Supplementary for "Auto-X3D: Ultra-Efficient Video Understanding via Finer-Grained Neural Architecture Search"

1. Searching Details

We use SGD optimizer, with a cosine learning rate decay scheduler, to optimize model weights, and the architecture parameters are optimized with Adam optimizer. The learning rate of SGD optimizer is set to 1.6 and 0.025 for Adam optimizer. The searching takes 600 epochs, where the first 200 epochs are used for supernet warm up, where only the models weights are updated while keeping architecture parameters frozen. We use a batch size of 192.

2. Training Details

We initialize the derived model using [3]. Models are trained for 300 epoch. We adopt SGD as the optimizer with learning rate of 1.6, weight decay of 5×10^{-5}, momentum of 0.9, and a cosine scheduler of learning rate decay. The batch size is set to 192. We adopt dropout of 0.5 before the head of the network. Following [4], we apply auto-augment [1] on each frame during training (consistent within each input video clip). For AutoX3D model family (S, M, L), we adopt scale jittering ranges of $[182, 228]$, $[256, 320]$, and $[356, 446]$, followed by a crop size of 160, 224, and 312, respectively.

3. General Fair Channel Selection Discussion

In the main text we give a specific example of fair channel selection that we choose 5 channel candidates. Here we discuss a general channel selection strategy. Considering N channel candidates while N has to be an odd number, the update probability of each filter is $p = \frac{M}{N}$ following fair selection strategy, where $M = \frac{N+1}{2}$. We give another example where we choose 7 channel candidates, shown in Figure 1. The update probability of each filter is $p = \frac{4}{7}$.

4. More Results on Kinetics

In this section, we present the results of AutoX3D models without attention blocks in Table 1. We can observe that they still outperform X3D models with a large margin.

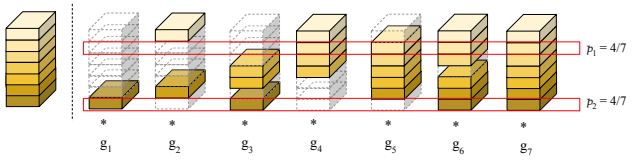

Figure 1: Another case of fair channel selection where $N = 7$ and $M = 4$. More general cases are similar to it.

Model	Params (M)	GFLOPS ×views	Top-1 Acc (%)
X3D-S [2]	3.8	2.7 × 30	73.3
AutoX3D-S w/o attention	3.8	2.7 × 30	74.2
AutoX3D-S (ours)	3.8	3.4 × 30	**74.7**
X3D-M [2]	3.8	6.2 × 30	76.0
AutoX3D-M w/o attention	3.8	6.2 × 30	76.4
AutoX3D-M (ours)	3.8	6.8 × 30	**76.7**
X3D-L [2]	6.1	24.8 × 30	77.5
AutoX3D-L w/o attention	6.1	24.8 × 30	78.3
AutoX3D-L (ours)	6.2	27.8 × 30	**78.8**

Table 1: **Comparisons with other NAS models on Kinetics-400.** AutoX3D and X3D models are evaluated using 30-views testing.

5. Searched Architecture

We present the architecture of our AutoX3D models in Table 2, where AutoX3D-L (Table 2b) is obtained by naively stretching the depth of AutoX3D-S (Table 2a) by $2\times$.

Max Input $C \times T \times S^2$	Block Type	Expansion	Channel	Number	Spatial Stride	Att. Block
$3\times13\times160^2$	data	1	24	1	1	-
$3\times13\times80^2$	1×3^2	1	24	1	2	-
$28\times13\times40^2$	t3_s3	2.25	16	1	2	-
$28\times13\times40^2$	t3_s3	5.25	16	2	1	-
$64\times13\times20^2$	t3_s3	4.5	48	2	2	-
$64\times13\times20^2$	t1_s3	2.25	48	3	1	GloRe
$132\times13\times10^2$	t1_s5	4.5	72	2	2	-
$132\times13\times10^2$	t3_s3	3.75	72	3	1	-
$132\times13\times10^2$	t5_s3	2.25	88	3	1	-
$132\times13\times10^2$	t3_s3	3	88	3	1	-
$264\times13\times5^2$	t3_s5	3.75	144	2	2	-
$264\times13\times5^2$	t1_s3	3	144	2	1	-
$264\times13\times5^2$	t3_s3	3	192	3	1	-
432	pool	-	432	1	-	-
2048	fc	-	2048	1	-	-

(a) **The architecture of AutoX3D -S**. AutoX3D -M has the same architecture with AutoX3D -S, except its input frame number is 16.

Max Input $C \times T \times S^2$	Block Type	Expansion	Channel	Number	Spatial Stride	Att. Block
$3\times16\times312^2$	data	1	24	1	1	-
$3\times16\times156^2$	1×3^2	1	24	1	2	-
$28\times16\times78^2$	t3_s3	2.25	16	1	2	-
$28\times16\times78^2$	t3_s3	5.25	16	5	1	-
$64\times16\times40^2$	t3_s3	4.5	48	4	2	-
$64\times16\times40^2$	t1_s3	2.25	48	6	1	GloRe
$132\times16\times20^2$	t1_s5	4.5	72	4	2	-
$132\times16\times20^2$	t3_s3	3.75	72	6	1	-
$132\times16\times20^2$	t5_s3	2.25	88	6	1	-
$132\times16\times20^2$	t3_s3	3	88	6	1	-
$264\times16\times10^2$	t3_s5	3.75	144	4	2	-
$264\times16\times10^2$	t1_s3	3	144	4	1	-
$264\times16\times10^2$	t3_s3	3	192	6	1	-
432	pool	-	432	1	-	-
2048	fc	-	2048	1	-	-

(b) **The architecture of AutoX3D -L**.

Table 2: **The architecture of AutoX3D models.**

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 1

[2] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1

[4] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 1