Watching Too Much Television is Good: Self-Supervised Audio-Visual Representation Learning from Movies and TV Shows

Anonymous Author(s)
Affiliation
Address
`email`

1 Implementation

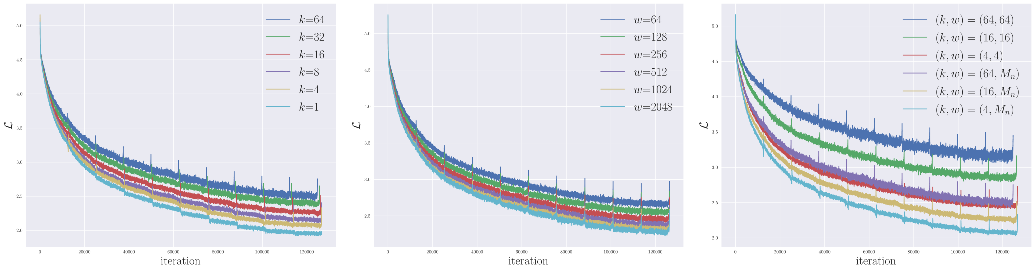
Please refer to `sampling_policy.py` included in the supplemental materials.

2 Effect of the Sampling Policy on Training Loss

Figure 1a illustrates the self-supervised training loss ($\mathcal{L}$) for different sample size (k) values. We observe that for the baseline ($k = 1$), the loss reduces faster than the cases where $k > 1$ while according to Table 1, the generalization to the downstream tasks is worse. We expected that an increase in k, *i.e.* larger portion of negative instances are comprised of hard negatives, makes it harder for the optimization to reduce the contrastive loss, a pattern which the Figure 1a perfectly shows.

Figure 1b illustrates the effect of sampling window (w) for a fixed sample size of $k = 16$. A smaller w increases the probability of instances sampled from the same long-form content to be temporally close, hence sound/look more similar to one another. That results in a harder instance-discrimination task which our objective function represents. We can see that the behavior of the self-supervised training loss very well follows the aforementioned intuition.

Figure 1c shows how drawing temporally adjacent samples from the same long-form content ($w = k$) makes the self-supervised task significantly harder specially when sample size (k) increases. We can see from Figure 1c and Table 1 that such hard constraint negatively affects the learning process and yields poor generalization to the downstream tasks.



(a) (b) (c)

Figure 1: Effect of the proposed sampling policy on the self-supervised training loss.

Table 1: Ablation study of the proposed sampling policy on different downstream tasks, measured by top-1 classification accuracy.

pretraining dataset: Movie				
k	w	HMDB51	ESC50	UCF101
1	–	60.32	86.50±0.46	85.69
4	M_n	61.37	89.91±0.06	85.38
8	M_n	62.09	88.75±0.31	86.06
16	M_n	62.92	88.33±0.36	86.30
32	M_n	61.04	88.00±0.42	85.98
64	M_n	61.30	86.83±0.17	85.43
16	64	60.26	87.00±0.20	83.61
16	128	60.58	86.50±0.31	85.30
16	256	62.02	87.75±0.23	84.85
16	512	61.30	87.08±0.13	85.38
16	1024	60.65	86.16±0.13	84.61
16	2048	61.83	87.66±0.13	85.11
4	4	60.19	88.00±0.51	84.66
16	16	56.86	88.75±0.31	82.71
64	64	57.45	84.58±0.37	82.68

pretraining dataset: TV				
k	w	HMDB51	ESC50	UCF101
1	-	56.40	85.50±0.54	84.37
8	M_n	61.50	87.50±0.42	85.96
16	M_n	61.69	89.00±0.47	85.64
8	64	60.58	88.00±0.23	85.96
8	128	60.00	85.66±0.27	85.77
16	256	61.30	86.41±0.27	85.01

3 Error bars for ESC50 Experiments

Table 1 is copied from the main submission except we have included the standard error over three times conducting the fine-tuning for the audio classification task. We did not observe meaningful sensitivity for the action recognition tasks.

4 Experimental Setup: More Details

Pretraining. Unless mentioned otherwise, all the models are trained for 10 epochs using ADAM [3] optimizer, with an initial learning rate of 10^{-4} which linearly warms up to 0.002 during the first epoch. We use a cosine learning rate schedule and a batch size of 512. Kernel size of both convolution layers in either h_f or h_g is 1.

Downstream Evaluation. On UCF101[6] and HMDB51[4], we respectively train for a total of 150 and 200 epochs using SGD, with an initial learning rate of 10^{-3} which linearly warms up to 0.2 during the first 25 epochs. Momentum and weight decay are respectively set to 0.9 and 10^{-4}. We use a cosine learning rate schedule and a batch size of 96. On ESC50[5], we train for a total of 200 epochs with warm up during first 25 epochs. Other optimization parameters are the same as those in action recognition tasks.

Comparison with state-of-the-art. To be comparable with the spatial resolution used by the best performing methods reported in Table 5 of the main submission, we resize the shorter side to 224 pixels, and then randomly crop them into 200×200 pixels. Spatial resolution in downstream evaluations are proportionally adjusted. Note that this is exclusive only to our numbers reported in the Table 5 of the main submission.

5 Comparison of the Training Cost

According to the Table 10 in the arxiv version of XDC[1], when using AudioSet [2], XDC models are trained for a total of 2.8M iterations [1], this increases to 9.3M iterations when using IG-Random/Kinetics datasets for pretraining. In comparison, our models whose performance is reported in the Table 5 of the main submission where trained only for $\approx$ 280K iterations which is orders of magnitude smaller.

References

[1] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, volume 33, pages 9758–9770, 2020.

[2] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.

[3] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.

[5] K. J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015.

[6] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[1] $(es/bs) \times te$