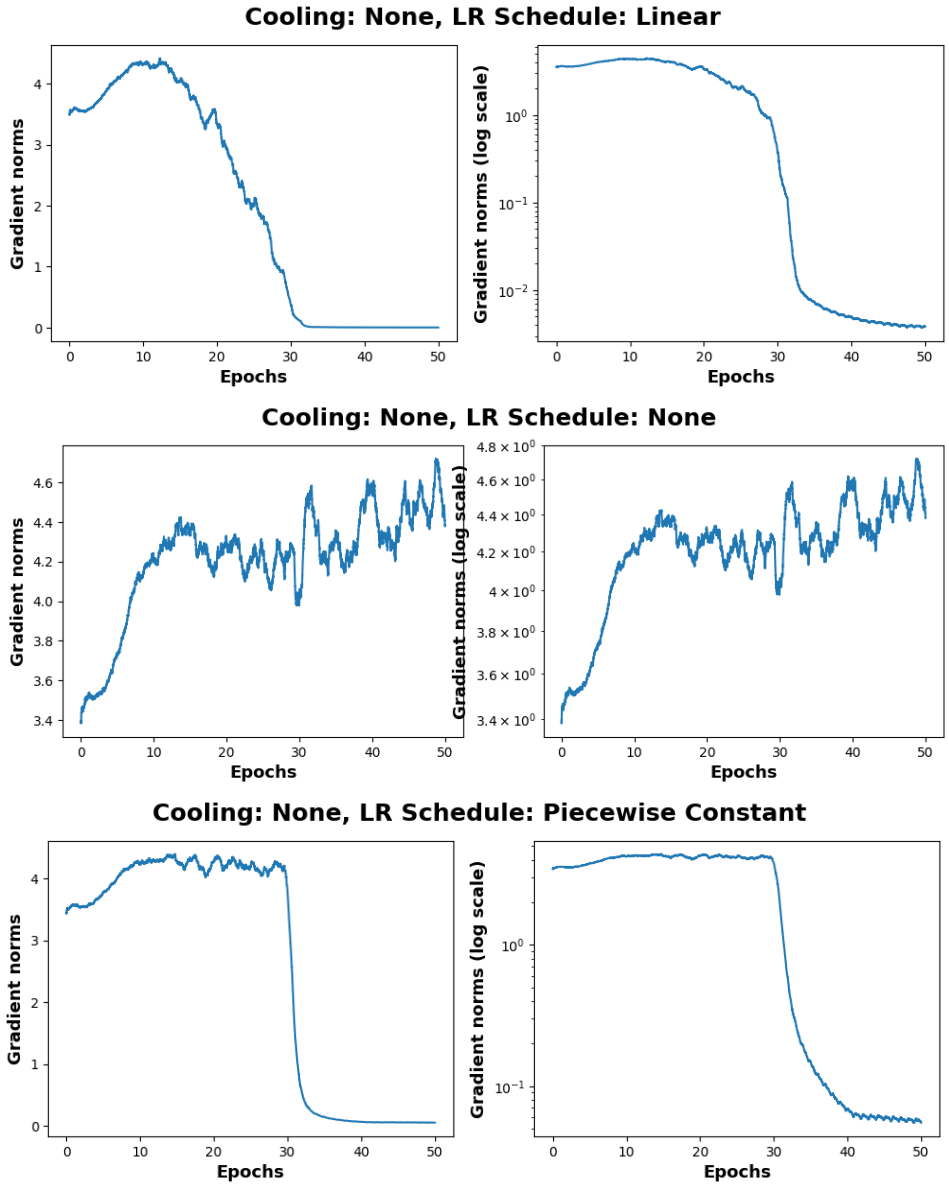

Figure 5: The evolution of gradient norms during training for a network trained without Cooling and using various learning rate schedules. Left and right columns correspond to plots without and with log scale, respectively.

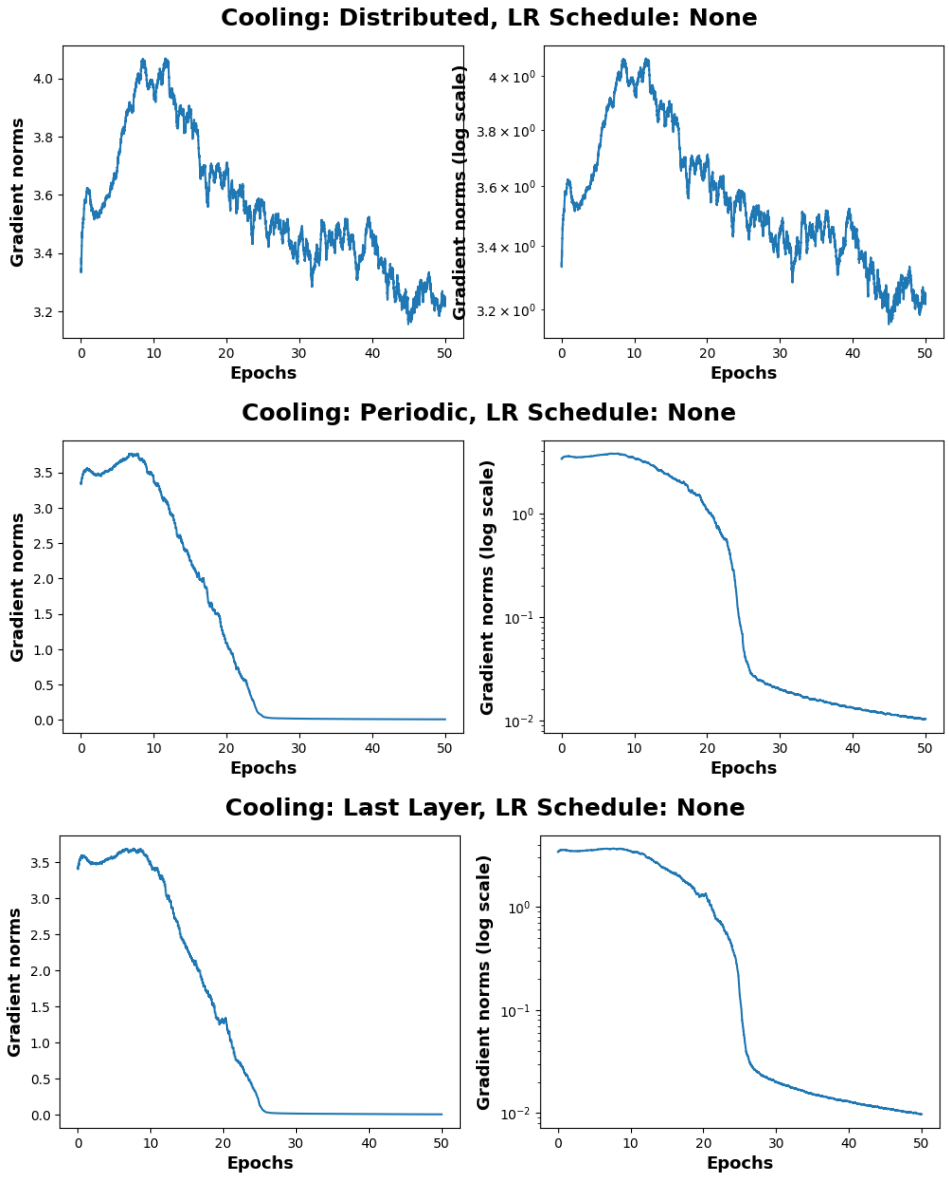

Figure 6: The evolution of gradient norms during training for a network trained with Cooling and no learning rate schedule. Left and right columns correspond to plots without and with log scale, respectively. We see a significant difference between periodic and last layer Cooling on the one side and distributed Cooling on the other side. Whereas distributed Cooling has a relative small effect on the gradient norms, the effect of periodic and last layer Cooling is considerable.

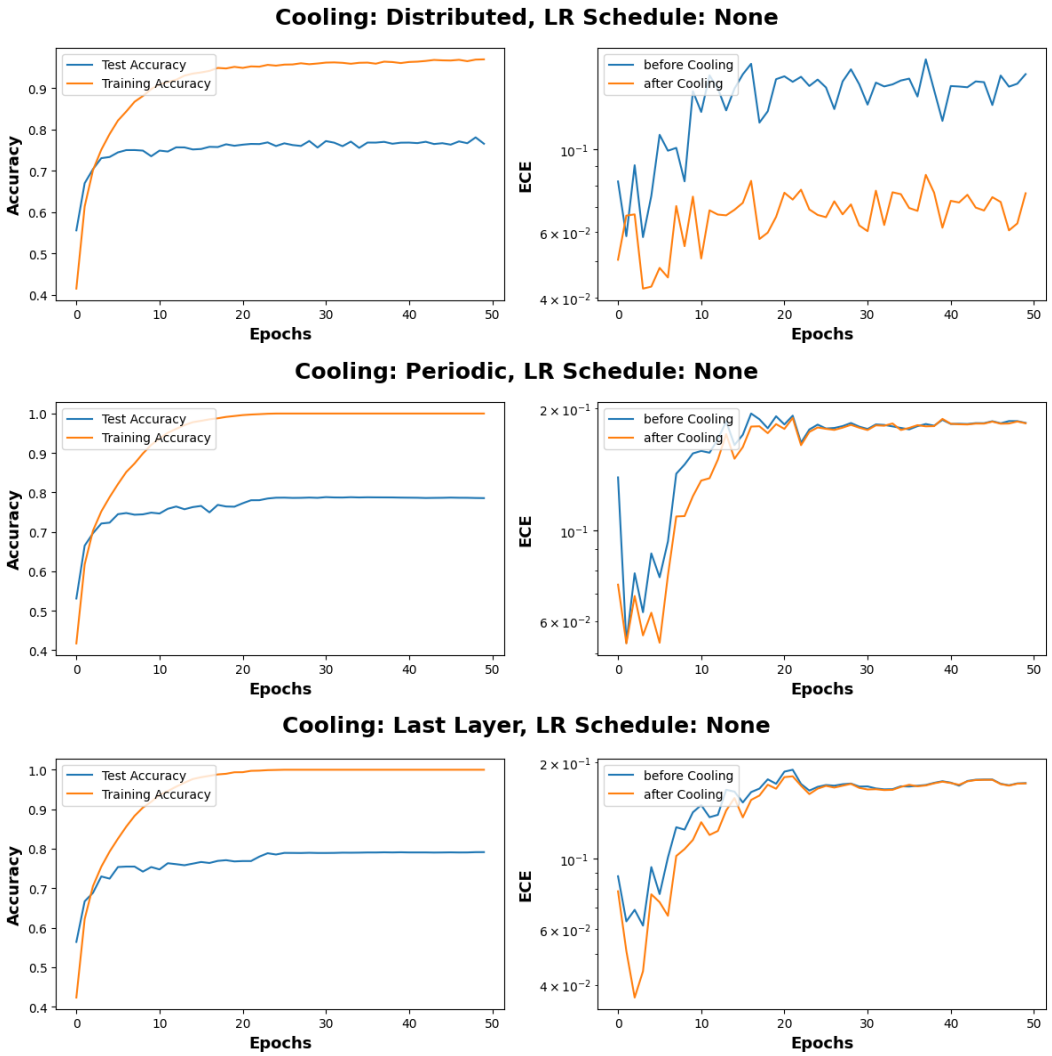

Figure 7: The evolution of accuracies (left) and ECEs (right) during training for a network trained with Cooling and no learning rate schedule. We see a significant difference between periodic and last layer Cooling on the one side and distributed Cooling on the other side. Whereas after each epoch, the ECEs decrease as a result of distributed Cooling, they stay almost constant for periodic and last layer Cooling. We also note that the networks reach a training accuracy of 100% when periodic or last layer Cooling was used. On the other hand, the training accuracy stays slightly below 100% for distributed Cooling.

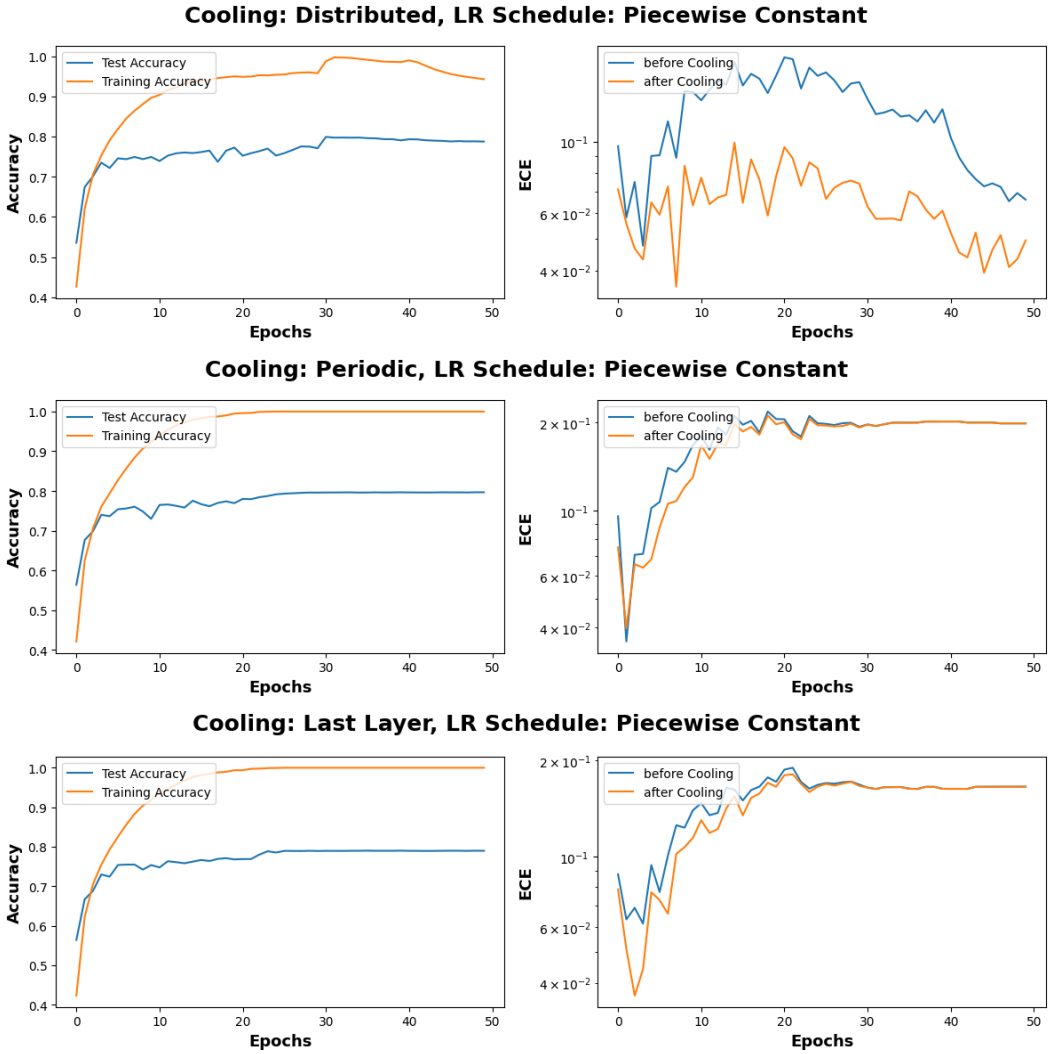

Figure 8: The evolution of accuracies (left) and ECEs (right) during training for a network trained with Cooling and a piecewise constant learning rate schedule. We see a significant difference between periodic and last layer Cooling on the one side and distributed Cooling on the other side. Whereas after each epoch, the ECEs decrease as a result of distributed Cooling, they stay almost constant for periodic and last layer Cooling. We also note that the networks reach a training accuracy of 100% when periodic or last layer Cooling was used. On the other hand, the training accuracy stays slightly below 100% for distributed Cooling.

Act	Lr	Warmup	Opt	No Cooling	Distributed	Periodic	Last Layer
CReLU	**None**	**False**	**Adam**	$71.8_{\pm 1.0}$	$81.2_{\pm 0.3}$	$83.9_{\pm 0.4}$	$83.6_{\pm 0.3}$
CReLU	**None**	**False**	**SGD**	divergence	$79.7_{\pm 0.4}$	$84.2_{\pm 0.1}$	$84.4_{\pm 0.2}$
CReLU	**None**	**True**	**Adam**	$72.4_{\pm 0.7}$	$80.8_{\pm 0.5}$	$83.6_{\pm 0.2}$	$83.5_{\pm 0.3}$
CReLU	**None**	**True**	**SGD**	divergence	$79.4_{\pm 0.3}$	$83.9_{\pm 0.2}$	$84.5_{\pm 0.0}$
CReLU	**PC**	**False**	**Adam**	$82.7_{\pm 0.1}$	$83.9_{\pm 0.1}$	$83.7_{\pm 0.3}$	$83.6_{\pm 0.2}$
CReLU	**PC**	**False**	**SGD**	divergence	$84.4_{\pm 0.3}$	$84.5_{\pm 0.2}$	$84.1_{\pm 0.3}$
CReLU	**PC**	**True**	**Adam**	$82.7_{\pm 0.2}$	$84.0_{\pm 0.2}$	$83.6_{\pm 0.4}$	$83.8_{\pm 0.2}$
CReLU	**PC**	**True**	**SGD**	divergence	$84.2_{\pm 0.1}$	$84.1_{\pm 0.2}$	$84.2_{\pm 0.0}$
CReLU	**Cosine**	**False**	**Adam**	$83.6_{\pm 0.4}$	$82.3_{\pm 0.2}$	$83.7_{\pm 0.2}$	$83.7_{\pm 0.2}$
CReLU	**Cosine**	**False**	**SGD**	$83.8_{\pm 0.2}$	$83.2_{\pm 0.4}$	$84.1_{\pm 0.3}$	$84.1_{\pm 0.1}$
CReLU	**Cosine**	**True**	**Adam**	$83.6_{\pm 0.1}$	$82.7_{\pm 0.2}$	$83.5_{\pm 0.1}$	$83.6_{\pm 0.2}$
CReLU	**Cosine**	**True**	**SGD**	$83.7_{\pm 0.1}$	$82.9_{\pm 0.5}$	$84.0_{\pm 0.1}$	$84.1_{\pm 0.0}$
CReLU	**Exp Decay Fast**	**False**	**Adam**	$81.4_{\pm 0.2}$	$80.7_{\pm 0.4}$	$81.1_{\pm 0.2}$	$80.9_{\pm 0.2}$
CReLU	**Exp Decay Fast**	**False**	**SGD**	$81.9_{\pm 0.4}$	$81.6_{\pm 0.3}$	$81.9_{\pm 0.2}$	$81.9_{\pm 0.1}$
CReLU	**Exp Decay Fast**	**True**	**Adam**	$81.6_{\pm 0.2}$	$81.1_{\pm 0.1}$	$81.4_{\pm 0.1}$	$81.7_{\pm 0.1}$
CReLU	**Exp Decay Fast**	**True**	**SGD**	$82.0_{\pm 0.5}$	$81.7_{\pm 0.3}$	$82.2_{\pm 0.2}$	$82.2_{\pm 0.0}$
CReLU	**Exp Decay Slow**	**False**	**Adam**	$82.0_{\pm 0.5}$	$80.6_{\pm 0.3}$	$82.1_{\pm 0.1}$	$81.8_{\pm 0.4}$
CReLU	**Exp Decay Slow**	**False**	**SGD**	$82.9_{\pm 0.5}$	$81.5_{\pm 0.4}$	$82.5_{\pm 0.1}$	$82.5_{\pm 0.3}$
CReLU	**Exp Decay Slow**	**True**	**Adam**	$82.4_{\pm 0.4}$	$81.0_{\pm 0.1}$	$82.2_{\pm 0.2}$	$82.1_{\pm 0.3}$
CReLU	**Exp Decay Slow**	**True**	**SGD**	$82.9_{\pm 0.3}$	$81.6_{\pm 0.1}$	$82.7_{\pm 0.0}$	$82.7_{\pm 0.1}$
CReLU	**Linear**	**False**	**Adam**	$83.4_{\pm 0.1}$	$82.3_{\pm 0.2}$	$83.5_{\pm 0.5}$	$83.5_{\pm 0.1}$
CReLU	**Linear**	**False**	**SGD**	$83.6_{\pm 0.3}$	$82.8_{\pm 0.2}$	$84.0_{\pm 0.1}$	$84.0_{\pm 0.2}$
CReLU	**Linear**	**True**	**Adam**	$83.2_{\pm 0.3}$	$82.3_{\pm 0.3}$	$83.4_{\pm 0.3}$	$83.5_{\pm 0.0}$
CReLU	**Linear**	**True**	**SGD**	$83.5_{\pm 0.4}$	$82.7_{\pm 0.4}$	$83.8_{\pm 0.2}$	$84.0_{\pm 0.0}$

Table 3: Additional Cifar10 ablations when using the CReLU activation with a VGG architecture. We report classification accuracy on different training settings: the learning rate schedule, using warmup, the type of optimizer and the different Cooling modes. We observe consistent improvement of last-layer Cooling against the baseline across all settings. Most important, the best performance is reached when not using a learning rate schedule.

Act	Lr	Warmup	Opt	No Cooling	Distributed	Periodic	Last Layer
ReLU	**None**	**False**	**Adam**	$74.6_{\pm 0.5}$	$77.3_{\pm 0.4}$	$78.1_{\pm 0.3}$	$78.2_{\pm 0.2}$
ReLU	**None**	**False**	**SGD**	$74.7_{\pm 1.1}$	$77.6_{\pm 0.2}$	$79.2_{\pm 0.4}$	$79.2_{\pm 0.2}$
ReLU	**None**	**True**	**Adam**	$74.7_{\pm 0.8}$	$77.3_{\pm 0.5}$	$78.4_{\pm 0.2}$	$78.5_{\pm 0.1}$
ReLU	**None**	**True**	**SGD**	$74.2_{\pm 0.2}$	$77.8_{\pm 0.9}$	$79.2_{\pm 0.3}$	$79.1_{\pm 0.3}$
ReLU	**PC**	**False**	**Adam**	$77.7_{\pm 0.6}$	$78.4_{\pm 0.1}$	$78.2_{\pm 0.4}$	$78.5_{\pm 0.2}$
ReLU	**PC**	**False**	**SGD**	$78.0_{\pm 0.2}$	$79.1_{\pm 0.5}$	$79.1_{\pm 0.5}$	$79.2_{\pm 0.0}$
ReLU	**PC**	**True**	**Adam**	$77.4_{\pm 0.2}$	$78.5_{\pm 0.4}$	$78.1_{\pm 0.3}$	$78.3_{\pm 0.1}$
ReLU	**PC**	**True**	**SGD**	$77.2_{\pm 0.3}$	$78.5_{\pm 0.1}$	$78.3_{\pm 0.2}$	$78.6_{\pm 0.4}$
ReLU	**Cosine**	**False**	**Adam**	$77.9_{\pm 0.1}$	$77.7_{\pm 0.2}$	$78.2_{\pm 0.1}$	$78.2_{\pm 0.2}$
ReLU	**Cosine**	**False**	**SGD**	$78.6_{\pm 0.2}$	$78.3_{\pm 0.1}$	$78.9_{\pm 0.3}$	$78.9_{\pm 0.1}$
ReLU	**Cosine**	**True**	**Adam**	$77.6_{\pm 0.2}$	$77.5_{\pm 0.3}$	$78.1_{\pm 0.3}$	$78.1_{\pm 0.3}$
ReLU	**Cosine**	**True**	**SGD**	$78.2_{\pm 0.3}$	$77.6_{\pm 0.1}$	$78.3_{\pm 0.2}$	$78.4_{\pm 0.7}$
ReLU	**Exp Decay Fast**	**False**	**Adam**	$76.0_{\pm 0.1}$	$76.3_{\pm 0.4}$	$76.2_{\pm 0.1}$	$76.1_{\pm 0.3}$
ReLU	**Exp Decay Fast**	**False**	**SGD**	$77.0_{\pm 0.1}$	$77.1_{\pm 0.1}$	$77.3_{\pm 0.2}$	$76.8_{\pm 0.6}$
ReLU	**Exp Decay Fast**	**True**	**Adam**	$76.3_{\pm 0.2}$	$76.3_{\pm 0.4}$	$76.4_{\pm 0.1}$	$76.3_{\pm 0.3}$
ReLU	**Exp Decay Fast**	**True**	**SGD**	$77.2_{\pm 0.3}$	$76.7_{\pm 0.6}$	$76.8_{\pm 0.2}$	$76.7_{\pm 0.1}$
ReLU	**Exp Decay Slow**	**False**	**Adam**	$76.6_{\pm 0.5}$	$76.2_{\pm 0.2}$	$76.7_{\pm 0.3}$	$76.8_{\pm 0.1}$
ReLU	**Exp Decay Slow**	**False**	**SGD**	$77.7_{\pm 0.3}$	$77.3_{\pm 0.4}$	$77.6_{\pm 0.1}$	$77.5_{\pm 0.3}$
ReLU	**Exp Decay Slow**	**True**	**Adam**	$76.9_{\pm 0.1}$	$76.2_{\pm 0.4}$	$76.7_{\pm 0.2}$	$76.7_{\pm 0.1}$
ReLU	**Exp Decay Slow**	**True**	**SGD**	$77.6_{\pm 0.2}$	$76.4_{\pm 0.4}$	$77.3_{\pm 0.1}$	$77.4_{\pm 0.1}$
ReLU	**Linear**	**False**	**Adam**	$77.5_{\pm 0.3}$	$77.7_{\pm 0.1}$	$78.1_{\pm 0.5}$	$78.0_{\pm 0.1}$
ReLU	**Linear**	**False**	**SGD**	$78.7_{\pm 0.6}$	$78.4_{\pm 0.3}$	$78.8_{\pm 0.5}$	$78.7_{\pm 0.1}$
ReLU	**Linear**	**True**	**Adam**	$77.9_{\pm 0.1}$	$77.5_{\pm 0.3}$	$78.0_{\pm 0.2}$	$77.9_{\pm 0.5}$
ReLU	**Linear**	**True**	**SGD**	$77.9_{\pm 0.4}$	$77.7_{\pm 0.1}$	$78.1_{\pm 0.3}$	$78.4_{\pm 0.0}$

Table 4: Additional Cifar10 ablations when using the ReLU activation with a VGG architecture. We report classification accuracy on different training settings: the learning rate schedule, using warmup, the type of optimizer and the different Cooling modes. The best performance is achieved by our proposed last layer Cooling variant when using a constant learning rate.