Supplementary Material: A Multigrid Method for Efficiently Training Video Models

Chao-Yuan Wu[1,2] Ross Girshick[2] Kaiming He[2]
Christoph Feichtenhofer[2] Philipp Krähenbühl[1]

[1]The University of Texas at Austin [2]Facebook AI Research (FAIR)

1. Supplementary Experiments

1.1. R101-SlowFast Results

We demonstrate generalization of multigrid training to deeper backbones by extending our default R50-SlowFast network to R101-SlowFast. All other designs and training procedures remain unchanged.

backbone		speedup	top-1	top-5
R50 (default)	Baseline	-	75.6	91.9
R50 (default)	**Multigrid**	**4.5**×	**76.4**	**92.4**
R101	Baseline	-	76.5	92.4
R101	**Multigrid**	**4.4**×	**77.0**	**92.9**

As expected, R101-SlowFast outperforms R50-SlowFast and we observe a consistent speedup and accuracy gain over the baseline with multigrid training.

1.2. Long Cycle Design

By default we use multiple long cycles that are synchronized with the stepwise learning rate (LR) schedule (*i.e.*, one long cycle period per LR stage). We compare our default design ('multi-cycle') with an alternative that uses only a single long cycle period ('single-cycle') throughout all of training. Note that the single-cycle design does not use a fine-tuning phase as it is unclear how to incorporate it into this design.

	long cycle design	speedup	top-1	top-5
Baseline	-	-	75.6	91.9
Multigrid	single-cycle	5.2×	74.4	91.8
	multi-cycle (default)	4.5×	**76.4**	**92.4**

We observe that our default, multi-cycle design works better. In the multi-cycle design, the later shapes, which are closer to the final testing distribution, are used with each LR. We conjecture that exposing the model to these shapes with the larger (earlier) LRs is important for generalizing to the testing distribution. In contrast, the single-cycle design only uses the later shapes with relatively low LRs.

1.3. Cosine Learning Rate Schedule

We develop multigrid training assuming a stepwise LR schedule. Next we experiment with a cosine LR schedule. We experiment with both the multi-cycle and the single-cycle design for long cycles. *No further modifications* are applied to multigrid training.

LR schedule		speedup	top-1	top-5
Stepwise (default)	Baseline	-	75.6	91.9
	Multigrid	**4.5**×	**76.4**	**92.4**
Cosine	Baseline	-	75.8	92.0
	Multigrid (single long cyc.)	**5.2**×	75.4	92.1
	Multigrid (multi long cyc.)	4.2×	75.3	92.1

We observe that multigrid training on a cosine schedule obtains a slightly lower accuracy than the default stepwise schedule. The lower accuracy is possibly due to the relatively smaller learning rates used in larger (later) shapes as the LR is monotonically decreasing in a cosine schedule. However, it still obtains a consistent speedup and a comparable accuracy to baseline, suggesting robustness of the multigrid strategy. The two long-cycle designs obtain a similar accuracy.

1.4. Testing Settings

Next we present results with additional test-time settings that are common in the literature. Here we use the 64-frame R50-SlowFast due to its high accuracy. Our multigrid method trains this model 5.5× faster than the baseline.

	center 224^2		3-crop 224^2		3-crop 256^2	
	top-1	top-5	top-1	top-5	top-1	top-5
Baseline	75.9	92.1	76.5	92.2	77.2	92.5
Multigrid	**77.6**	**93.2**	**78.1**	**93.5**	**78.1**	**93.4**

As expected, using 3-crop (left-center-right) testing improves accuracy for both baseline and multigrid training.

2. Supplementary Implementation Details

The I3D and I3D-NL architectures used in generalization analysis are shown below (assuming 16×224×224 inputs):

Layer	Specification	Output size
conv_1	1×7×7, 64, stride 1, 2, 2	16×112×112
pool_1	1×3×3 max, stride 1, 2, 2	16×56×56
res_2	$\begin{bmatrix} 1\times1\times1, 64 \\ 1\times3\times3, 64 \\ 1\times1\times1, 256 \end{bmatrix} \times 3$	16×56×56
res_3	$\begin{bmatrix} 1\times1\times1, 128 \\ 1\times3\times3, 128 \\ 1\times1\times1, 512 \end{bmatrix} \times 4$	16×28×28
res_4	$\begin{bmatrix} 3\times1\times1, 256 \\ 1\times3\times3, 256 \\ 1\times1\times1, 1024 \end{bmatrix} \times 6$	16×14×14
res_5	$\begin{bmatrix} 3\times1\times1, 512 \\ 1\times3\times3, 512 \\ 1\times1\times1, 2048 \end{bmatrix} \times 3$	16×7×7

'I3D-NL' additionally uses non-local operators [3] after blocks 1 and 3 of res_3, and blocks 1, 3, and 5 of res_4.

Something-Something V2 Training. We use a linear warm-up [1] for 2k iterations from 0.0001 and a weight decay of 10^{-6}. As Something-Something V2 requires distinguishing between directions, we disable random flipping during training. Following [2], we use segment-based input frame sampling, *i.e.*, we split each video into segments, and from each of them, sample one frame to form a clip.

Charades Training. The baseline method trains for 28k iterations with a learning rate of 0.0375, which is decreased by a factor of 10 at iteration 20k and 24k.

References

[1] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2

[2] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2

[3] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2